A Real-Time Microcomputer-Assisted System for Translating Aural, Monophonic Tones Into Music Notation as an Aid in Sightsinging (Pitch Extraction, Computer).

Randall Martin Kolb

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation


This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.
A REAL-TIME MICROCOMPUTER-ASSISTED SYSTEM FOR TRANSLATING AURAL, MONOPHONIC TONES INTO MUSIC NOTATION AS AN AID IN SIGHTSINGING

University Microfilms International

Copyright 1984 by Kolb, Randall Martin
All Rights Reserved
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark. ✓

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print ______
3. Photographs with dark background ______
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) ___________ lacking when material received, and not available from school or author.
12. Page(s) ___________ seem to be missing in numbering only as text follows.
13. Two pages numbered ___________. Text follows.
14. Curling and wrinkled pages ______
15. Other ____________________________________________________________
A REAL-TIME MICROCOMPUTER-ASSISTED SYSTEM
FOR TRANSLATING AURAL, MONOPHONIC TONES
INTO MUSIC NOTATION AS AN AID
IN SIGHTSINGING

A Dissertation
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
The School of Music

by
Randall Martin Kolb
B.M., Louisiana State University, 1975
M.M., Louisiana State University, 1977
May 1984
# TABLE OF CONTENTS

LIST OF TABLES ................................................................. iv

LIST OF ILLUSTRATIONS ................................................... v

ABSTRACT ........................................................................ vi

Chapter

I. GENERAL DESCRIPTION .............................................. 1

   Introduction .............................................................. 1
   General Capabilities .................................................. 3
   Historical Perspective .............................................. 6
   Design Problems .................................................... 12
      Fundamental frequency .......................................... 12
      Pitch ................................................................. 16
      Rhythm ............................................................... 19

II. IMPLEMENTATION OF THE TRANSLATOR .................. 23

   Hardware Implementation ......................................... 23
   Pitch follower ........................................................ 23
   Pitch detector interface ........................................... 25
   Terminology .......................................................... 25
   Theory of operation ............................................... 26
   Schematic diagram description ................................ 28
   The computer ......................................................... 34
   Other hardware ...................................................... 36
   Software Implementation .......................................... 37
   General description ............................................... 37
   Music Control Program .......................................... 39
   Sightsinger's Mirror ............................................... 51

III. THE USER'S VIEW OF THE SYSTEM ......................... 58

   Introduction .......................................................... 58
   Initiating the Session .............................................. 59
   The Main Menu ..................................................... 61
   The "Tunes" Option ................................................ 62
   The "Free Mode" Option .......................................... 68
LIST OF TABLES

1. MCP System Calls ................................................. 40
2. Time Utilization of the Computer Sightsinging System by
   the Participating Members of the Experimental Group .... 72
3. Sightsinging Examination Grades of Experimental Group
   Before and After the Experiment ........................... 74
4. Sightsinging Examination Grades of Control Group
   Before and After the Experiment ........................... 75
LIST OF ILLUSTRATIONS

1. Electronic Schematic Diagram of Pitch Detector Interface . . 29
2. The Main Menu .......................................................... 61
3. Key Identification Message ........................................... 63
4. Menu for Aborting Melody ............................................. 64
5. Instructions for Singing the Tonic Triad ......................... 65
6. Instructions Prior to Singing a Melody ............................ 66
7. Menu at Interruption of Melody Performance .................. 67
8. Menu at Conclusion of Melody Performance ................... 67
9. Menu at End of Melody Library ..................................... 68
10. Graph of the Grades of the Experimental Group ............ 77
11. Questionaire Given to Participating Members
    of the Experimental Group ........................................... 79
ABSTRACT

The problems of real-time pitch detection by computer were studied in an attempt to develop a learning tool for sightsinging students. Specialized hardware and software were developed for the task of solving three problems in real time: (1) the extraction of the fundamental frequency of vocal tones, (2) the detection of the pitch of those tones, and (3) the display of those pitches in standard music notation for visual feedback to the student.

The system that was developed has the capabilities of: (1) displaying standard music symbols in high resolution graphic images, (2) accessing a library of melodies to be used in student testing, (3) generating sound for the purposes of aurally establishing the key of melodies and playing those melodies, (4) displaying in real time the pitches the student is singing, (5) evaluating the student's performance of a melody, (6) transposing the key of melodies to a key that allows the melody to lie comfortably in the student's vocal range, (7) tailoring the sequence of melody selection to the abilities of the student, (8) creating and maintaining a file of session statistics for each user during each practice session without any human intervention, and (9) operating in a free mode in which the student may sing any series of pitches he desires.

These capabilities of the system were demonstrated and tested during the second half of the Fall semester in 1983 at Louisiana State
University in order to attempt to provide an indication of the system's pedagogical value. The system was used by freshman music majors who were randomly selected from the music theory classes. While formal proof of the system's value was not forthcoming from the experiment, informal results indicate that the system may hold a very useful position in the development of sightsinging skills.
CHAPTER I

GENERAL DESCRIPTION

Introduction

The sightsinging of music involves the translation of a visual form into an aural one. Taking music dictation is the exact reverse of this process. The development of sightsinging and dictation skills requires the building of a logical connection of the visual and the aural senses, and it is this connection which poses a significant problem to the sightsinging student. To build this connection the sightsinging student must experience the relationship between these visual and aural functions in as many ways as possible. At first this relationship can be experienced by having the student observe the conversion of one form to the other. For example, the student could listen to the performance of music while following the score. In a case such as this a visual form, the score, is translated into an aural form. There are several aids by which these translations may be made from notation to sound, but the translation of sound to notation is generally impractical. It would be very convenient if a student could sing any series of pitches he desires and then see how those pitches would be notated without the need for a human translator. Indeed, a computer-based system which is capable of translating aural tones into standard music notation symbols is presently feasible, and it is the development and testing of this translator which is the purpose of this study.
Obviously, students have successfully developed sightsinging skills without the aid of such a translator. One might wonder, however, whether the same students could have achieved the same results at a faster rate had such a tool been available. This is a relevant question to the music educator since one of the responsibilities of educators is to seek new learning tools and better methods of teaching. It is the intention of this investigator to provide for the music educator the information necessary for beginning the evaluation of a new learning tool for sightsinging. More specifically, the purposes of the present study are to: (1) design and build a translator capable of determining the pitch of aural tones and displaying those pitches in music notation and, (2) test the translator using sightsinging students to provide an indication of the translator's pedagogical value.

The microcomputer currently offers the most practical method of implementing a sound-to-notation translator since these computers are now generally available and are becoming more and more inexpensive. The inherent flexibility and speed of computers are necessary requirements for the implementation of the translator.

The computer, of course, requires a program to control it. A program is a series of instructions which cause the computer to behave in a desired manner. Programs are often referred to as "software" to distinguish them from the hardware on which they are executed. "Hardware" refers to the physical electronic circuits and body which make up the computer.
General Capabilities

In order to be useful the translator must be capable of performing certain functions and meeting certain requirements. The translator resulting from the present project exhibits the following capabilities.

The translator developed for this project displays standard music notation with a clarity that requires very little adjustment on the part of the user. While this is essential, a microcomputer's ability to display graphic images is not a facility which can be taken for granted. Only a fraction of the computers available at the present time are designed to allow the degree of control over the display image that is necessary to draw standard music symbols with a resolution that does not strain the eye.

Before a melody can be displayed in music notation the melody must exist in a form which has meaning to a translator, and to be accessible by a translator the melody must reside in a specific position in the computer's memory. Several independent melodies may be grouped together in this manner to form a library. This translator has a facility for selecting melodies from this library and presenting them to the user for sightsinging practice. The size of these libraries is sufficient to contain a wide variety of melodies, being limited only by the amount of memory available to the computer.

The ability to generate sound for the purposes of aurally establishing the key and playing the melody is a necessary requirement. Before a user can sing a displayed melody he must know the pitch of the key note, and when he is unable to sing a melody without help he must hear the melody played for him. As presently implemented, the
translator aurally establishes the key by playing the scale up and down one octave followed by a four-part chord progression: tonic, subdominant, dominant-seventh, tonic. On the other hand, melodies are played in unison.

The most important and unique capability of the translator is that of listening to a user sing and extracting the fundamental pitch of his tones in real time. Working in "real time" refers to the collection of data as it becomes available and processing it before subsequent incoming data needs to be collected and processed, as opposed to storing incoming data for processing at a later time. It is upon this capability that the value of the translator as a learning tool rests. While not perfect, the present translator does exhibit a degree of accuracy, speed, and flexibility that can generally be judged to be more than adequate.

Once a user's performance has been heard by the translator it must be evaluated. When a melody has been sung, the present translator points out errors of pitch to the user and expresses the degree of performance accuracy as a percentage. An additional performance attempt is always allowed of the user, or he may bypass the melody altogether. He may also have the melody played and re-played for him at any time.

In order to make the user as comfortable as possible, the actual pitch of the key note of melodies is transposed to a key that allows the melody to lie comfortably in that user's vocal range. If this were not done male users would likely be limited to melodies displayed in the bass clef, while females would be similarly limited to treble
clef use. Furthermore, accuracy of pitch is more likely when the voice is not strained.

Once the translator is given access to a library of melodies there are three available schemes by which melodies may be selected for use. Melodies may be selected in the order in which they are stored in the library or they may be selected randomly. Optionally, the melody selection scheme could be tailored to the needs and ability of each user by considering the level of difficulty of each melody and the user's performance accuracy. Thus a user who performs poorly on a given melody would then be given a less difficult example, while a student who performs well would be given a more difficult melody.

An auxiliary cassette tape recorder is used to collect data on the practice session of each user. This file of session statistics is automatically created and updated by the system. Records are periodically written to the file at intervals during the user's session without the need for any human intervention. This file contains information that identifies the user, his range, the date, the melodies selected, the number of times the user attempted a performance of a melody, his performance accuracy and much other useful data.

The translator is also capable of executing in a free mode in which a blank staff is displayed to allow the user to sing any series of pitches he desires. This function allows the user the capability of experimenting with the sound of various intervals while getting immediate visual feedback on the results. Such experimentation is likely to strengthen the logical connection between the aural and visual sense of intervals.
These capabilities of the translator were tested during the second half of the Fall semester in 1983 at Louisiana State University in order to attempt to determine some initial pedagogical value of the system. The translator was used by freshman music majors who were randomly selected from the music theory classes. While formal proof of the translator's value was not forthcoming from the experiment, informal results indicate that the translator may hold a very useful position in the development of sightsinging skills.

**Historical Perspective**

It should be unnecessary at the present time to defend the use of the computer as an educational tool. Much research and many experiments have already been done demonstrating the pedagogical value of the computer. Unfortunately, there remains a significant amount of reluctance on the part of many educators and students to accept the introduction of a device often viewed as impersonal and intimidating. This attitude is justified when one considers the vast amount of educational programs that are so poorly written that the intended learning experience becomes instead a session of frustration. However, it would be unfair and unwise to place the blame on the computer when the culprit is poor software.

At the root of this unpleasant situation lies the fact that computers do not yet communicate fully in human languages. Instead of adapting the machine to the person, the person has often found himself adapting to awkward functions that suit only the machine. This situation is avoidable, and man-machine interfaces are being greatly improved.
The general effectiveness of computer-assisted instruction (CAI) has been studied by investigators in both general and specific applications. In 1974 the Association for Educational Data Systems journal reported on thirty-three studies,¹ the majority of which found that CAI demonstrated the ability to increase achievement equal to or above traditional instruction methods. The most consistently effective mode of instruction found in almost all of the studies was that of drill and practice. Perhaps not surprisingly the report found that CAI was more effective for poor students than it was for the better students. This observation probably has less to do with the use of CAI versus traditional instruction methods than with the fact that poor students have more room for improvement.

Since the middle 1960s there have been numerous experiments in which the use of CAI has been applied to the field of music. One of the earliest of these experiments, described in 1967 by Kuhn and Allvin at Stanford University,² was concerned with the use of a pitch extraction device, working in conjunction with an IBM 1620 computer, to test the accuracy of pitches sung by students. One of the main objectives of this Stanford experiment was to test the feasibility of computer-assisted music instruction. Since the interaction between student and machine was positive and the students showed an increased awareness of pitch, the experiment can be judged to be very successful.

The differences between the Stanford experiment and the experiment being conducted by this writer are important. First, Stanford's pitch extraction procedure was not performed in real time; consequently, there was no immediate visual feedback for the students.
Second, Stanford's input/output interface employed a typewriter-like terminal which was incapable of displaying standard music notation. Third, Stanford's use of the IBM 1620 computer cannot be considered to be an inexpensive solution, though it is likely that the program could be rewritten today for a less expensive computer with the same capabilities.

At Pennsylvania State University Deihl\textsuperscript{3} conducted a study from 1967 to 1969 to develop a CAI program that trained students to develop an ability to discriminate between subtleties of phrasing, articulation, and rhythm by comparing written music examples with various performed versions of those examples. According to pretest and post-test scores of the participants the program was apparently successful in improving the aural discrimination abilities of the students.

The University of Illinois was the scene of much activity in the early 1970s centered around the use of the PLATO computer-assisted system. Placek\textsuperscript{4} demonstrated a program for teaching rhythm and developed a device capable of fast random access to the rhythm exercises. Later Peters and Sanders\textsuperscript{5} used the same device, with the added capability of recording student performances, to provide drill and practice on sightsinging exercises. The computer, however, did not actually evaluate the performance; the students were expected to compare the pre-recorded model with their own recorded performance. Peters also conducted a study in which he found it feasible to evaluate the pitch and rhythm of a trumpet performance by computer. He concluded that the cost would be unreasonably high and the task of removing unwanted noise would be a formidable one.
In 1973 Thompson developed a program at the University of Utah for teaching the sightsinging of pitch phrases. The impetus behind the study was the generation of the pitch phrases by the computer according to the level of difficulty selected by the student. Again a formal experiment showed that the sightsinging ability of the students improved as a result of the program.

By 1974 Kuhn had developed drill and practice exercises for ear training at Stanford University. Exercises were created to teach the aural recognition of intervals, triads, chord progressions, modulations, and rhythmic and melodic dictation. The most significant features of this system were: (1) the use of sound for output as opposed to textual information, (2) immediate feedback to provide instant reinforcement, (3) individualization by allowing the student to choose the material according to his needs, and (4) the automatic recording of student performance data for evaluation of learning patterns.

Hofstetter, at the University of Delaware, also developed a series of ear training exercises for intervals, chord qualities, harmonies, and rhythms. Named GUIDO (Graded Units for Interactive Dictation Operations), the system was originally developed on a Burroughs 6700 computer using Tektronix graphics terminals but was later adapted for use on the PLATO system. A formal experiment using freshman ear-training students demonstrated the ability of the system to help students improve their ear training abilities more than those students who did not use the system.

Still another successful installation for ear training drill and practice was initiated in 1977 at North Texas State University.
Designated the Automatic Music System (AMUS), the system is based on microcomputers that are interfaced to a Hewlett-Packard 2000 computer. In an experiment involving two freshman music theory classes, each taught by the same instructor, one class used the computer-assisted instructional materials while the other class did not. At the end of the experiment, the students who had used the computer showed a significantly higher level of performance ability than the other group. Encouraged by this success, the university has since expanded its ear training laboratory with additional computers and developed additional instructional material.

There are currently several projects underway to study the feasibility of real-time pitch extraction of vocal and instrumental tones. There is little information available on these studies since as of this writing none of the investigators has published any information on the status of their projects. At the present time there are four universities at which significant studies are known to be taking place: Florida State University, University of Delaware, North Texas State University, and Laval University.

The projects taking place at Florida and Delaware are very similar in that both are currently being developed on large computers through the PLATO system of computer-based education. The Delaware project, developed by Hofstetter and Conrad, is still in the development and testing phase and does not currently employ the capability of displaying pitches in standard music notation. In the Florida project, headed by Taylor, pitches are displayed in music notation though the interpretation of rhythm has not yet been implemented. Currently work is being done to establish the communication links
between the terminals and the PLATO system, but future plans include the transfer of the sightsinging system to microcomputers. Neither the Delaware nor the Florida systems are yet available for the use of students.

At North Texas State University Clendinning studied the problems of pitch detection and developed an algorithm for accomplishing the task. A software simulation of the algorithm showed promising results and a hardware implementation is currently in development by Dworak. Kolosick estimates that the hardware will be ready by 1985 at which time he hopes to develop software for the use of sightsinging students on the Apple computers currently in use in the ear training lab.

Under Prevel's direction, the music and computer science departments at Laval University have developed a microcomputer system for ear training exercises called the Exercette system. It has now come to the attention of this writer that as recently as three months prior to the time of this writing, a hardware pitch extraction device was developed to connect with this system giving it the capability of real-time pitch detection for display in music notation. While this system has not been evaluated by this writer, the claims made by Prevel appear to indicate that his system provides a very useful and inexpensive learning tool for the sightsinging and ear training student.

The recent infusion of microcomputers into the marketplace has helped to remedy one of the most serious problems associated with the use of CAI, that of affordability. Most of the experiments and studies cited above were developed on computer hardware costing
several thousand dollars. For example, Hofstetter estimates that just one PLATO terminal as installed at the University of Delaware costs approximately $3850 per year to operate. The microcomputers available today can be purchased for much less making them more attractive and cost-effective. Indeed, several hundred thousand such computers have already been purchased by individuals and schools intent upon taking advantage of the educational potential that has already been successfully demonstrated.

It is not enough that a learning tool exhibit sound pedagogical value; it must also be affordable to the institutions and students who desire to use it. The translator resulting from the present study was developed with this goal in mind. In addition to the technical and pedagogical goals of this project a financial goal of $2000 was set for the translator. It is intended that the final realization of the translator be affordable by those university music schools which are interested in providing such a tool for the laboratory use of theory students.

**Design Problems**

**Fundamental Frequency**

One of the reasons that a translator has not been developed before the present time is that the problems encountered are very serious. Since human intelligence is not yet programmable and computers and other hardware have not been programmed to be as versatile, intelligent, and complicated as the human brain, the interpretation of some data by computers will likely diverge from the interpretation of the same data by humans. The act of extracting the pitch of a complex
tone, while a trivial matter for the ear-brain combination, is a formidable task for the computer. Since it is not yet clearly understood how the ear perceives pitch the ear's technique of pitch detection can not be imitated by hardware and software. Other techniques are available, but they can not yet be expected to meet the ear's level of accuracy and speed. A certain degree of tolerance must be required, therefore, of the users of such a translator in terms of performance.

The first problem one encounters is that of determining the fundamental frequency of a tone. The human voice generates complex tones whose harmonic content and distribution vary widely from person to person, from note to note, and even from one cycle of a frequency to the next. It is this wide degree of variance over time and context that is the root of most of the problems.

Consider the case in which a person deliberately attempts to sing a single pitch with as much stability as he is capable. Even though the singer may not perceive any change in the sound a close analysis of the sound will probably reveal wide variations of harmonic content and instability in the fundamental frequency. An analysis of the harmonics will show not only changes in the number of harmonics present, but also an independence of each harmonic in terms of its relative amplitude. Indeed, often the first harmonic, one octave above the fundamental, exhibits an amplitude larger than the fundamental. The characteristics of these harmonics are a function of tension on the vocal cords, the air pressure used to generate the tone, the structure of the vocal tract and the structure of the physical environment in which the person is singing. Each of these
factors serves to either enhance or suppress certain frequencies, and it is clear that the combined effect of all four factors can be expected to provide the potential for an infinite number of possible waveforms. Add to this scenario the fact that the waveform undergoes significant changes depending upon the position of the pitch in the singer's range and the extraction of the fundamental frequency is made even more difficult. If the singer is moved to a different room or another part of the same room, the characteristics of the waveform will change. The same result occurs if the distance between the singer and the microphone is changed.

Further complicating the matter of pitch detection is the element of noise. Besides the obvious sources of background noise there is the noise introduced by the singer himself. The sound of the pronunciation of consonants while singing does not contain just one fundamental frequency; thus, no pitch can be detected. The sounds of breathing must also be ignored. It must also be recognized that all of the users of the translator cannot be expected to sing with the same purity of tone. Some voices will be inherently more noisy than others.

Despite the unpredictable nature of the tones, many techniques and algorithms for extracting the fundamental frequency have been devised with varying degrees of success. An algorithm is a methodical procedure for solving a specific problem. Two of the most sophisticated of these techniques are Fourier transform analysis and autocorrelation. Fourier transform analysis is a mathematical concept which expresses the relationships between a waveform and its frequency spectrum. In autocorrelation analysis a section of a waveform
covering many cycles of the frequency of the tone is compared with itself at various time delays and the delay that yields the highest correlation between the two sections is assumed to represent the time of one cycle of the fundamental frequency.

The most significant problem with these techniques is speed. While the Fourier transform is probably the most accurate, the time required for the computation of just one segment of the sound on an average microcomputer can span a period from several seconds to several minutes depending upon the implementation. Autocorrelation also requires a large number of calculations and thus a large amount of time, but a modified autocorrelation algorithm has been devised to minimize the number and complexity of these calculations. A software implementation of the modified autocorrelation algorithm by this writer achieved a computation time of approximately one second, but the degree of accuracy was sacrificed to an unacceptable level. Clearly, software solutions that are highly accurate are not suitable for real-time use.

The solution to this problem can be found by implementing the fundamental frequency extraction procedure in hardware. Since electronic circuits can generally be designed to perform complicated functions much more quickly than those functions can be performed in software, it should be possible to obtain the speed necessary for real-time pitch detection without sacrificing accuracy.

The hardware approach to fundamental frequency detection is not without problems. One effort toward this end, the hardware implementation of the modified autocorrelation algorithm by Dubnowski, Schafer, and Rabiner, resulted in a circuit of approximately 150
integrated circuits. While the solution is suitable for real-time use the circuit is rather large and prone to errors in low frequencies.

More recently DSP Systems Corporation has announced the availability of an electronic circuit which performs Fourier transforms in real time. While the circuit has not been tested by this writer the claims of the manufacturer appear to indicate that the circuit would satisfy the speed and accuracy requirements of a real-time pitch detector. Unfortunately the $2800 purchase price places the circuit out of the range of the financial goals of this project.

The solution to the fundamental frequency detection problem chosen for this project is a hardware pitch and envelope follower designed for electronic music studio applications by Gentle Electric of Delta, Colorado. This device extracts the fundamental frequency and amplitude of complex tones for use in the control of electronic music synthesizers. The pitch extraction algorithm used cannot be described since it is currently proprietary information, but the performance of the circuit has proven to be excellent in terms of both speed and accuracy. Furthermore, the purchase price of the hardware is within the range of the financial goals of this project.

Pitch

One must be careful to distinguish between the fundamental frequency of a tone and its pitch. Extracting the fundamental frequency of a tone is only the first step in determining the pitch of the tone. One cannot successfully mimic the pitch perception abilities of the ear without considering the context-sensitive nature of pitch. To ignore the context in which a frequency occurs is to ignore the role
of the brain in the interpretation of the sound as it is processed by the ear.

The fundamental frequency of a tone at a given point in time can easily be assigned to a pitch. One need only decide upon the boundary frequency between adjacent pitches and then find the pitch whose upper and lower boundary frequencies are above and below, respectively, the given frequency. For example, if the boundary frequency between the pitches $a^1$ (440 Hz) and $b$-flat$^1$ (466 Hz) is 453 Hz, and the boundary between $a^1$ and $g$-sharp$^1$ (415 Hz) is 427 Hz, then any given frequency between 427 Hz and 453 Hz will be assigned to the pitch of $a^1$. This technique of pitch perception will work very well provided that every tone that is processed is always very stable and extremely well tuned and that changes from one pitch to another are instantaneous. Such demands, however, cannot and should not be placed on human performers.

Again consider the case in which a person deliberately attempts to sing a single pitch with as much stability as he is capable, and also assume that the fundamental frequency is being accurately extracted. Even though the ear will generally perceive one pitch, a graph of the fundamental frequency extracted over a period of time may show excursions into the range of the pitch one-half step above and/or below the pitch perceived. If the singer intentionally imposes a frequency modulated vibrato on the tone, the range of frequencies extracted may cover an even wider range of pitches. Clearly a frequency extracted at any one point in time may or may not belong to the class of frequencies associated with the perceived pitch. The role of a given frequency, therefore, cannot be determined until a number of
frequencies occurring before and after the given frequency have been evaluated.

The exact number of frequencies evaluated will depend upon the rate at which the frequencies are given. This rate is referred to as the sampling rate. Clearly if the sampling rate is too slow, then quick changes in pitch will be missed. If the sampling rate is set to the speed at which the fastest changes in pitch are expected, then there will be only one frequency sample for those fast pitches. As seen above, one sample is not enough to accurately determine a pitch. Therefore, the sampling rate must be increased to a level at which even the shortest expected pitch will allow enough time for several frequency samples. The translator developed as a result of this project employs a modest sampling rate of 50 Hz; that is, the fundamental frequency is sampled 50 times per second. Since the human voice is relatively slow to change pitch, this sampling rate proved adequate.

It should be noted that a faster sampling rate does not necessarily yield a faster rate of pitch detection. The ear requires a certain amount of time before a pitch can be detected. Since the ear needs at least two complete cycles of a frequency in order to determine a pitch, the exact amount of time required is dependent upon the frequency. Two cycles of a frequency of 100 Hz, for example, would require one fiftieth of a second. This frequency corresponds roughly to the pitch of great G, one and one-half octaves below middle C (C). Higher frequencies would, of course, require less time. The pitch detection rate of the translator, however, could be no faster than the rate needed for the lowest expected pitch. If the lowest
expected pitch is great C (65 Hz), two octaves below c\textsuperscript{1}, then the pitch detection time would be limited to approximately one thirtieth of a second. This time represents the shortest possible duration of a pitch to be perceived by the ear and should be considered to be only the upper limit for a pitch detection rate for a machine. A more practical value of time will have to be selected since the human voice rarely produces tones that are stable after only two cycles. By means of experimenting with the translator it was determined that the detection of pitch at a level of accuracy comparable to the ear required a period of approximately one-fourth of a second.

The use of faster pitch detection rates demonstrates the type of problem one encounters when trying to replicate a function performed by the brain. For example, unintelligent hardware does not distinguish between a frequency modulated vibrato and a rapid alternation between two pitches. Likewise, if a person sings a portamento between two pitches, all of the pitches in between may also be detected. Similar inaccuracies result when a person sings a large interval, slightly missing the second pitch and quickly adjusting it. In these cases it becomes apparent that the most serious problem with the translator is that it can be too accurate. It is also clear that the brain plays a very sophisticated role in what appears to be a simple matter.

**Rhythm**

The interpretation of rhythmic data poses problems very similar to those found in pitch detection. Every note that is sung will not necessarily be held for the entire value of the written note. The distinction between actual rests and articulations that shorten note
values is not programmable to a high degree of accuracy. In some cases, such as in the execution of repeated notes, the breaks between notes can be ignored. In other cases, ignoring breaks between notes would constitute ignoring a singer's attempt at a precise performance. A decision algorithm which correctly resolves these questions of interpretation in every case cannot be expected. Instead a compromise will have to be found and singers will have to accept a certain degree of discrepancy.

It is not uncommon for sightsinging students to pause at difficult passages in a melody before continuing. If the translator could not identify such a hesitation then every following pitch will likely be found to be incorrect, since the student's position within the melody will not be known. Ignoring these pauses, however, overlooks a performance error. Similarly if the student slows the tempo either suddenly or gradually it will be difficult to decide whether the discrepancy is due to a rhythm error, a pitch error, or a tempo error.

It should be noted that the rhythmic value of a pitch cannot be known until the following pitch or rest is detected. Furthermore, if the pitch detection algorithm does not perform accurately, the ability to interpret rhythm becomes useless except in those cases in which rhythm is used alone for its own sake. For this reason it was felt that the interpretation of rhythm, though useful, was not essential to the purposes of the present project, and thus, was not implemented.
Notes


10 Interview with Michael Arenson, University of Delaware, Newark, Delaware, 18 January 1984.


13 Interview with J. Timothy Kolosick, North Texas State University, Denton, Texas, 18 January 1984.

14 Interview with Martin Prevel, Laval University, Quebec City, Canada, 21 January 1984.


17. Ibid.


CHAPTER II

IMPLEMENTATION OF THE TRANSLATOR

**Hardware Implementation**

**Pitch Follower**

The hardware selected for the task of extracting the fundamental frequency for this project is the Gentle Electric Model 101 pitch and envelope follower, a device designed for electronic music studio applications. This pitch follower is currently available in a self-contained module for approximately $800, but the circuit boards alone can be purchased for less than $300. While the more expensive module is suitable, it is the less expensive alternative which has been selected for this project.

Since the pitch follower is designed as an interface between external sounds and electronic music synthesizers it provides many functions that are unnecessary for the purposes of the present project. The pitch follower is intended to process a waveform that is input through a microphone and provide outputs which have characteristics of that waveform in a manner which is suitable for the control of synthesizers. For example, one of the outputs provided is a signal whose voltage is proportional to the fundamental frequency of the input signal. As the fundamental frequency of the input signal rises, so does the voltage of the output signal. This function is very useful for the control of voltage-sensitive modules of synthesizers.
such as voltage-controlled oscillators, voltage-controlled amplifiers, and voltage-controlled filters. Another example is the linear envelope output signal whose voltage is proportional to the peak amplitude of the input signal.

The output signal of interest in the present project is the fundamental frequency pulse wave output signal. This signal is a simple pulse wave at the fundamental frequency of the input signal and is passed directly to the interface hardware described below. The hardware requires at least two cycles of the input signal to extract the fundamental frequency so the time required for the pulse wave output to stabilize is dependent upon the frequency of the signal. While the fundamental extractor is not confused by harmonics in the input signal, it cannot work with more than one tone at a time nor can it work with the unusual timbres of some bells. For the purposes of this study, however, it is well suited.

Two other potentially useful signals provided are the "Gate" and "Good Pitch" signals. The "Gate" signal jumps to ten volts when an input signal is present and falls back to zero volts when there is no input signal or the input is too soft to be useful. The "Good Pitch" signal jumps to fifteen volts when the input signal is pitched and has settled for at least two cycles. It falls back to zero volts when the pitch changes or when the input signal is unpitched.

None of the output signals provided by the pitch follower are in a form that can be directly used by a computer. Microcomputers typically operate with TTL (transistor-transistor logic) signals which are in only two possible states: low, from 0 to 0.8 volts, and high, from 2.4 to 5 volts. In order for the computer to use the information
provided by the pitch detector, the pitch detector's signals must first be converted to this binary, TTL language and organized in a meaningful manner. It therefore becomes necessary to build a unit which is capable of interfacing these two dissimilar devices.

The Pitch Detector Interface

Terminology

A schematic diagram of the electronic interface designed by this writer is shown in Fig. 1. Before the theory of operation of this interface can be discussed, however, it is necessary to define some terms commonly used in reference to computers and electronics.

Almost all computers are designed to operate with binary data. This means that a signal line can be in only one of two possible states at a time: high and low, sometimes referred to as on and off. The exact voltages representing these states depends upon the physical integrated circuit devices used. The voltages representing these states in TTL circuits, probably the most commonly used in computers, are fixed at the levels discussed above. Other types of logic exist, however. Complementary metal-oxide semiconductor (CMOS) devices, for example, may operate at a supply voltage of from three to fifteen volts. The low state is always close to zero volts while the high state is always slightly under the supply voltage. Both types of devices, TTL and CMOS, are used in the pitch detector/computer interface.

Regardless of the device type used the low state of a signal line may be used to represent the numeral zero while the high state may represent the numeral one. This signal line, representing one binary
digit, is called a "bit." Bits are usually grouped together in a logical unit of eight and, as such, are referred to as a "byte." Since each individual bit may be either high or low, it can be shown that there are 256 possible combinations of the states of these bits within one byte. It is the byte which is the fundamental unit of information in most computers.

A computer can communicate with external devices through a hardware communication link referred to as a "port." A port is said to be either serial or parallel depending upon the manner in which data is transferred between the two devices. A serial port has only one signal line for the transmission of binary data in one direction. A parallel port has several, usually eight, separate signal lines over which binary data may be passed. The pitch detector interface is designed to connect to the computer through an eight-bit parallel port.

**Theory of operation**

The pitch detector/computer interface is basically a frequency counter; it simply counts the number of pulses in the input signal for a fixed amount of time and stores that number until the next count is finished. For example, if the input frequency is 440 Hz and the counter operates for one second, then the counter will have counted 440 pulses at the end of the one-second period. That count is then stored, replacing any previous count, and the cycle can be repeated.

One second is too long a period to wait before the frequency of the signal can be known. It is desirable to display the pitch of the tone as quickly as possible, preferably as quickly as the ear is capable of perceiving the pitch. To remedy this situation it is
necessary to count the pulses of the input signal for a much shorter period of time. Doing so without adjusting the frequency of the input signal, however, yields a pulse count that does not represent the frequency. For example, if the input signal is 100 Hz and the pulses are counted every 1/10 second, the resultant count will be 10. While the count is a function of the input frequency, it is more desirable that the count not have to be adjusted.

In the interface designed for this project this problem is solved by multiplying the input frequency by 100 and counting the pulses of this frequency for 1/100 second. This scheme yields a pulse count that exactly represents the frequency of the input signal and does so at a rate that is fast enough to be useful. As currently implemented the counter requires 1/100 second to store the pulse count and reset the counter to zero in preparation for the next count cycle. The actual cycle time, therefore, is 1/50 second: 1/100 second for these "housekeeping" duties and 1/100 second for counting the pulses of the signal.

At any time the computer is capable of reading the latest count available through the eight-bit parallel port to which the interface is connected. As discussed above eight bits are capable of 256 combinations representing any count from 0 to 255. Unfortunately, an upper frequency limit of 255 Hz, corresponding to a flat middle C (c^1), is inadequate for the range of frequencies expected. If the highest pitch expected to be sung is c^3 then it is necessary to be able to count to at least 1047, the frequency representing this pitch.

The addition of one bit to any logical group of bits effectively doubles the number of combinations of states of the bits. One bit has
two states, two bits have four states, three bits have eight states, etc. Therefore, eleven bits, capable of 2048 combinations, are needed to exceed the minimum count of 1047. While ten bits, allowing 1024 combinations, are very close to this minimum count in practice it is no more difficult to add four bits to the counter than it is to add only one. The interface employs, therefore, a twelve-bit counter capable of counting to a frequency of 4096 Hz, roughly corresponding to the pitch c⁵.

Since the parallel port to which the interface is attached has only eight bits, the use of a twelve-bit counter presents another problem. In order to pass all twelve counter bits through the eight-bit port it will be necessary to do so in two steps. As currently implemented, the interface requires that the computer read the low-order eight bits of the counter in one cycle and the remaining four bits in another cycle. The interface hardware is designed to allow the computer to select either portion of the data as desired.

Schematic diagram description

The electronic schematic diagram (Fig. 1) of the pitch detector interface is divided into six logical sections by dashed lines. The following paragraphs briefly describe the function and operation of each of these sections.

Section A takes the fundamental frequency pulse wave provided by the pitch detector and conditions it to be compatible with the complementary metal-oxide semiconductor (CMOS) devices to which it is directed. The pulse wave from the pitch detector alternates between +10 volts and -10 volts. The level of this signal is shifted to +15
Section A
Input conditioning circuit

Section B
Frequency multiplier

Fig. 1. Electronic schematic diagram of pitch detector interface
Section C, part 2
Frequency counter

GOOD PITCH
GATE

Section C, part 2
Frequency counter

LATCH

FREQ X 100

COUNT

CLEAR

SELECT

Fig. 1--Continued
volts and 0 volts making it suitable for the CMOS devices in section B.

The function of section B is to multiply the frequency of the given signal by a factor of 100. This operation is accomplished by using a phase-locked loop in combination with a divide-by-100 circuit. The phase-locked loop device delivers a signal at the same frequency as the input signal by continuously comparing the frequency of the two signals. This output signal is routed back into the device in a feedback loop in order for the comparison to be made. As the frequency of the input signal changes, the device senses the change and adjusts the frequency of the output signal until the two signals are again of the same frequency.

In order to force this circuit to function as a frequency multiplier it is necessary to break the feedback loop of the output signal and route it to a divider circuit. The divider circuit puts out one pulse for every 100 that it counts in its input signal, thus dividing the frequency of the signal by 100. The output of this divider circuit is then routed back to the phase-locked loop device to complete the feedback loop. The phase-locked loop is then forced to increase the frequency of its output signal by a factor of 100 in order to produce a frequency in the feedback loop that matches that of the input signal. The level of this output signal is then shifted to a level that is compatible with the TTL devices to which it is then directed.

Section C forms the frequency counter portion of the interface. The three four-bit counters (74LS161) shown in the diagram are cascaded in a manner to function as one twelve-bit counter. These
devices simply count the number of pulses in the signal provided by the frequency multiplier circuit of section B.

In order to be useful the counter must be stopped at regular intervals in order to store the count and reset the counter to zero for the next count cycle. This is the function of the frequency counter time base integrated circuit (7207). This time base device provides all of the control signals needed to initiate the count cycle, stop the counting period after 1/100 second, store the resultant count in the octal latch devices (74LS374), and reset the counter to zero in preparation for the next counting period. Also provided by this device is a signal that is routed to a status line on the computer port which indicates when the counting cycle has been completed.

The function of section D is to manage the task of passing the twelve-bit count over the eight-bit port in two steps. This circuit places either the first eight bits or the upper four bits of the count on the port depending upon the state of a single signal provided by the computer. If this selection signal is low the first eight bits of the count are placed on the port; if the signal is high the upper four bits of the count are placed on the port leaving the other four bits unused.

Section E monitors the status of the "Gate" and "Good Pitch" signals provided by the fundamental frequency extractor hardware as described above. Both of these signals should be high during normal operation. If either signal goes low then the frequency extracted during that time should probably be viewed with suspicion. These two signals, therefore, are passed to the computer for evaluation on two of the four unused bits in the upper portion of the frequency count.
The Computer

The microcomputer for which this project was implemented is the **Sorcerer** built by Exidy, Incorporated. This computer is well suited to the needs of this study for several reasons. It is an inexpensive computer with excellent graphic capabilities and a standard input/output port for connection to the pitch detector interface. In addition both the Louisiana State University School of Music (where this project was formally tested) and this writer each currently own one of these computers making it readily available for the development of this project.

It is unfortunate that during the course of this project the manufacturer of the computer went out of business thereby ending the future availability of the computer. It is also recognized that since the **Sorcerer** was not well marketed it is neither a very well known nor widely used computer. The popularity and success of a computer cannot be predicted with certainty. One can only prepare for changes in the marketing potential of a computer by adapting a project for several of the available computers. Fortunately, the goals of this project are not contingent upon the popularity of a particular computer. It is first necessary to demonstrate the feasibility and value of the sound-to-notation translator. When these demonstrations prove successful the project may be adapted for implementation on other computers.

The **Sorcerer** is based on the Z-80 processor, one of the most common microprocessors in use today. There are many computer manufacturers who have marketed computers based on this processor and it should be a straight-forward matter to adapt the programming of this project to most of these computers. There will be, however, many
compatibility problems, especially in terms of graphic capabilities, which will interfere with the implementation of the project on some computers. This problem is a function of hardware realizations by different manufacturers and cannot be avoided.

Though many of the features of the Sorcerer were impressive at the time it was designed, by 1984 standards they are considered to be conservative and non-standard. The Z-80 processor, for example, at the present time is driven by a clock running at rates of 4 MHz, 6 MHz, and even 8 MHz in some computers, while in the Sorcerer an older version of the processor is driven at the conservative rate of 2.1 MHz. Transferring software from the Sorcerer to some other computer based on a faster version of the processor results in a very significant increase in processing speed. In the present project the time required to display a pitch upon detection can be reduced by such a transfer.

The Sorcerer is designed to display a screen image consisting of 30 lines, each with a maximum length of 64 characters. Each character is formed within an array of 64 pixels arranged in an 8 by 8 matrix. A pixel is the smallest point of light that a video screen is capable of displaying. There are 256 separate and independent characters which can be displayed on the screen at any time. Half of these, including the upper and lower case letters, numerals, and other special characters, are fixed and cannot be changed. The remaining 128 characters may be defined by the programmer who is given control over each separate pixel in the character image.

The Sorcerer computers on which the project was implemented and tested are each supplied with 48K bytes of random-access memory (RAM).
RAM is memory that may be both read and written by a program. One "K," as it is most often used in computer terminology, is equal to 1024 and memory is usually measured in units of "K" bytes. One "K" of memory, therefore, normally refers to 1024 bytes.

A "ROM PAC" cartridge may be inserted into the Sorcerer for the purpose of executing the program contained within the cartridge. As its name implies, ROM (read-only memory) is memory that may be read from but not written to. Frequently-used programs are often stored in ROM for convenience in use.

The only secondary storage medium available to the Sorcerer on which this project was implemented is that of cassette tape. While it is an inexpensive solution to the need for mass storage, cassette tape is inherently slow and error prone. The use of cassette tape may be justified for small storage needs but for the storage of large programs and files it becomes intolerable. Consequently, the program developed for the present project was stored in a ROM cartridge that could be inserted into the Sorcerer and be available for use as soon as power was applied to the computer. Since the program required approximately 16K bytes of memory and the ROM cartridges developed for the Sorcerer were designed to hold only 8K bytes, the cartridge was modified by this writer to accommodate the additional capacity.

Other Hardware

In order for a computer to produce sound an electronic device is needed to convert the numbers provided by the computer to a corresponding voltage. As the numbers provided by the computer become larger, the voltage produced rises correspondingly. The computer may then send a string of numbers, representing any waveform and evenly
spaced in time, to this device whose output may then be directed to a conventional loudspeaker. A device which performs this function is called a digital-to-analog converter (DAC) and this is the technique used in the present project for the production of sound.

The design for the digital-to-analog converter used in this project is one provided by Ciarcia.1 As designed, this device takes an eight-bit value supplied by the computer at a parallel port for conversion. Since one bit was required in the pitch detector interface for the selection of the portions of the frequency value being passed to the computer, the digital-to-analog converter was slightly modified to use only the upper seven bits leaving the remaining low-order bit free for this selection function.

The output of the digital-to-analog converter must be amplified before it can be routed to a loudspeaker. A conventional stereo integrated amplifier was used in the present study for this purpose. The amplifier used supplies a connection for the use of headphones. Since the computer was not installed in a separate room dedicated for its use, the use of headphones was advantageous.

Software Implementation

General Description

In order to make this project suitable for real-time use it is necessary to choose a computer programming language which facilitates high speed processing. For this reason, programming at the assembly language level was selected, since it is at this level that the highest degree of speed can be attained. While it would be possible to use a higher level language in some of the functions in which speed
is not critical, it is probably best that the entire project be written in assembly language in order to minimize problems of compatibility with other computer systems. Since the computer on which the program is implemented is based on the Z-80 processor, the program is written in Z-80 assembly language.

The processing tasks of the present project were divided into two large programs: the Music Control Program (MCP) and the Sight-singer's Mirror (SSM) program. MCP is a music operating system whose purpose is to coordinate and standardize the execution of ordinary music functions such as displaying staves, clefs, key signatures, notes, accidentals, bar lines and performing other duties associated with these tasks. SSM is the program which uses MCP to perform the tasks specifically associated with a sightsinging student's session such as selecting melodies, listening to the student sing them, evaluating the student's performance, writing records of session statistics to tape, and other such functions. Both of these programs are discussed separately below.

In order to avoid duplication of effort and to minimize the size of the programs, both MCP and SSM take advantage of sub-programs which are always present in the Sorcerer computer for the performance of low-level tasks. Included among these tasks are those for displaying characters on the screen, reading characters typed at the keyboard, displaying a string of text on the screen, writing records to cassette tape, and reading records from tape. These sub-programs, or routines, are part of the Sorcerer Monitor program which is stored in ROM and is normally used for the control of the computer.
Music Control Program

MCP is a set of small programs, or subroutines, with one common entry point. Any program which uses MCP must always transfer control to it at the same position in the program and supply it with a value which indicates which of the fifty-three distinct functions are to be performed. Many of these functions, often referred to as "system calls," require additional parameters which must be supplied by the calling program. When the function is completed MCP returns control to the calling program. A list of all fifty-three system calls with a brief description of each is shown in table 1. These system calls may be organized into eight groups each of which will be described in more detail. A listing of the program is given in appendix A.

Video screen control. The video screen may be divided into as many as sixteen logical sections called partitions. A partition may contain any number of lines of any length and may be placed at any position on the screen provided the entire partition can fit on the screen. It may be desired, for example, to divide the whole screen into two independent logical screens for two different purposes. Data may then be displayed and erased in one partition without affecting the other partition.

There are four system calls provided for the definition and maintenance of these partitions by a program. System call 0 is used to define the dimensions of any partition. To define a partition the program must supply the following parameters: (1) the number of the partition being defined, (2) the screen row and column position where the top left-hand corner of the partition is to be placed, (3) the
### TABLE 1

**MCP SYSTEM CALLS**

<table>
<thead>
<tr>
<th>Function Code</th>
<th>Function Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Defines a screen partition</td>
</tr>
<tr>
<td>1</td>
<td>Activates a screen partition</td>
</tr>
<tr>
<td>2</td>
<td>Erases all of a screen partition</td>
</tr>
<tr>
<td>3</td>
<td>Returns the number of the active partition</td>
</tr>
<tr>
<td>4</td>
<td>Displays a staff and clef in the active partition</td>
</tr>
<tr>
<td>5</td>
<td>Displays only a staff in the active partition</td>
</tr>
<tr>
<td>6</td>
<td>Displays a clef on the staff in active partition</td>
</tr>
<tr>
<td>7</td>
<td>Erases the staff from the current position and closes it with either a single or double bar line</td>
</tr>
<tr>
<td>8</td>
<td>Displays a bar line at the current position on the staff in the active partition</td>
</tr>
<tr>
<td>9</td>
<td>Sets the key</td>
</tr>
<tr>
<td>10</td>
<td>Displays the key signature for the current key</td>
</tr>
<tr>
<td>11</td>
<td>Returns the number and type of accidentals for the current key</td>
</tr>
<tr>
<td>12</td>
<td>Resets fields indicating the key and transposition</td>
</tr>
<tr>
<td>13</td>
<td>Displays the time signature</td>
</tr>
<tr>
<td>14</td>
<td>Sets transposition indicators</td>
</tr>
<tr>
<td>15</td>
<td>Displays an entire melody in the screen partitions given</td>
</tr>
<tr>
<td>16</td>
<td>Translates a melody to the form required by the subroutines which play it</td>
</tr>
<tr>
<td>17</td>
<td>Displays a melody in the partitions given until all of the partitions are filled</td>
</tr>
<tr>
<td>18</td>
<td>Displays a melody in the active partition until the end of the staff is reached</td>
</tr>
<tr>
<td>19</td>
<td>Displays a melody in the active partition until the end of the measure is reached</td>
</tr>
<tr>
<td>20</td>
<td>Displays a pitch when given its scale degree and octave transposing if required</td>
</tr>
<tr>
<td>21</td>
<td>Displays a pitch when given its letter name, accidental and octave transposing if required</td>
</tr>
<tr>
<td>22</td>
<td>Displays a pitch when given its unadjusted form as used by MCP</td>
</tr>
<tr>
<td>23</td>
<td>Displays a pitch when given its integer form as used by the subroutines which play it</td>
</tr>
<tr>
<td>24</td>
<td>Erases the previously displayed pitch and backs up the current position</td>
</tr>
<tr>
<td>25</td>
<td>Erases the given number of columns in the active partition without affecting the current position</td>
</tr>
<tr>
<td>26</td>
<td>Displays the given accidental</td>
</tr>
<tr>
<td>27</td>
<td>Transposes a given letter name and accidental in the original key to the new key</td>
</tr>
<tr>
<td>28</td>
<td>Converts a scale degree to a letter name in the current key</td>
</tr>
<tr>
<td>Function Code</td>
<td>Function Performed</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>29</td>
<td>Converts a letter name to the scale degree in the current key</td>
</tr>
<tr>
<td>30</td>
<td>Converts a letter name to the absolute pitch form</td>
</tr>
<tr>
<td>31</td>
<td>Converts an unadjusted pitch to a letter name and accidental in the current key</td>
</tr>
<tr>
<td>32</td>
<td>Converts an absolute pitch to an integer pitch</td>
</tr>
<tr>
<td>33</td>
<td>Converts an integer pitch to an unadjusted pitch</td>
</tr>
<tr>
<td>34</td>
<td>Translates and plays the melody currently in the tune table</td>
</tr>
<tr>
<td>35</td>
<td>Aurally establishes the current key</td>
</tr>
<tr>
<td>36</td>
<td>Plays pitches relative to a given offset</td>
</tr>
<tr>
<td>37</td>
<td>Plays the pitches stored at the address given</td>
</tr>
<tr>
<td>38</td>
<td>Establishes a transposition offset for all sound output</td>
</tr>
<tr>
<td>39</td>
<td>Returns the next valid pitch that is heard</td>
</tr>
<tr>
<td>40</td>
<td>Returns the average of four consecutive pitch samples and the average deviation factor</td>
</tr>
<tr>
<td>41</td>
<td>Returns the pitch of the next frequency heard</td>
</tr>
<tr>
<td>42</td>
<td>Returns the next frequency heard</td>
</tr>
<tr>
<td>43</td>
<td>Converts a frequency to its pitch</td>
</tr>
<tr>
<td>44</td>
<td>Initializes the input pitch fields</td>
</tr>
<tr>
<td>45</td>
<td>Returns the number of half steps between two pitches</td>
</tr>
<tr>
<td>46</td>
<td>Returns the highest and lowest pitches in a melody</td>
</tr>
<tr>
<td>47</td>
<td>Returns the accidental in the key signature for a given pitch</td>
</tr>
<tr>
<td>48</td>
<td>Sets the current position on the staff in the active partition</td>
</tr>
<tr>
<td>49</td>
<td>Returns the current position on the staff in the active partition</td>
</tr>
<tr>
<td>50</td>
<td>Inhibits the storing of pitches in the tune table</td>
</tr>
<tr>
<td>51</td>
<td>Enables the storing of pitches in the tune table</td>
</tr>
<tr>
<td>52</td>
<td>Displays text in the active partition</td>
</tr>
</tbody>
</table>

length of the partition line, (4) the number of lines in the partition, and (5) the line number from the top of the partition on which the top line of a staff is to be displayed.

When a system call is executed for the purpose of displaying a clef, a key signature, or a note on a staff and more than one staff is displayed on the screen, MCP must have a way to decide which staff is
the intended object of the system call. This problem is resolved by always performing such functions on the staff in the active partition. Only one partition may be active at a time and it is the programmer's responsibility to activate the partition in which subsequent system calls are to act. Any partition which has previously been defined may be activated by supplying the number of the partition to MCP through system call 1.

System call 2 provides a means by which the contents of a partition may be erased. Regardless of its position on the screen, a partition can be cleared without affecting any other part of the screen. System call 3 allows the program to inquire of MCP the number of the partition that is currently active.

**Music symbol display.** There are twelve system calls provided for the display of standard music symbols. System call 4, for example, will clear the active partition, display a staff in that partition, and place the desired clef at the beginning of the staff. The calling program must supply the line number from the top of the partition where the top line of the staff is to be placed and a code representing the clef to be displayed. The staff and clef may be displayed independently if desired through system calls 5 and 6.

If only a portion of a staff is used the remainder may be erased by using system call 7. The calling program need only supply the type of bar line, single or double, to be placed on the end of the shortened staff. A single bar line may be placed on the staff at any time without affecting its length through system call 8.

Notes may be displayed in any of four ways: (1) by scale degree, scale degree modifier (raised or lowered), and octave, (2) by letter
name, accidental, and octave, (3) by the unadjusted pitch code as used internally by MCP, and (4) by an integer representing the absolute pitch. If transposition indicators have been previously set, as will be explained below, these pitches will be transposed to the new key prior to display. Each of these pitch display methods deserve further discussion.

To display a note by scale degree the calling program must supply five parameters: the scale degree, the scale degree modifier, the octave, the rhythmic value, and the current position increment. The scale degree may be any value between one and seven inclusive, representing the degree of the scale in the current key.

The modifier is a value between -2 and +2 which indicates the amount and direction that a scale degree is being altered. A value of 1, for example, indicates that the scale degree is to be raised one-half step. Minus one indicates that the scale degree is to be lowered one-half step. A value of zero is used when the scale degree is not to be altered.

The octave specifier departs from conventional practice to facilitate ease in programming. An octave indicator of zero refers to the pitches between small a and g-sharp¹ inclusive. Octave one refers to the pitches a¹ to g-sharp², while octave minus one refers to the pitches between great A and small g-sharp. Pitches in the remaining octaves are numbered similarly with negative octaves below small a and positive octaves above.

Since rhythm has not yet been implemented in MCP the only rhythmic value which can be displayed is the whole note, currently
represented by any non-zero value. A value of zero causes the filled-in whole note to be displayed.

Associated with each partition is a current position field which contains the column position in which the next symbol is to be displayed. This current position is adjusted each time a symbol is displayed so that subsequent symbols will be properly spaced. The current position increment is a value that will be added to the current position after the note is displayed and thus represents the number of columns to be left vacant following the pitch.

Displaying a note by letter name is very similar to the display by scale degree since three of the parameters which must be supplied are the same in both cases: the octave, the rhythm, and the current position increment. Instead of the scale degree and modifier, however, the letter name and accidental are provided.

The letter name may be any value between 65 and 71, the ASCII (American Standard Code for Information Interchange) value for the letters A through G, respectively. ASCII codes are numeric equivalents for a set of displayable characters, including upper and lower case letters and special symbols, and non-displayable control characters.

The accidental is indicated by a value between -2 and +2 inclusive. A natural is indicated by a value of zero, a sharp by positive one, a double sharp by positive two, a flat by minus one, and a double flat by minus two.

In order to conserve memory space MCP condenses the pitch information into a value which occupies less space than the storage of the scale degree or letter name scheme requires. Most of the manipula-
tions with a pitch are performed internally by MCP with the pitch in this condensed form called the unadjusted pitch. When the pitch is displayed it is adjusted for the octave and clef in effect as well as the position of the staff on the screen. The system call which provides for the display of a pitch by its unadjusted value is provided for only those programmers who are familiar with this scheme.

In some cases it is necessary to refer to a pitch in absolute terms. When a pitch is heard, for example, its spelling is not known until one considers the key in which the pitch occurs and possibly the following pitch as well. In these cases the pitch is referred to by one integer regardless of the number of ways in which that pitch may be spelled. Provided the key has been set pitches may then be displayed by supplying an integer representing the pitch desired, as well as the rhythm and current position increment.

In each of these system calls for displaying a note any accidental is automatically displayed if it is required. If an accidental is already in effect from a note occurring earlier in a measure, this condition is detected and the redundant accidental is not displayed. A system call is provided, however, for the independent display of accidentals.

Key control. There are six system calls provided for the manipulation of the key. One of these system calls, system call 9, identifies and validates the key. For this system call the calling program must supply the key letter name in ASCII, the accidental modifying that letter name, and the mode. The accidental may only be minus one for flat, zero for natural, or positive one for sharp. The mode may
be only one of two values: zero for major or one for minor. This system call also verifies that the specified key is a valid key.

Once the key has been set the key signature may be displayed through system call 10. No parameters need to be supplied for this system call since all of the information that it needs is assumed to have already been provided. This system call determines the type of accidental that is needed for the key signature, the number of these accidentals, and their placement on the staff regardless of the clef in use.

Two other useful system calls are provided to supply information concerning keys. System call 11 returns the type of accidental and the number of these accidentals in a given key. System call 47 returns the accidental in effect by the key signature for a given pitch.

A means of transposing from one key to another is provided by system call 14. The calling program must supply the letter name and accidental of the key to which pitches are being transposed. The mode is assumed to be the same as the current key. In addition, the calling program may indicate the specific octave to which the pitches will be transposed. For example, the transposition of the key of C major to D major may be accomplished by transposing up a major second, down a minor seventh, or either of these plus one or more octaves.

**Melody display and manipulation.** Melodies may be displayed in whole or in part through the use of system calls 15 through 19. To display a whole melody the calling program needs to supply two memory addresses to system call 15: one address for the beginning position of the melody and one for the start of a list of screen partition
numbers to be used. System call 15 will display the staff, clef, key signature, and melody in the first partition given until it is filled. It then continues the melody in the next partition given until either the whole melody is displayed or all of the given partitions are filled. If desired the calling program may display a melody until the end of the staff is reached or the end of the measure is found through the use of system calls 18 and 19, respectively.

As pitches are displayed they are normally stored in a table by MCP for its own internal use. This function is automatically performed by MCP and the programmer need not be aware of it. Once a melody has been displayed through system call 15 and thus stored in the tune table it may be redisplayed, if necessary, by system call 18. The only difference between these two system calls is that the former operates on a melody supplied by the calling program and the latter operates on the melody that is currently stored in the tune table and is slightly faster.

System call 16 will translate a melody provided by the calling program into the format used by MCP and store it in the tune table without displaying it. If it is desired to display a melody without disturbing a different melody that is stored in the tune table, system call 50 will prevent the storage of the new melody in the tune table. This feature can be re-enabled by system call 51. The only remaining system call dealing with melodies is number 46 which returns the highest and lowest pitches that occur in a melody.

**Sound output.** There are five system calls for the production of sound. The simplest of these, system call 35, plays the ascending and descending scale of the current key followed by the four-part chord
progression: tonic, subdominant, dominant seventh, tonic. This is useful for aurally establishing a key for the user. No parameters need be supplied for the performance of this system call.

All pitches that are displayed and thus stored in the tune table may be played through system call 34. The calling program supplies only one parameter, that of the tempo to be used for playback. Entire melodies may be played in this manner.

There are two system calls provided for playing material supplied by the calling program. By supplying a tempo and a memory address where the material to be played is stored, the calling program may have MCP play an exercise of up to four voices. The length of the exercise is limited only by the amount of storage available. The two system calls provided for this purpose differ only in the manner in which they treat the pitch data. System call 37 plays the absolute pitches exactly as they are stored. System call 36 adds the stored pitches to a given root pitch yielding, in effect, an aural transposition of the exercise.

System call 38 provides a means of transposing aural pitches independently of the key in which they are written. A numeric value must be specified representing the distance in half steps that the pitches are to be adjusted. Positive values transpose the sounded pitches upward; negative values transpose downward. An offset of +1, for example, effects an aural transposition up one half step while an offset of -2 causes a transposition down one whole step. This facility is provided so that material that is displayed in one key may be played in another without transposing it. What is gained is the ability to allow users to sing a melody in a key that is different
from the one displayed, thereby allowing the placement of the melody in the user's vocal range.

**Sound input.** At the lowest level of the system calls for sound input is system call 42 which returns the frequency provided by the pitch detector interface at the parallel port to which it is attached. This frequency may then be converted to a pitch by using system call 43. Both of these functions may be performed in one step through system call 41. The remaining two system calls of this group deserve a more detailed discussion.

System call 40 stands at the next hierarchic level above these system calls in that it uses system call 41 to get two pitch samples which are then averaged with the preceding two pitch samples. These pitches are arithmetically averaged in order to cancel the effect of instability in the frequencies of the pitch being sung. For example, if the four extracted pitch samples are represented by the values 24, 24, 23, and 24, the rounded average, 24, eliminates the effect of the sample that is one-half step lower. On the other hand, if the four pitch samples are 10, 20, 30, and 40, the average of 25 should not be interpreted as a stable pitch. For this reason an average deviation factor is also calculated to indicate the level of stability in the pitches.

The average deviation factor is calculated by finding the absolute value of the difference between each pitch sample and the average of the four samples. All four of these differences are then summed to yield the average deviation factor. In the first case of the four pitch samples mentioned above, for example, the average deviation factor would be one, indicating a high degree of stability. Only one
of the four pitch samples in this case deviates from the average of 24 and its difference is only a value of one. In the second case, however, each pitch sample is vastly different from the average of 25. The pitch samples, taken in the order given, have a deviation from the average of 15, 5, 5, and 15. The average deviation factor of 40, calculated by summing these four figures, indicates a high degree of instability.

System call 39 uses system call 40 to determine when a pitch has been stable for the minimum length of time to be considered a valid pitch. When system call 40 returns the same pitch average for a fixed number of consecutive times, the pitch is assumed to be valid. This number was determined by experimentation and is currently set to a value of six. Since one pitch average returned by system call 40 represents four frequency samples, two of which overlap with the adjacent pitch average, the total minimum number of frequency samples that must be read before a valid pitch can be found is fourteen. With the frequency samples being provided at a rate of 50 per second, these fourteen samples translate into a time period of approximately one-fourth second.

Conversion routines. Since a pitch may be represented in several ways it often becomes necessary to convert a pitch represented in one form to another form. For example, a pitch represented by a scale degree may need to be converted to its actual letter name and accidental. The reverse conversion may also be necessary. There are several routines provided in MCP for this purpose. Routines for the conversions just mentioned, for example, are found in system calls 28 and 29. Also included in this group is a system call for transposing
a pitch letter name to another key. Other less commonly used conversion routines are also provided.

**Miscellaneous routines.** There are currently three system calls which do not fit neatly into any one of the preceding system call groups. System calls 48 and 49 may be used to set and return, respectively, the current staff position in the active screen partition. Another system call, number 52, is provided to facilitate the display of text within a partition.

**Sightsinger's Mirror**

Though MCP is obviously designed for the control of music functions there is no one specific music application for which it is intended. Indeed, its value lies in its generality making it suitable for a variety of music applications. MCP is not independent; another program is required before the value of MCP can be uncovered. In the current project this controlling program is Sightsinger's Mirror (SSM), another Z-80 assembly language program which utilizes the functions of MCP to implement the specific application of sight-singing.

Because they are not designed to be performed independently the subroutines of SSM cannot be as clearly divided into groups as in MCP. The functions of MCP serve as temporary sub-processes of a controlling program which is expected to control and organize these functions in a meaningful way. The subroutines of SSM, therefore, will be discussed in the general order in which they are likely to be executed. The actual processing path, however, is subject to many factors, one of which is the unpredictable nature of the needs and requests of the user.
A listing of the SSM assembly language program is given in appendix B. Continuing the standard established in the MCP listing, each subroutine of the program is marked by its name and textual description. The cryptic nature of many of the names of the subroutines is due to the six-character limitation on names as practiced by most assemblers for the Z-80 processor. References to the subroutines will be made with these names.

One of the features of SSM is that of automatically writing user session statistics records to cassette tape. If these records are to be evaluated then there must be a means of preventing the records associated with one user from being confused with the records of another user. This problem is solved by assigning each user a unique name which is used as the tape file name under which his records are written. All of the records that have been written to cassette tape may then be sorted by this tape file name thus ensuring that all records are properly associated with a specific user.

There is also another reason why each user has been assigned a unique name by which he is identified to the computer system. Since an experiment was performed to test the effectiveness of the present project it was necessary to ensure that the students who used the system were authorized to do so.

The participants in the experiment were divided into two groups: those who were to use the system for sightsinging practice and the control group of those who were not to use the system. The data collected from the experiment would be adversely affected if members of the control group were given access to the sightsinging system. Consequently, a procedure was implemented in SSM to force each user to
identify himself with his assigned system-user name. The term "logon" is often used as both a verb and a noun in reference to this type of procedure. Only those names which were properly assigned by this writer were recognized by this logon procedure program of SSM. In addition, a password was associated with each user name to make an unauthorized access attempt more difficult. Only after a recognized user name and its associated password were supplied by a user was he allowed to proceed.

Since the user's system name was used as a cassette tape file name any restrictions applying to the names of tape files applied also to the user names. As implemented on the Sorcerer computer, SSM uses routines which reside in the ROM-based monitor program of the computer to transfer these records to tape. These Sorcerer routines restrict tape file names to five characters, thus SSM user identification names were also similarly restricted.

When the SSM program is started it performs some system environment configuration tasks in the subroutine "COLD" and then asks for the date in the "GETDAT" subroutine. Since the Sorcerer computer on which this project was implemented was not equipped with an electronic clock and calendar, the operator who initiates the program is required to supply the program with the date. This date is one of the data elements included in one of the records written to cassette tape.

A list of authorized user names and passwords must then be loaded from cassette tape. This task is performed by the "LODUSR" subroutine which guides the operator through the steps necessary for loading this tape file. A library of melodies, as described in chapter 1, may also be similarly loaded from cassette tape under the control of "SETTG."
Once these tasks are completed the system is ready for use by a sightsinging student.

The "LOGON" subroutine waits for a user to type his system name. When this happens the subroutine checks to see if that name is included in the list of authorized names previously loaded from tape. If the name is included the user is asked to type his password. The password typed by the user must match the password in the list associated with the user's system name. If the name or password are found to be invalid the subroutine displays a message telling the user that he is not an authorized user. Otherwise, control is passed to the "GETRNG" subroutine.

The "GETRNG" subroutine asks the user to sing the highest and lowest pitches which he feels comfortable singing. Melodies which are to be sung are aurally transposed to a key that is centered between these two pitches for the user's convenience.

The user is then given a choice of two modes of sightsinging practice. In "Free" mode the user is given a blank staff on which is displayed any series of pitches he desires to sing. In "Tunes" mode melodies are displayed for the user to sing and his performance is evaluated and graded. The subroutine "CTLFRE" controls the free mode of operation while "CTLTUN" controls the tune mode.

"CTLFRE" uses "QRYFRE" to ask the user for his choice of clef and key signature. Control is then passed to "FREMID" which controls the display of the pitches the user sings.

"CTLTUN" first shows the user the name of the melody library that is loaded, if any. If the user decides to load a different library of melodies, "SETTGG" guides him in the steps necessary for doing so.
"CTLTUN" then passes control to "SELTUN" which selects a melody from the library to display.

There are three schemes which may be used for the selection of melodies from a library. The simplest of these is the linear scheme in which melodies are selected in the order in which they occur in the library. A second method of selection provided is that of random selection. In the third scheme a melody is selected based upon its level of difficulty and the current ability of the user. Associated with each melody library is a field which indicates the scheme to be used for selecting melodies. "SELTUN" checks this selection indicator field and then passes control to appropriate melody selection subroutine: "LINSEL," "RNDSEL," or "RATSEL."

When a melody has been selected for use a single bit that is reserved for the purpose is set to a value of one to indicate that the melody has been used. This bit setting prevents the same melody from being selected again. If a user sings every melody in a library these bits are reset so that the melodies may be re-selected. These bits are also reset for the next user when a user finishes his practice session.

"LISTUN" controls the display of the selected melody and asks the user to identify the key. Once the key is identified a scale and four-part chord progression are played to aurally establish the key. The user is then asked to sing the tonic triad before continuing. Once this task is accomplished the "SINGIT" subroutine is given control in order to listen to the user's performance of the melody and display the pitches he sings. When the user finishes the melody
"EVALUT" evaluates the performance and displays a percentage figure representing his performance accuracy.

The subroutine "CREREC" is periodically performed to create a record to be written to cassette tape. This subroutine is performed once when a user starts a session, once for each time that a melody library is selected, once for each melody that is selected, and once for each performance attempt of a melody. These four different types of records are each created by four different subroutines: "USEREC," "GRPREC," "TUNREC," and "PERREC," respectively.

These records are always written to a block of memory called a buffer. This buffer is 256 bytes of memory storage in size. It is only when this buffer is filled with records that the contents of the buffer are written to cassette tape by the subroutine "WRTBUF." When this occurs the user's session must be halted for the few seconds that are required for this transfer. The buffer is then considered to be empty and more records may then be created in the same manner.

While SSM is made up of many more subroutines than have been briefly discussed in the preceding paragraphs, most of them perform low-level tasks and are subservient to the subroutines whose functions were described. For the programmer who is interested in the details of these subordinate subroutines, it is felt that the documentation provided for each subroutine in the listing of appendix B is adequate for an understanding of their functions.
Notes

CHAPTER III
THE USER'S VIEW OF THE SYSTEM

Introduction

A sightsinging student who desires to use the system developed in this project should not need to know about how the computer was programmed, the techniques and algorithms used, or the specifications of the computer and other hardware. It makes little difference if spectacular and clever techniques were used in the development of a product if the final product as a whole has no useful value. What is important is that the learning tool allow the student user to be concerned only with the matters of developing the skills for which the tool was developed.

To reap the benefits from the system the user must know how to respond to the queries of the program and how to control the program so that it performs the functions he desires. The less time required of the user to adapt to the protocols of the learning tool, the more time he has to use the tool to his advantage. It is possible that the system could be designed to be quickly and easily controlled by a skilled user, but such a design may necessitate the use of many cryptic commands which are difficult and time-consuming to learn. To make a system easy to learn, however, it may be necessary to provide more information at more frequent intervals for the user's response.
thereby using more time for protocol as a convenience to the user leaving less time for learning.

The use of menus is a very time-efficient technique for presenting a set of options to a user for his response. A menu is a list of available options each of which consists of a brief description of the function of the option and the means by which that option may be initiated. In the present project a one-line description of an option is paired with a single letter which when pressed on the keyboard performs the tasks of that option. An example of one such menu can be found below in Fig. 2 on page 61.

Different menus are presented to the user at various points in his session to inform him of the processing paths available to him. Since all users cannot be expected to be efficient typists the use of a single keystroke for performing all functions is advantageous. Furthermore, since an explanation of all options is displayed on the screen there is no need for the user to consult a written user's manual or commit the meanings of the options to memory.

Initiating the Session

Before the user can begin a session he must identify himself to the system. The system will indicate that it is ready for a user by displaying the "Username" prompt. A prompt is a cue or request for information given by the computer. When the "Username" prompt is displayed the user should respond by typing his assigned user name. If this name is included in the table of authorized user names, the program responds with a request for the user's password. A password may be any word or meaningless string of characters that is known only to the authorized user. Passwords are frequently used as a means of
preventing the attempted access of computer resources by unauthorized users.

As the user types his password the characters are not displayed on the screen in order to prevent others from seeing it. If the password given does not match the one associated with the user name the system responds with the message:

You are not authorized to use this system.

The "Username" prompt is then displayed again as before. If the user has simply made a typographical error he may start over again at this point.

When the responses to both prompts are found to be satisfactory the program attempts to establish the user's singing range. The user is first instructed to find the lowest pitch that he feels he can comfortably sing:

Find the LOWEST pitch that you can comfortably sing.
Press any key when ready.

When that pitch has been decided upon he may then press any key on the keyboard to continue. The program responds with:

Sing that pitch.

The user may then sing his low pitch into the microphone. When the program detects that pitch the message "Stop singing" is displayed and the same pitch is played back through the headphones. The program then asks the user to verify the pitch played back with the message:

Is this the right pitch? (Y/N).
The user responds with "Y" (Yes) if he is satisfied with the pitch or "N" (No) if not. In the latter case the program returns to the same request for his lowest pitch again.

When the lowest pitch is agreed upon the same process is performed so that the computer can obtain the highest comfortable pitch as well. If the user's range is less than an octave he is given a warning concerning his limited range but the program attempts to accomodate that range in any case. Once the user's range has been set the main menu is displayed.

The Main Menu

The main menu (Fig. 2) is the pivot point for all processes handled by the computer. From this menu the user may choose any of four possible actions: melody exercises, free mode, reset range, and exit. Each of these four options may be chosen by responding with the letter shown by each description. For example, if the user wishes to change his range he would respond with "R."

Fig. 2. The Main Menu

The following options are available:

'T' - Tunes
'F' - Free mode
'R' - Reset your Range
'X' - Exit

Select the letter of the option you would like:

A response of "R" returns the user to the same range finding process described above. When the user is finished with his practice session he would respond with "X." The "Free mode" option allow the user to sing any series of pitches he desires to see how those pitches
would be notated on the staff. The "Tunes" option tests the user's ability to sightsing melodies which are selected and displayed by the computer.

It should be noted that the "Tunes," "Free mode," and "Reset range" options each take the user to a process that will eventually return to this main menu. Therefore, the only means by which the user may end his practice session is through the "Exit" option on this menu.

The "Tunes" Option

When the "Tunes" option is selected from the main menu the computer program first determines whether it has a melody library to use. If not, a melody library will have to be loaded from cassette tape and the following message is displayed:

No tune group has been loaded.  
Do you want to load a tune group?  (Y/N)

A response of "N" (No) returns the user to the main menu. A response of "Y" (Yes) initiates the melody library loading process with the following message:

Place the desired tune group cassette tape in the tape player. Press any key when ready.

At this point the user may still return to the main menu by pressing the "ESC" (Escape) key. If he presses any other key the program leads him through the process of loading the melody library:

Press the PLAY button on the tape player.
After a few seconds another message will be displayed to indicate that the melody library has been found on the tape and is being loaded. When the library is finished loading the following message is displayed:

Tune group xxxxx is now loaded.
Do you want to use this tune group? (Y/N)

The "xxxxx" above is replaced with the name of the melody library that has been loaded. The melody library loading process may be repeated by responding with "N" (No). A response of "Y" (Yes) indicates that the user is satisfied with the melody library that is loaded and the program then proceeds to select a melody from this library.

When a melody has been selected it is displayed on the screen in standard music notation. Beneath this melody text is displayed asking the user to identify the key of the melody and brief instructions are given to show him how to respond (Fig. 3). To identify the key the user should type the key letter using upper case letters for major keys and lower case letters for minor keys. Sharps, if needed, may be

Fig. 3. Key identification message

What key is this tune in?

Major - Upper case
Minor - Lower case
Sharp - '#' or 's'
Flat - Double letter or 'f'

Example: c# - C sharp minor
       AA - A flat major

'? ' - Show the answer
ESC - Display option menu
indicated with either "#" or "s" following the key letter. Flats are specified with either "f" following the key letter or by repeating the key letter. If necessary, the user may have the key identified for him by responding with "?".

The "ESC" key may be used at this point to display the menu shown in Fig. 4 giving the user a means of aborting the melody that is displayed. The user may skip any melody he desires by responding with "S" causing the program to select another melody as before. The "R" response can be used to return to the same melody, while the "M" response takes the user back to the main menu.

Fig. 4. Menu for aborting melody

Would you like to:

'S' - Skip this tune
'R' - Return to where you were
'M' - Display main menu

Select the letter of the option you would like:

Once the key of a melody is correctly identified the program continues by playing the scale up and down followed by the chord progression: tonic, subdominant, dominant seventh, tonic. This scale and chord progression are played through the headphones for the user to aurally establish the key. The actual sounded pitch may not correspond with the written pitch since the range of each melody is centered within the user's range. While the written melody is not transposed, the pitches of the melody are transposed to a key that allows the melody to lie comfortably within the user's vocal range.

After the key-identifying scale and chord progression are played the user is asked to sing the tonic triad up and down (Fig. 5). As
the pitches of the triad are sung correctly the numerals representing those pitches are displayed: "1" for tonic, "3" for mediant, "5" for dominant. Thus the user knows immediately whether the correct pitches are being sung. If he feels the need to start the triad again he may press the space bar. If he needs to have the triad played for him he may press the "P" key. The program will play the tonic triad when so requested only three times. After the third request the program continues as if the user had correctly sung the triad. The "ESC" key may be used as in the key query above to display the menu for aborting the melody as shown in Fig. 4.

Fig. 5. Instructions for singing the tonic triad

Sing the tonic triad (1 3 5 3 1).
I will display the number of each pitch when you sing it correctly.

'P' - Play the triad
ESC - Display option menu
SPACE - Start triad over again

When the tonic triad has been sung correctly or has been played three times the instructions of Fig. 6 are displayed beneath the melody. Pressing the "T" key causes the tonic pitch to be played and it may be played as many times as desired. The "ESC" key is used for the same melody aborting function as before. Pressing any other key is interpreted to mean that the user is prepared to sing the melody.

As the user sings the pitches of the melody correctly the interior of the notes representing those pitches is colored indicating to the user to continue with the next pitch. The user should continue
singing the melody in this manner until the end of the melody is reached or a problem is encountered.

Fig. 6. Instructions prior to singing a melody

Look over the tune for a minute and press any key when you are ready to sing it.

'T' - Play tonic note
ESC - Display option menu

Below the melody a blank staff is displayed with the same clef and key signature. As the user sings, the pitches he sings are drawn on this staff. When a pitch is sung correctly it is displayed on the lower staff directly below the corresponding pitch of the upper staff. If the pitch is sung incorrectly the pitch in error is left displayed on the lower staff even after the pitch is corrected. This procedure allows the user to finish the entire melody and then look back to see which pitches were missed. More importantly he can compare the pitch sung with the intended pitch and analyze his errors. Over a period of time the user may then be able to identify certain tendencies and trends in his errors.

At any point during his performance of the melody the user can press any key on the keyboard to interrupt the process and display the menu shown in Fig. 7. This menu is identical in function to the menu of Fig. 4 except for the addition of one other option: "Play the tune." This option, invoked with the "P" key, causes the melody to be displayed on the screen again and then played. When the melody is finished playing the user is given the option of trying to sing it again.
Fig. 7. Menu at interruption of melody performance

Would you like to:

'S' - Skip this tune
'R' - Return to where you were
'P' - Play the tune
'M' - Display main menu

Select the letter of the option you would like:

When the melody is sung to the end a message is displayed near the bottom of the screen showing the number of notes in the melody and the number of those that were missed. A score expressed in terms of the percentage correct is also displayed. Pressing any key thereafter will cause the menu of Fig. 8 to be displayed.

If option "A" is selected from this menu the user is given another opportunity to sing the same melody. If option "C" is chosen a new melody is selected from the melody library and the entire process is repeated with the new melody. Otherwise the user may have the melody played or he may return to the main menu.

Fig. 8. Menu at conclusion of melody performance

Would you like to:

'C' - Continue with another tune
'A' - Try this tune again
'P' - Play the tune
'M' - Display main menu

Select the letter of the option you would like:

When every melody of the library has been sung and the program is unable to select another the menu of Fig. 9 is displayed. Option "A" causes all of the melodies in the library to be reset as if the library had just been loaded. Therefore all of the melodies will
become available for presentation again. The "L" option allows the user to load another melody library from cassette tape.

Fig. 9. Menu at end of melody library

You have sung every tune that I have in this tune group.

Would you like to:

'A' - Do the same tunes again
'L' - Load a new tune group
'M' - Display main menu

Select the letter of the option you would like:

The "Free Mode" Option

Free mode may be selected only from the main menu by responding with "F." In order to determine the manner in which to display the staff two questions are asked of the user. The first question asks for the user's choice of clef. The only valid responses are "T" for treble clef and "B" for bass clef. The second question asks for the user's choice of key signature. The key should be specified in the same manner and syntax as before.

Using this information a blank staff is displayed with the clef and key signature selected. Any series of pitches desired may then be sung and each pitch is displayed as soon as it is detected. When the end of the staff is reached a second staff is automatically displayed. When the second staff is filled the entire screen is cleared and a new staff is given. This procedure will continue until the user presses any key on the keyboard returning him to the main menu. All pitches are displayed without any transposition and are unaffected by the user's vocal range.
CHAPTER IV

THE EXPERIMENT

The Design of the Experiment

To test the general value of the system described in the preceding chapters an experiment was performed with the students enrolled in the beginning music theory course at Louisiana State University in the Fall semester of 1983. The experiment began one week after midterm examinations and continued for a period of six weeks to the end of the semester.

The entire population of the freshman music theory course was randomly divided into two groups: the experimental group and the control group. The music theory course in which the subjects of the experiment were enrolled was taught in four separate classes each with a different instructor. To minimize the effect of any one instructor each class was separately divided into two equal groups, one for the control group and one for the experimental group. The experimental group of each class was then combined into one as was the control group of each class. Since each class was approximately the same size the effects of each instructor were also approximately equalized.

While the subjects of the experimental group were selected randomly their participation was entirely voluntary. A demonstration of the computer sightsinging system was given prior to seeking the agreement of the selected subjects. Those subjects who declined to participate were assigned to the control group by default. An attempt
was then made to select another participant at random to fill the newly vacated position in the experimental group.

At the time the experiment began there were seventy-eight students enrolled in the music theory course and this group made up the entire population of experimental subjects. While it was desirable to divide the subjects into two equal groups the experimental group began with thirty-three members (six members short of the intended goal) due to the limited availability of the computer system. There was only one computer sightsinging system for the use of all participants and its availability was limited to the normal working hours of the ear training laboratory in which it was installed. Since each member of the experimental group was required to use the sightsinging system for a minimum of one hour per week the number of participants was limited to the number of hours which the laboratory was open. The scheduled operating hours of the laboratory fluctuated during the semester but averaged seven hours per day, five days per week, for a total of thirty-five hours per week. Thus the maximum number of active members of the experimental group was limited to thirty-five.

By the end of the semester six students had resigned from the theory course thereby reducing the entire population to seventy-two. Of these six students four were members of the experimental group. The experiment ended, therefore, with forty-three members in the control group and twenty-nine in the experimental group.

The participation of the subjects of the control group was limited only to the testing phase of the experiment. The grades on the regular mid-term and final sightsinging examinations were recorded as a measurement of the sightsinging ability and progress of each
student. These examinations were the same for each class and were administered by each student's classroom theory instructor. Since no other demands beyond the normal requirements of the course were imposed on the members of the control group they were generally unaware of their participation in the experiment.

The members of the experimental group were given access to the computer system for sightsinging practice in the ear training laboratory during the six-week period of the experiment. Each subject was required to practice a minimum of thirty minutes twice per week. Additional practice was encouraged although the availability of the computer was limited to the operating hours of the laboratory. These practice sessions marked the only difference between the experimental and control groups.

It was hoped that the relative value of the sightsinging system could be determined from the difference, if any, between the rates of improvement of the two groups over the period of the experiment as determined by the scores on the mid-term and final sightsinging examinations. Unfortunately there were several problems with the implementation of the experiment which prevented any valid conclusions from being drawn from the collected data.

The regular operating hours of the ear training laboratory in which the computer system was installed were limited to the daytime hours on weekdays. However, because of personnel problems the laboratory failed to open during many of these regular hours. In addition the operating schedule was changed several times leaving many students unable to predict when the laboratory would be available.
Regardless of these scheduling problems it is unlikely that full cooperation could be expected from all of the participants in the experiment. Indeed, of the original thirty-three members of the experimental group only eighteen actually used the computer system. Of these eighteen only three used the system for a length of time that approached the required minimum of one hour per week. Table 2 shows the number of minutes spent per week on the system by each of the participating members of the experimental group. The students are listed in decreasing order of total time spent. From the data in Table 2 it becomes apparent that no valid conclusions may be drawn.

**TABLE 2**

**TIME UTILIZATION OF THE COMPUTER SIGHTSINGING SYSTEM BY THE PARTICIPATING MEMBERS OF THE EXPERIMENTAL GROUP**

(Time Expressed in Minutes)

<table>
<thead>
<tr>
<th>Student</th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
<th>Week 5</th>
<th>Week 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65</td>
<td>75</td>
<td>50</td>
<td>60</td>
<td>35</td>
<td>45</td>
<td>330</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>97</td>
<td>85</td>
<td>15</td>
<td>25</td>
<td>262</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>68</td>
<td>55</td>
<td>85</td>
<td>30</td>
<td>238</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>45</td>
<td>10</td>
<td></td>
<td>155</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>35</td>
<td></td>
<td>50</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td></td>
<td>35</td>
<td></td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td></td>
<td>65</td>
<td></td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td></td>
<td></td>
<td>50</td>
<td>90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>30</td>
<td>60</td>
<td></td>
<td>90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>35</td>
<td></td>
<td></td>
<td>85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>85</td>
<td></td>
<td></td>
<td></td>
<td>85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>30</td>
<td>45</td>
<td></td>
<td></td>
<td>75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>45</td>
<td></td>
<td>30</td>
<td>75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>30</td>
<td>30</td>
<td></td>
<td></td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
concerning the pedagogical value of the system since the number of students who used the system for a significant amount of time is too small to be statistically valid.

While one of the reasons for this lack of participation may be the scheduling problems mentioned above it is also possible that many of the students did not feel comfortable with the system and thus avoided it. This possibility gains validity from the responses of the participants to the questionnaire that was presented to them at the end of the experiment. Approximately half of the participants said that they felt intimidated or frustrated with the system some of the time.

The lack of participation of the remaining members of the experimental group may be due to scheduling problems or a tendency to feel uneasy with computers. It should be noted, however, that while the members of the experimental group were selected randomly each student was told that his participation was entirely voluntary. In addition the demonstration of the system to the entire population was met with a large amount of enthusiasm. Many of the students who were not selected to participate in the experiment expressed disappointment and a request for inclusion. Thus the lack of cooperation was probably due more to personal scheduling problems than to a fear of the computer.

Analysis of the Experiment Data

Tables 3 and 4 show the sightsinging grades of the members of the experimental and control groups, respectively. With an informal examination of these grades it becomes apparent why no statistically valid conclusions may be drawn.
TABLE 3

SIGHTSINGING EXAMINATION GRADES OF EXPERIMENTAL GROUP
BEFORE AND AFTER THE EXPERIMENT

<table>
<thead>
<tr>
<th></th>
<th>Mid-term Exam</th>
<th>Final Exam</th>
<th>Time Spent</th>
<th>Mid-term Exam</th>
<th>Final Exam</th>
<th>Time Spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td></td>
<td></td>
<td></td>
<td>Class B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>80</td>
<td>262</td>
<td>75</td>
<td>80</td>
<td>330</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>155</td>
<td>63</td>
<td>80</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>82</td>
<td>95</td>
<td>85</td>
<td>88</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>83</td>
<td>90</td>
<td>91</td>
<td>89</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>77</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>82</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>63</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class C

<table>
<thead>
<tr>
<th></th>
<th>Mid-term Exam</th>
<th>Final Exam</th>
<th>Time Spent</th>
<th>Mid-term Exam</th>
<th>Final Exam</th>
<th>Time Spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>88</td>
<td>238</td>
<td>96</td>
<td>90</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>60</td>
<td>95</td>
<td>100</td>
<td>85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>65</td>
<td>80</td>
<td>-</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80</td>
<td>90</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

It was expected that the members of the experimental group would show a higher gain from mid-term grade to final exam grade as compared to the members of the control group. It was also expected that those members of the experimental group who used the system the most would show the most improvement. The student who used the system the most did show an improvement of five points, but the student who ranked second in time spent on the system showed a loss of ten points from the mid-term grade to the final grade. Continuing this line of investigation the student who ranked third in terms of time spent on the system showed essentially no change (-2 points) while the fourth
ranking student showed the largest gain of the entire experimental group: 25 points. Further destroying any chance for finding a pattern in these data, the next ranking student showed a large loss of 28 points.

**TABLE 4**

SIGHTSINGING EXAMINATION GRADES OF CONTROL GROUP
BEFORE AND AFTER THE EXPERIMENT

<table>
<thead>
<tr>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
<th>Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid-term</td>
<td>Final</td>
<td>Mid-term</td>
<td>Final</td>
</tr>
<tr>
<td>85</td>
<td>63</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>100</td>
<td>97</td>
<td>98</td>
<td>90</td>
</tr>
<tr>
<td>100</td>
<td>93</td>
<td>89</td>
<td>94</td>
</tr>
<tr>
<td>55</td>
<td>73</td>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>74</td>
<td>77</td>
</tr>
<tr>
<td>95</td>
<td>77</td>
<td>55</td>
<td>85</td>
</tr>
<tr>
<td>85</td>
<td>97</td>
<td>78</td>
<td>67</td>
</tr>
<tr>
<td>65</td>
<td>73</td>
<td>88</td>
<td>86</td>
</tr>
<tr>
<td>83</td>
<td>83</td>
<td>68</td>
<td>76</td>
</tr>
<tr>
<td>95</td>
<td>83</td>
<td>99</td>
<td>95</td>
</tr>
<tr>
<td>87</td>
<td>90</td>
<td>100</td>
<td>96</td>
</tr>
<tr>
<td>87</td>
<td>90</td>
<td>100</td>
<td>98</td>
</tr>
<tr>
<td>85</td>
<td>62</td>
<td>60</td>
<td>75</td>
</tr>
<tr>
<td>95</td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Clearly, given this limited amount of data, there is no correlation that can be found between the amount of time spent on the system and the degree of improvement in grades. Showing a similar lack of correlation are the grades of the members of the experimental group both before and after the experiment as illustrated in the graph of
Fig. 10. Each asterisk, representing a student of the experimental group, is placed on the chart at the point where his grade before the experiment intersects his grade after the experiment. For instance, the asterisk at the bottom of the graph represents the student who scored 88 on the mid-term exam before the experiment and 60 on the final exam after the experiment.

With larger experimental groups and more consistency within the group one normally expects to find a graph in which the asterisks have been concentrated in a smaller area that stretches at an angle across the graph. In the graph of Fig. 10, however, there are too few asterisks for a pattern to be detected and there is too much variance in their placement. Informally one can see from this graph that it is difficult to predict with a high degree of certainty the final grade of a student from his grade prior to participating in the experiment process.

A more formal statistical analysis of the experiment data draws the same conclusions. A regression analysis was calculated on the experiment data using the Statistical Analysis System (SAS) computer software package on an IBM 3031 computer. The regression coefficient, or R-square factor, was calculated for the experimental and control groups. This coefficient expresses the percentage of the experiment data which can be assimilated in a predictable and coherent system and may range in value from zero to one. A value of zero means that there is no recognizable order to the data while a value of one indicates that all of the data contribute to a logical and predictable system.

The R-square calculated for the control group was 0.467 while the same factor calculated for the experimental group was 0.032. For the
Fig. 10. Graph of the grades of the experimental group.
latter group this factor indicates that, given the grades of a group of students prior to participating in the experiment, the final grades of only three percent of those students can be accurately predicted. With such a low value for the regression coefficient one is unable to prove the hypothesis that the members of the experimental group will show more improvement in sightsinging ability than the members of the control group.

It should be noted, however, that it has not been proven that the system developed in this study has no effect on a student's sightsinging ability. Rather, the results of this experiment are inconclusive since the sizes of the groups were too small to be statistically significant and none of the members of the experimental group actually used the system for the required minimum amount of time. Before the efficacy of the sightsinging system can be determined it will be necessary to conduct another experiment with a larger and more dependable population of participants.

**The Questionaire**

After the experiment had been completed a questionnaire (Fig. 11) was given to each of the eighteen participating members of the experimental group. Thirteen of these questionnaires were returned and attempts to obtain the remaining five were unsuccessful. This questionnaire consists of eight questions, each of which will now be discussed separately.

1. **Do you feel that this system detected your pitches accurately?** There were five possible responses to this question: (1) all of the time, (2) most of the time, (3) some of the time, (4) seldom,
Fig. 11. Questionaire given to the participating members of the experimental group

QUESTIONARE FOR PARTICIPANTS IN THE
COMPUTER-ASSISTED SIGHTSINGING EXPERIMENT

1. Do you feel that this system detected your pitches accurately?
   - All of the time
   - Most of the time
   - Some of the time
   - Seldom
   - Never

2. Did you feel frustrated or intimidated at times while using the system?
   - All of the time
   - Most of the time
   - Some of the time
   - Seldom
   - Never

3. Due to your use of this system, do you feel that your sightsinging ability actually improved?
   - Very much
   - A little
   - No improvement
   - Harmed my skills

4. If this system continues to be available for your use, will you very likely use it?
   - Definitely
   - Probably
   - Probably not
   - Definitely not

5. What did you like the most about this system, or what feature did you find to be the most valuable?

6. What did you dislike the most about this system?

7. Describe any problems that you had.

8. Other comments.
and (5) never. Ten of the thirteen students (77%) responded with "most of the time." One student responded with "all of the time" while the remaining two said "some of the time."

The accuracy of the pitch detection scheme depends on many factors. The system performs best when the student's mouth is as close to the microphone as possible without touching it and there is no background noise. These conditions cannot be expected to be met in all cases. Furthermore the perceived accuracy of the system is subject to the pitch perception ability of the student users. It is entirely possible that a student may not be singing the pitch that he firmly believes he is singing.

2. Did you feel frustrated or intimidated at times while using the system? The available responses to this question are the same as the previous question. The responses to this question were almost evenly divided between "some of the time" and "seldom." Seven responded with the former, six with the latter. Perhaps some of the reasons for this frustration can be found in the responses to questions six and seven of the questionnaire discussed below.

3. Due to your use of this system, do you feel that your sight-singing ability actually improved? The four responses provided to this questions were: (1) very much, (2) a little, (3) no improvement, and (4) harmed my skills. Five of the respondents (38%) checked "very much" while seven (54%) checked "a little." The only remaining response was "no improvement."

The most interesting finding of these responses is that the three students who used the system the most all responded that their sight-singing ability improved very much. The less that the system was used
the more likely that the student's response was "little" or "no" improvement. One strange deviation from this trend appeared in the response from the user who used the system the least. This student, who used the system for only ten minutes during the entire experiment period, said that her sightsinging ability improved very much as a result of using the system.

4. If this system continues to be available for your use, will you very likely use it? The four responses to this question that are available are: (1) definitely, (2) probably, (3) probably not, and (4) definitely not. Four responded with "definitely," six with "probably," and three with "probably not" yielding 31%, 46%, and 23%, respectively. One respondent who checked "probably not" added the comment: "because of time schedule."

Generally the responses to this question correlated with the preceding one. A student who felt that his sightsinging ability improved very much was definitely likely to use the system again. A student who felt that his sightsinging ability improved a little was probably likely to use the system again. The one student who said that his sightsinging ability saw no improvement also said that he probably would not continue to use the system.

5. What did you like the most about this system, or what feature did you find to be the most valuable? Almost all of the responses to this question pointed to the advantages gained from seeing the pitches displayed and bridging the visual and aural senses. Some of the responses are quoted below in their entirety.

"Its ability to visualize the pitch I was singing."

"I liked knowing when I sang the right pitch. It helped my confidence."
"It was easy to correct mistakes when they were recorded and shown."

"System helped hearing of melodies in head before actually singing them. Develops inner ear."

"It would tell you whether your voice was sharp, flat, or correct."

"It made me more aware of what I was doing wrong."

"Being able to see what I was singing immediately notated musically helped me connect my singing to my reading."

"Singing free style and compare with tunes in sightsinging book."

6. What did you dislike the most about this system? The most serious of the responses to this question concern the speed and accuracy of the pitch detection scheme and the inability of the system to recognize and ignore "accidental" pitches. These responses point out the need for an improvement in the means of correcting pitches that have been sung without having to start over again. The more valuable responses to the question are given below.

"Sometimes I felt the pitches I sang did not correspond to the pitches it pictured."

"I could not adjust my pitch when I would be a half tone off. This was frustrating. I wish I could have heard the pitch I was having trouble adjusting."

"... no range transpositions to different keys should be used. Bad for sense of constant pitch."

"It was too touchy. It was hard to correct accidental mistakes."

"Wish that sightsinging tunes from book could be programmed for use."

7. Describe any problems that you had. Most of the respondents mentioned problems in gaining access to the system either because the laboratory was not open or because someone else was using the computer. Others simply reiterated the difficulties explained in the previous question. Some of the responses follow.
"Problems with it hearing my pitch--it sometimes delayed."

"The computer room was never open!!"

"Gaining entrance into computer sometimes difficult."

"Not enough lab time for everyone to use it."

"The instructions could have been a little clearer on how to operate the computer itself."

"Time. Lab not open when I could come."

"... the person who was supposed to be working was not there so I was not able to use it as much as I wanted."

8. Other comments. This last request yielded no useful information. Few of the respondents had any comments to add and those who did respond said nothing of interest for the evaluation of the computer system. Some of the responses are given below.

"Thanks for letting me use the computer. I hope you can continue to improve it."

"The students should be instructed on how to load the computer themselves so they don't have to depend on the instructor to load it."

"Very good idea; needs some of the bugs taken out, but it has great capabilities as a teaching aid. Good work!!"

Conclusions

At the outset it was recognized that the pedagogical value of the sightsinging system developed in this study could not be easily proven and measured with an experiment in which the results are affected by many uncontrollable and often unidentified variables associated with human subjects. Indeed, no formal proof or measurement of value was found with any statistically valid significance. However, these results can be interpreted only in association with the design and execution of one experiment and it would be unwise to assume that the system's value would be the same under all other conditions.
The experiment conducted contained unavoidable flaws in both the design and execution. Perhaps the most serious flaw in the design of the experiment is that the mid-term and final sightsinging grades which were used to measure the sightsinging ability of the students are affected by more than the skills that are developed in the computer system. The computer system was designed to help a sightsinging student improve his reading of pitches while the examination grades are affected by the student's ability to read both pitches and rhythm. Furthermore, the examinations measured two other tasks not directly related to the sightreading of pitches: singing scales and singing a prepared piece.

Additionally, the experimental group was especially inadequately represented for statistical validity and the level of cooperation from those members was much less than had been hoped. Since the number of subjects was limited to the number of students enrolled in the beginning music theory course this shortcoming was unavoidable.

Even if the experiment groups had been very large and the grades of each student accurately represented the same skills developed on the computer system and those grades showed a significant improvement in the skills of the experimental group over the control group it still would not be clear which facet of the computer system was responsible for any measurable improvement. It is possible that such improvement may be due to the greater amount of sightsinging practice that the members of the experimental group had as a result of the availability of the computer system. It may be that the ability of some students improved simply because they were forced to identify the key of the melody before singing it. The same may be said of the
possibility that the student may not have formed the habit of singing the tonic triad prior to singing the melody as he is required to do in the implementation of the current system. It also would not be clear to what extent the principle of immediate visual feedback of pitch contributed to the improvement of the student's sightsinging ability. Rather, the only conclusions that could be drawn with certainty are those that relate to the specific implementation of the system as a whole.

The goals of the present study include the search for answers to questions less ambitious than these. Can the principle of immediate visual feedback of pitch be demonstrated? Can this principle be demonstrated with a high degree of accuracy? Can this principle be utilized in a larger system designed for sightsinging practice? Does the student find the system pleasant enough to use that its value, if any, can be determined by further experimentation? Does the student user feel that the system is capable of helping him improve his sightsinging ability? As the present study has shown, the answer to each of these questions is a definite "yes."

The principle of immediate visual feedback of pitch has been demonstrated in the present study and it has been done with a degree of accuracy that has satisfied the student users of the experiment. The evidence for this assertion can be found in the responses of the experiment participants to the questionnaire. This pitch detection ability has been employed in a large program that is designed to give the student a tool with which he may practice his sightsinging skills and it does so without being difficult to use. Furthermore, the experiment participants feel that this system can help them to improve
their sightsinging ability. The experiment has provided evidence to support these claims and it is in this sense that the experiment, as well as the computer-assisted sightsinging system, may be judged to be a success.
CHAPTER V

CONCLUSION

Pedagogical Benefits

The most important and unique feature of the system developed in the present study is the ability to determine the pitch of aural tones and to display those pitches in music notation in real time. It is on this feature that the value of the system rests. Because of this ability the sightsinging student can get immediate visual feedback on his performance and enjoy the advantages that such feedback provides.

One of the advantages of this immediate visual feedback is that the student is forced to become more actively conscious of the pitches he is singing. When sightsinging the student must originate the sound of a pitch in his mind and associate that sound with the pitch as he expects it to be displayed on the staff. He is thus forced to conceive of the visual and aural forms of the sound simultaneously. With immediate visual feedback he very quickly learns whether he is correct or not and the association of the sound with the symbol can be corrected before the sound-symbol association is strengthened.

Once an error is indicated another advantage is gained from the analysis of the error. Often it is not enough to know when an error is made; it is also necessary to know the exact pitch of the note in error, whether it is higher or lower than the target pitch and by how much. Consistently recurring errors in singing a specific interval,
for example, can be more easily identified and, consequently, given special attention.

Another advantage gained from the system's ability to provide immediate visual feedback is the role the system can play as a confidence builder. Without feedback a student who is unsure of the pitch he is singing remains unsure and finds it difficult to continue. With feedback he gains confidence upon seeing that he is correct and he is then at ease to continue without fear of accumulating more errors. Many of the comments on the questionnaire discussed in the previous chapter appear to confirm this notion.

To summarize these advantages the immediate visual feedback allows for the development of the relationship between the visual and aural senses of music. Since sightsinging involves the translation of a visual form into an aural one it is likely that this feedback would be beneficial to the sightsinging student in the development of his skills.

While it is true that many students are not comfortable with the computer as a learning tool there are also many students who enjoy it and even prefer it over conventional learning tools. Many students of sightsinging, particularly those without vocal training, express timidity and embarrassment at the prospect of singing for other people. Those same students, however, are likely to find it less intimidating to sing for a computer which has no means of judging their vocal talent. It is possible, therefore, that such students will devote more time to sightsinging practice on the computer than they would if it were not available. Indeed, for any student the
computer may be the only practical learning tool available for developing sightsinging skills.

One other advantage to the system developed in this study deserves mention. Since the system is inexpensive it is likely to be affordable by those university music schools which desire to provide such a tool for the laboratory use of the sightsinging students. Indeed, many individual personal computer owners might find it within their financial grasp.

It is difficult to judge the monetary cost of the system since much of it was developed by this writer and is not being marketed. Furthermore, the computer on which this project was implemented is no longer available. However, the cost of the system can be approximated by adding the original cost of the Sorcerer computer in 1978 ($1200) and the cost of the Gentle Electric Pitch Follower ($300) to the estimated cost of the software and hardware developed by this writer ($300). The total cost of $1800 is below the $2000 financial goal of the project. For those music school laboratories which already own a suitable computer, then, assuming the software could be adapted for that computer, the cost would be considerably lower.

As far as can be presently determined the computer-assisted sightsinging system developed in the present study is the first inexpensive system capable of the real-time translation of aural pitches to music notation that has been made available to sightsinging students. Not only has the system been developed, but it also performs well enough to satisfy the many of the needs of the sightsinging student. It is within these characteristics of the system that the significance of the present study lies.
Recommendations

There are several areas of future study from which the current implementation of the system could benefit. The development of a real-time pitch detection algorithm that improves upon the detection speed of the current implementation would be of value. While the pitch detection speed of the current project is adequate many students find that they have to consciously restrain themselves from singing faster than the system can follow. Such restrictions should be avoided when possible so that the student may be as comfortable as possible.

An improvement in the pitch detection speed of the system would also facilitate the implementation of rhythm. In the current project rhythmic interpretation was not implemented since it is dependent upon the accurate detection of pitches. A successful demonstration of a fast and accurate pitch detection scheme could then lead to the interpretation of rhythmic data.

Rhythmic interpretation is a topic that deserves isolated study in itself. There are many problems, some of which have already been described, to be solved when deciding upon the rhythmic value that is to be assigned to a given sound. Solving these problems prior to confusing them further with the problems of pitch detection would be beneficial. Furthermore, the sightsinging student could benefit from a system that allows him to practice rhythmic reading alone.

One of the goals of the present study is to demonstrate the principle of immediate visual feedback of pitch in sightsinging. While an attempt has been made in the present study to find an indi-
cation of the value of this principle it is felt that further study is needed before the merit of the principle can be known with certainty.

The experiment conducted in the present study tested only one specific implementation of the system. In any experiment one must be careful to attribute its success or failure to only those factors which contributed to that result. It is entirely possible, for example, that the only reason that the ability of some sightsinging students improved during the experiment was that the system forced them to correctly identify and aurally establish the key before they were allowed to sing. Their improvement may have had little or nothing to do with the principle of immediate visual feedback of pitch.

More study is needed before conclusions may be drawn concerning the value of immediate visual feedback of pitch. Studies are needed which isolate the value of various protocols associated with the task of sightsinging. For example, how important is it that the student identify the key or sing the tonic triad before singing a melody? What information should be presented to the student and what is the best way to do so? Once questions such as these are answered subsequent experiments may be designed which test precisely those facets which are desired. If they are to be meaningful, however, those experiments must employ a much larger number of participants than in the present study.

One final recommendation concerns the implementation of the current project on other more popular computers. If other studies are to be conducted the software will have to be adapted for use on the computers most likely to be encountered in music school laboratories.
Transferring the software to these computers can prevent the lengthy process of duplicating the work that has already been accomplished in the present study.

From these recommendations it becomes apparent that there is much work that remains to be done before all of the questions raised in the present study can be answered. It is hoped that these questions will stimulate further studies which will eventually be of benefit to the sightsinging student. If any of the information provided in this study proves to be useful in future research then the present study may be considered to be successful.
SELECTED BIBLIOGRAPHY

Books


Periodicals


Edwards, Judith; Norton, Shirley; Taylor, Sandra; Van Dusseldorp, Ralph; and Weiss, Martha. "Is Computer-Assisted Instruction Effective?" Association for Educational Data Systems 7 (Summer 1974):122-126.


Sondhi, M. M. "New Methods of Pitch Extraction." Institute of Electrical and Electronics Engineers Transactions on Audio and Electroacoustics 16 (June 1968):262.


Theses and Dissertations


Interviews


Prevel, Martin. Laval University, Quebec, Quebec, Canada. Interview, 21 January 1984.

APPENDIX A

LISTING OF MUSIC CONTROL PROGRAM
TITLE MUSIC CONTROL PROGRAM
SUBTTL RAM DATA SECTION

; MCP RAM DATA STORAGE AREA

; ASSEMBLED SEPARATELY FROM THE REMAINDER OF MCP TO KEEP
; LOAD ADDRESS INDEPENDENT OF THE ROM CODE ADDRESS.

.Z

DSEG
ORG 0000H

.PHASE 9788H

GLOBAL SAFBAR
GLOBAL TUNBL
GLOBAL TUNEND
GLOBAL WRKSPC
GLOBAL WRKSPC
GLOBAL PTR1
GLOBAL PTR2
GLOBAL PTR3
GLOBAL PTR4
GLOBAL INC1
GLOBAL INC2
GLOBAL INC3
GLOBAL INC4
GLOBAL IGFSET
GLOBAL TEMPO
GLOBAL SAMP32
GLOBAL SAMPL1
GLOBAL SAMPL2
GLOBAL SAMPL3
GLOBAL SAMPL4
GLOBAL UNATBL
GLOBAL KEYLET
GLOBAL KEYACD
GLOBAL KEYMOD
GLOBAL OLDKYL
GLOBAL OLDKYA
GLOBAL DBLFIL
GLOBAL UNHAC
GLOBAL XPOSX
GLOBAL XPOSX
RAM DATA SECTION

GLOBAL XPOSICO
GLOBAL WRNOUT
GLOBAL INHSTO
GLOBAL SAVEIX
GLOBAL SAVEIY
GLOBAL TEMP01
GLOBAL TEMP02
GLOBAL PARBIT
GLOBAL PARACT
GLOBAL PART01
GLOBAL PART02
GLOBAL PART03
GLOBAL PART04
GLOBAL PART05
GLOBAL PART06
GLOBAL PART07
GLOBAL PART08
GLOBAL PART09
GLOBAL PART10
GLOBAL PART11
GLOBAL PART12
GLOBAL PART13
GLOBAL PART14
GLOBAL PART15
GLOBAL PART16

EXTERNAL TTSBAR

TUNE TABLE

A record of the entire tune is stored here. There is an entry in the table for each pitch. Each entry consists of four bytes in the format: (1) unadjusted pitch, (2) octave, (3) rhythm and (4) current position increment. Bar lines are represented by the value 0FEH in each of these four bytes; double bar by 0FFH.

The unadjusted pitch is in the following format:

<table>
<thead>
<tr>
<th>BIT</th>
<th>VALUE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-5</td>
<td>000</td>
<td>natural in key signature</td>
</tr>
<tr>
<td></td>
<td>001</td>
<td>flat in key signature</td>
</tr>
<tr>
<td></td>
<td>010</td>
<td>sharp in key signature</td>
</tr>
<tr>
<td></td>
<td>011</td>
<td>natural not in key signature</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>flat not in key signature</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>sharp not in key signature</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>double flat</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>double sharp</td>
</tr>
</tbody>
</table>
Note on line

Note on space

Unused

Displacement from top of partition

At a staff orientation of 0 and unadjusted for octave. A0 = 6, B0 = 5, C0 = 3, etc.

The octave byte is a plus-or-minus binary value of the octave in the range -8 to +7. The zero octave extends from the 'A' on the top line of the bass clef to the 'G' above middle 'C'. The octave above this one is 1, below it is -1, etc.

The rhythm byte is in the following format:

<table>
<thead>
<tr>
<th>BIT</th>
<th>VALUE</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>FERMATA in effect</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>NO FERMATA</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>NOTE IS TIED TO NEXT NOTE; NEXT NOTE IS NOT NECESSARILY TIED</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>NO TIE</td>
</tr>
<tr>
<td>4,3</td>
<td>0</td>
<td>NO DOT</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SINGLE DOT</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>UNUSED</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>DOUBLE DOT</td>
</tr>
<tr>
<td>2,1,0</td>
<td>000</td>
<td>DOUBLE WHOLE</td>
</tr>
<tr>
<td></td>
<td>001</td>
<td>WHOLE</td>
</tr>
<tr>
<td></td>
<td>010</td>
<td>HALF</td>
</tr>
<tr>
<td></td>
<td>011</td>
<td>QUARTER</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>EIGHTH</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>SIXTEENTH</td>
</tr>
<tr>
<td></td>
<td>110</td>
<td>THIRTY-SECOND</td>
</tr>
<tr>
<td></td>
<td>111</td>
<td>BLACK WHOLE</td>
</tr>
</tbody>
</table>

The current position increment is a binary value representing the starting position of the next element relative to the ending position of the previous element. A value of 1 will cause the element to be printed in the first column following the previous element.

SAFBAR: DEFB TTSBAR,TTSBAR,TTSBAR,TTSBAR ; SAFETY BAR LINE

TUWBL: DEFS 03F8H

TUNEND EQU $
RAM DATA SECTION

; WORK SPACE
; THIS SPACE IS USED TO STORE THE PLAYABLE FORM OF TUNES.

9B00  NRKSPC: DEFS 0400H

; WAVEFORM TABLE POINTERS FOR EACH OF THE 4 VOICES

9F00  PTR1: DEFS 2 ;VOICE 1
9F02  PTR2: DEFS 2 ;VOICE 2
9F04  PTR3: DEFS 2 ;VOICE 3
9F06  PTR4: DEFS 2 ;VOICE 4

; INCREMENTS THROUGH WAVEFORM TABLE FOR THE PITCHES OF
; EACH VOICE

9F00  INC1: DEFS 2 ;VOICE 1
9F08  INC2: DEFS 2 ;VOICE 2
9F0C  INC3: DEFS 2 ;VOICE 3
9F0E  INC4: DEFS 2 ;VOICE 4

9F10  00 IOFSET: DEFB 00H ;INTEGER OFFSET VALUE
9F11  TEMPO: DEFS 1 ;TEMPO OF TUNE
9F12  SAMP32: DEFS 2 ;NUMBER OF SAMPLES IN A 32ND NOTE
                  (INITIALIZED BY 'CALC32')

; INPUT SAMPLE STORAGE AREA

9F14  00 SAMPL1: DEFB 00H
9F15  00 SAMPL2: DEFB 00H
9F16  00 SAMPL3: DEFB 00H
9F17  00 SAMPL4: DEFB 00H

; UNADJUSTED PITCH TABLE

; EACH ENTRY IN THIS TABLE CORRELATES WITH A FREQUENCY RANGE
; IN THE FREQUENCY TABLE. THIS TABLE IS USED TO TRANSLATE
; A FREQUENCY INTO ITS UNADJUSTED PITCH. EACH ENTRY IS THE
; UNADJUSTED PITCH FOR THAT ABSOLUTE PITCH. THIS TABLE MUST BE
; RE-INITIALIZED FOR EACH KEY CHANGE, SINCE THE WAY A PITCH IS
; SPELLED IS DEPENDENT UPON ITS KEY. THE FIRST ENTRY IS FOR
; A, THEN B-FLAT, ETC UP TO A-FLAT.

9F18  UNATBL: DEFS 0CH
; KEY SIGNATURE DATA
; THE NEXT FIVE FIELDS MUST BE CONTIGUOUS

9F24
KEYLET: DEFS 1,00H ;KEY LETTER NAME
9F25
KEYACD: DEFS 1,00H ;KEY ACCIDENTAL TYPE
9F26
KEYMOD: DEFS 1,00H ;KEY MODE - MAJOR OR MINOR

9F27
OLDKYL: DEFS 1,00H ;ORIGINAL KEY LETTER BEFORE TRANSPOSE
9F28
OLDKYA: DEFS 1,00H ;ORIGINAL KEY ACCIDENTAL

9F29
DBLFLT: DEFS 1 ;DOUBLE FLAT FLAG
9F2A 00
UNACCD: DEFB 00H ;UNCONDITIONAL ACCIDENTAL FLAG. ANY NON-ZERO VALUE CAUSES DISPLAY OF ACCIDENTAL

; THESE THREE FIELDS MUST BE CONTIGUOUS

9F2B 00
XPOSHS: DEFB 00H ;TRANSPOSITION DISPLACEMENT IN HALF STEPS
9F2C 00
XPOSOD: DEFB 00H ;TRANSPOSITION DISPLACEMENT IN OCTAVES; SIGN INDICATES DIRECTION (NEGATIVE IS DOWN, POSITIVE IS UP)

9F2D 00
XPOSOC: DEFB 00H ;TRANSPOSITION CUT OFF LETTER; DEPENDING ON THE DIRECTION OF THE TRANSPOSITION, EVERY LETTER EITHER ABOVE OR BELOW THIS LETTER WILL HAVE TO HAVE ITS OCTAVE ADJUSTED

9F2E
WRKNOT: DEFS 5 ;TEMPORARY WORK AREA FOR NOTES TO BE DISPLAYED

9F33 00
INHSTO: DEFB 00H ;INHIBIT STORE IN TUNE TABLE FLAG

9F34
SAVEIX: DEFS 2,0 ;TEMPORARY STORAGE FOR REGISTER IX
9F36
SAVEIY: DEFS 2,0 ;TEMPORARY STORAGE FOR REGISTER IY

9F38
TEMPO1: DEFS 2 ;GENERAL TEMPORARY STORAGE
9F3A
TEMPO2: DEFS 2

; PARTITION CONTROL BLOCK

9F3C
PARBIT: DEFS 2,0 ;DEFINED PARTITIONS BIT MAP
9F3E
PARACT: DEFS 1,0 ;ACTIVE PARTITION NUMBER (1 - MAXPAR)

9F3F
PART01: DEFS 2,00H ;BEGINNING ADDRESS OF PARTITION
9F41
DEFS 1,00H ;LENGTH OF LINE IN PARTITION
RAM DATA SECTION

9F42 DEF 1,00H ; NUMBER OF LINES IN PARTITION
9F43 DEF 1,00H ; LINE NUMBER FOR BEGINNING OF STAFF.
9F44 DEF 1,00H ; (X COORDINATE FOR TEXT DISPLAY)
9F45 DEF 1,00H ; CURRENT HORIZONTAL POSITION.
9F46 DEF 1,00H ; (Y COORDINATE FOR TEXT DISPLAY)
9F47 DEF 1,00H ; RESERVES
9F47 PART02: DEF 8H,00H ; SAME FORMAT AS FOR PART01
9F4F PART03: DEF 8H,00H
9F57 PART04: DEF 8H,00H
9F5F PART05: DEF 8H,00H
9F67 PART06: DEF 8H,00H
9F6F PART07: DEF 8H,00H
9F77 PART08: DEF 8H,00H
9F7F PART09: DEF 8H,00H
9F87 PART10: DEF 8H,00H
9F8F PART11: DEF 8H,00H
9F97 PART12: DEF 8H,00H
9F9F PART13: DEF 8H,00H
9FA7 PART14: DEF 8H,00H
9FAF PART15: DEF 8H,00H
9FB7 PART16: DEF 8H,00H

; .DEPHASE
;
END
Macros:

Symbols:

9F29I  DBLFLT
9F0CI  INC3
9F18I  IOFSET
9F26I  KEYSDD
9F3E1  PARACT
9F47I  PART02
9F5F1  PART05
9F77I  PART08
9F8F1  PART11
9FA7I  PART14
9F86I  PTR4
9F14I  SAMPL1
9F17I  SAMPL4
9F38I  TEMPO1
8000X  TTSBAR
9F18I  UNATBL
9B88I  WRIKSPC
9F2C1  XPOS00

9F08I  INC1
9F0E1  INC4
9F25I  KEYACD
9F28I  OLDKYA
9F3C1  PARBIT
9F4FI  PART03
9F67I  PART06
9F7FI  PART09
9F97I  PART12
9FAFI  PART15
9F02I  PTR2
9F70I  SAFBAR
9F15I  SAMPL2
9F34I  SAVEIX
9F3AI  TEMPO2
9B00I  TUNEND
9F2AI  UNACCD
9F2DI  XPOS00

9F0A1  INC2
9F33I  INHST0
9F24I  KEYLET
9F27I  OLDKYL
9F3F1  PART01
9F57I  PART04
9F6FI  PART07
9F87I  PART10
9F9FI  PART13
9FB7I  PART16
9F84I  PTR3
9F12I  SMP32
9F16I  SMP33
9F36I  SAVEIY
9F11I  TEMPO
9708I  TUNBL
9F2EI  WRIKNOT
9F2BI  XPOS0S

No Fatal error(s)
TITLE MUSIC CONTROL PROGRAM
SUBTTL EXTERNAL NAMES SECTION

; MCP
; MUSIC CONTROL PROGRAM
; AN OPERATING SYSTEM FOR MUSIC FUNCTIONS.
; VERSION 1.0
; OCTOBER 1983
;
.Z80
;
; DATA SECTION

0000' DSEG ORG 0000H
;
0000' DORG EQU $
C000 BASE EQU 0C000H
;
C800 DBASE EQU BASE
;
MACLIB B:MCPEXT.MAC
;
; EXTERNAL NAMES FOR DATA IN SEPARATE RAM FILE
;
EXTERNAL SAFBAR
EXTERNAL TUNBL
EXTERNAL TUNEND
EXTERNAL MKSPC
EXTERNAL PTR1
EXTERNAL PTR2
EXTERNAL PTR3
EXTERNAL PTR4
EXTERNAL INC1
EXTERNAL INC2
EXTERNAL INC3
EXTERNAL INC4
EXTERNAL IOFSET
EXTERNAL TEMPO
EXTERNAL SAMP32
EXTERNAL SAMPL1
EXTERNAL NAMES SECTION

C  EXTERNAL  SAMPL2
C  EXTERNAL  SAMPL3
C  EXTERNAL  SAMPL4
C  EXTERNAL  UNATBL
C  EXTERNAL  KEYLET
C  EXTERNAL  KEYACD
C  EXTERNAL  KEYMOD
C  EXTERNAL  OLXYL
C  EXTERNAL  OLXYA
C  EXTERNAL  DBLFLT
C  EXTERNAL  UNACD
C  EXTERNAL  XPOS00
C  EXTERNAL  XPOS0D
C  EXTERNAL  XPOS0C
C  EXTERNAL  HRN001
C  EXTERNAL  INHIST0
C  EXTERNAL  SAVEIX
C  EXTERNAL  SAVEIY
C  EXTERNAL  TEMP01
C  EXTERNAL  TEMP02
C  EXTERNAL  PARBIT
C  EXTERNAL  PARACT
C  EXTERNAL  PART01
C  EXTERNAL  PART02
C  EXTERNAL  PART03
C  EXTERNAL  PART04
C  EXTERNAL  PART05
C  EXTERNAL  PART06
C  EXTERNAL  PART07
C  EXTERNAL  PART08
C  EXTERNAL  PART09
C  EXTERNAL  PART10
C  EXTERNAL  PART11
C  EXTERNAL  PART12
C  EXTERNAL  PART13
C  EXTERNAL  PART14
C  EXTERNAL  PART15
C  EXTERNAL  PART16
C  
C  
C  ;
C  
C  ;

SUBTL  EQUATE SECTION

@EJECT
EQUATE SECTION

MACLIB B\ASCII\EXT.MAC

; ASCII EQUATE TABLE

; CONTROL CHARACTERS

00001 C CURLFT EQU 01H ;CURSOR LEFT
0000A C LINLED EQU 0AH ;LINE FEED
00010 C CARRET EQU 00H ;CARRIAGE RETURN
00011 C HOME EQU 11H ;CURSOR HOME
00013 C CURHT EQU 13H ;CURSOR RIGHT
00017 C CURSUP EQU 17H ;CURSOR UP
00089 C TAB EQU 09H ;HORIZONTAL TAB

; PRINTABLE CHARACTERS

00209 C SPACE EQU 20H ;BLANK
0029A C ASTRSK EQU 2AH ;ASTERISK

; CLEFS

00300 C TRCL12 EQU 80H ;TREBLE CLEF CHARACTERS FROM TOP
00301 C TRCL22 EQU 81H ; TO BOTTOM, LEFT TO RIGHT
00302 C TRCL23 EQU 82H ; THE FIRST DIGIT IN THE NAME
00303 C TRCL32 EQU 83H ; REFERS TO THE ROW THE CHARACTER
00304 C TRCL41 EQU 84H ; IS PRINTED ON; THE SECOND DIGIT
00305 C TRCL42 EQU 85H ; REFERS TO THE COLUMN
00306 C TRCL43 EQU 86H
00307 C TRCL51 EQU 87H
00308 C TRCL52 EQU 88H
00309 C TRCL53 EQU 89H
0030A C TRCL61 EQU 8AH
0030B C TRCL62 EQU 8BH
0030C C TRCL63 EQU 8CH
0030D C TRCL72 EQU 8DH

; BASS

0030E C BSCL11 EQU 8EH
0030F C BSCL12 EQU 8FH
00310 C BSCL13 EQU 90H
00311 C BSCL21 EQU 91H
MUSIC CONTROL PROGRAM

EQUATE SECTION

0092  C  BSCL23 EQU 92H
0093  C  BSCL32 EQU 93H
0094  C  BSCL33 EQU 94H
0095  C  BSCL42 EQU 95H
0096  C  BSCL51 EQU 96H
   C  STAFF CHARACTERS
   C  VLINE EQU 897H  ; VERTICAL LINE FOR ENDS OF STAFF
0097  C  HLINE EQU 898H  ; HORIZONTAL LINE FOR LINES OF STAFF
0098  C  BARLN1 EQU 899H  ; TOP BARLINE CHARACTER
0099  C  BARLN2 EQU 89AH  ; MIDDLE BARLINE CHARACTER
009A  C  BARLN3 EQU 89BH  ; BOTTOM BARLINE CHARACTER
009B  C  DBLBAR EQU 89CH  ; DOUBLE BARLINE CHARACTER
009C  C  ;
   C  ACCIDENTALS
   C  NATURAL
   C  ANLTHL EQU 09DH  ; NATURAL ON LINE, TOP, NO LINE
009D  C  ANLTHL EQU 09EH  ; NATURAL ON LINE, TOP, WITH LINE
009E  C  ANLTHL EQU 09FH  ; NATURAL ON LINE, MIDDLE, NO LINE
009F  C  ANLTHL EQU 0A1H  ; NATURAL ON LINE, BOTTOM, NO LINE
00A0  C  ANLTHL EQU 0A2H  ; NATURAL ON LINE, BOTTOM, WITH LINE
00A1  C  ANSTHNL EQU 0A3H  ; NATURAL ON SPACE, TOP, NO LINE
00A2  C  ANSTHNL EQU 0A4H  ; NATURAL ON SPACE, TOP, WITH LINE
00A3  C  ANSTHNL EQU 0A5H  ; NATURAL ON SPACE, MIDDLE, NO LINE
00A4  C  ANSTHNL EQU 0A6H  ; NATURAL ON SPACE, MIDDLE, WITH LINE
00A5  C  ANSTHNL EQU 0A7H  ; NATURAL ON SPACE, BOTTOM, NO LINE
00A6  C  ANSTHNL EQU 0A8H  ; NATURAL ON SPACE, BOTTOM, WITH LINE
00A7  C  ANSTHNL EQU 0A9H  ; NATURAL ON SPACE, BOTTOM, WITH LINE
00A8  C  ANSTHNL EQU 0AAH  ; FLAT ON LINE, TOP, NO LINE
00A9  C  Aflenl EQU 0A9H  ; FLAT ON LINE, TOP, WITH LINE
00AA  C  Aflenl EQU 0ABH  ; FLAT ON LINE, BOTTOM, NO LINE
00AB  C  Aflenl EQU 0ACH  ; FLAT ON SPACE, TOP, NO LINE
00AC  C  Aflenl EQU 0ADH  ; FLAT ON SPACE, BOTTOM, NO LINE
00AD  C  Aflenl EQU 0AEH  ; SHARP ON LINE, TOP, ONLY, NO LINE
00AE  C  Aflenl EQU 0AFH  ; SHARP ON LINE, TOP, ONLY, NO LINE
00AF  C  ASLTON EQU 0AFH  ; SHARP ON LINE, TOP, ONLY, NO LINE
MUSIC CONTROL PROGRAM
EQUATE SECTION

00B0 C ASLTOL EQU 0B0H ; SHARP ON LINE, TOP, ONLY, WITH LINE
00B1 C ASLMLN EQU 0B1H ; SHARP ON LINE, MIDDLE, LEFT, NO LINE
00B2 C ASLMIL EQU 0B2H ; SHARP ON LINE, MIDDLE, LEFT, WITH LINE
00B3 C ASLMLN EQU 0B3H ; SHARP ON LINE, MIDDLE, RIGHT, NO LINE
00B4 C ASLMRL EQU 0B4H ; SHARP ON LINE, MIDDLE, RIGHT, WITH LINE
00B5 C ASLBN EQU 0B5H ; SHARP ON LINE, BOTTOM, ONLY, NO LINE
00B6 C ASLBDL EQU 0B6H ; SHARP ON LINE, BOTTOM, ONLY, WITH LINE
00B7 C ASSTLH EQU 0B7H ; SHARP ON SPACE, TOP, LEFT, NO LINE
00B8 C ASSTLL EQU 0B8H ; SHARP ON SPACE, TOP, LEFT, WITH LINE
00B9 C ASSTRN EQU 0B9H ; SHARP ON SPACE, TOP, RIGHT, NO LINE
00BA C ASSTRL EQU 0BAH ; SHARP ON SPACE, TOP, RIGHT, WITH LINE
00BB C ASSBLN EQU 0BBH ; SHARP ON SPACE, BOTTOM, LEFT, NO LINE
00BC C ASSBLIL EQU 0BCH ; SHARP ON SPACE, BOTTOM, LEFT, WITH LINE
00BD C ASSBN EQU 0BDH ; SHARP ON SPACE, BOTTOM, RIGHT, NO LINE
00BE C ASSBL EQU 0BCH ; SHARP ON SPACE, BOTTOM, RIGHT, WITH LINE
00BF C ASSPC1 EQU 0B8H ; SHARP FOR KEY SIGNATURE, SPECIAL

08C0 C ADLSLN EQU 0C0H ; DOUBLE SHARP ON LINE, ONLY, NO LINE
08C1 C ADLSLL EQU 0C1H ; DOUBLE SHARP ON LINE, ONLY, WITH LINE
08C2 C ADSTN EQU 0C2H ; DOUBLE SHARP ON SPACE, TOP, NO LINE
08C3 C ADSTTL EQU 0C3H ; DOUBLE SHARP ON SPACE, TOP, WITH LINE
08C4 C ADSSBN EQU 0C4H ; DOUBLE SHARP ON SPACE, BOTTOM, NO LINE
08C5 C ADSSBL EQU 0C5H ; DOUBLE SHARP ON SPACE, BOTTOM, WITH LINE

08C6 C NHWL EQU 0C6H ; NOTE HEAD, WHOLE, LINE
08C7 C NHWSLN EQU 0C7H ; NOTE HEAD, WHOLE, SPACE, TOP, NO LINE
08C8 C NHWSLN EQU 0C8H ; NOTE HEAD, WHOLE, SPACE, TOP, WITH LINE
08C9 C NHWSN EQU 0C9H ; NOTE HEAD, WHOLE, SPACE, BOTTOM, NO LINE
08CA C NHWSBL EQU 0CAH ; NOTE HEAD, WHOLE, SPACE, BOTTOM, WITH LINE

08CB C NHBL EQU 0CBH ; NOTE HEAD, WHOLE, LINE
08CC C NHBSTN EQU 0CCH ; NOTE HEAD, WHOLE, SPACE, TOP, NO LINE
08CD C NHBSTL EQU 0CDH ; NOTE HEAD, WHOLE, SPACE, TOP, WITH LINE
08CE C NHBSBN EQU 0CCH ; NOTE HEAD, WHOLE, SPACE, BOTTOM, NO LINE
08CF C NHBSBL EQU 0CFH ; NOTE HEAD, WHOLE, SPACE, BOTTOM, WITH LINE

MACLIB B:MCPEQUAT.MAC

EQUATE VALUES
EQUATE SECTION

; ADDRESSES

E010  KEYBOARD EQU 0E010H ; SORCERER MONITOR ROUTINE WHICH CHECKS
; TO SEE IF A KEYBOARD CHARACTER HAS BEEN PRESSED

; DATA

GLOBAL TTSBAR

0035  MAXFNC  EQU 35H  ; NUMBER OF SYSTEM CALLS
0062  MAXCLF  EQU 02H  ; LARGEST CLEF NUMBER POSSIBLE PLUS 1
0015  CLFLLEN  EQU 15H  ; NUMBER OF CHARACTERS IN CLEF ARRAY
03F8  TUNSLZ  EQU 03F8H ; TUNE TABLE SIZE
FFFFC  TTETLC  EQU 0FFFC  ; MINUS TUNE TABLE ENTRY LENGTH
0004  TTELLEN  EQU 0004H ; TUNE TABLE ENTRY LENGTH
00FF  DPORT  EQU 0FFH  ; DATA INPUT PORT
00FF  STPORT  EQU 0FFH  ; DATA INPUT STATUS PORT
00FF  NOISE  EQU 0FFH  ; VALUE FOR NOISE OR SILENCE
0066  MINSAM  EQU 06H  ; MINIMUM NUMBER OF CONSECUTIVE SAMPLES
; OF SAME PITCH BEFORE PITCH IS VALID
0004  NUMSMP  EQU 04H  ; NUMBER OF SAMPLES TO COLLECT FOR AVERAGE
0010  MINPIT  EQU 10H  ; LOWEST INTEGER PITCH TO ALLOW (C-2)
0040  MAXPIT  EQU 40H  ; HIGHEST INTEGER PITCH TO ALLOW (C2)
0005  MAXDEV  EQU 05H  ; HIGHEST AVERAGE DEVIATION
001A  FTLELEN  EQU 1AH  ; FREQUENCY TABLE LENGTH
F000  SCREEN  EQU 0F000H ; FIRST ADDRESS OF THE SCREEN
077F  SOHLEN  EQU 077FH  ; LENGTH OF ENTIRE SCREEN
001E  NULLIN  EQU 01EH  ; NUMBER OF LINES ON SCREEN
0040  LINLEN  EQU 40H  ; SCREEN LINE LENGTH
0180  LENX4  EQU  LINLENX4  ; LENGTH OF 4 SCREEN LINES
FF8C  NEGLEN  EQU 0FF8CH  ; NEGATIVE LINE LENGTH
0089  MINBAR  EQU 09H  ; MINIMUM MEASURE LENGTH
00FE  TTSBAR  EQU 0FEH  ; SINGLE BAR INDICATOR IN TUNE TABLE
00FF  TTDBAR  EQU 0FFH  ; DOUBLE BAR INDICATOR IN TUNE TABLE
0010  HDRLEN  EQU 10H  ; TUNE HEADER LENGTH
0040  TYPSC  EQU 0A0H  ; TUNE IN SCALE DEGREE/MODIFIER FORMAT
0058  TYPLET  EQU 080H  ; TUNE IN LETTER NAME/ACCIDENTAL FORMAT

; INDEX REGISTER EXTENDED INSTRUCTIONS

0024  INCYH  EQU 24H  ; INCREMENT HIGH BYTE OF IY
0025  DECYH  EQU 25H  ; DECREMENT HIGH BYTE OF IY
002C  INCYL  EQU 2CH  ; INCREMENT LOW BYTE OF IY
002D  DECYL  EQU 2DH  ; DECREMENT LOW BYTE OF IY
EQUATE SECTION

C ; RETURN CODE VALUES
C
C 0016 C OORANG EQU 16H ; OUT OF RANGE
C 0017 C EOQ EQU 17H ; END OF MEASURE
C 0018 C EOS EQU 18H ; END OF STAFF
C 0019 C ELP EQU 19H ; END OF PARTITION NUMBER LIST
C 001A C EOT EQU 1AH ; END OF TUNE
C 0088 C EOSTR G EQU 00H ; END OF TEXT STRING
C 00FD C KBINT R EQU 0FDH ; KEYBOARD INTERRUPT FLAG VALUE
C ;
C ; SINGLE-BIT ASSIGNMENTS
C
C 0086 C DOUBLE EQU 6 ; DOUBLE FLAT/SNAP BIT
C 0085 C FLTP EQU 5 ; FLAT/SNAP BIT
C 0084 C LINS P EQU 4 ; LINE/SPACE BIT
C 0086 C FERM EQU 6 ; FERMATA BIT
C 0085 C NTOED EQU 5 ; NOTE IS TIED TO NEXT NOTE
C 0084 C DBLDOT EQU 4 ; DOUBLE DOT
C 0083 C SNGDOT EQU 3 ; SINGLE DOT
C ;
C ; TUNE TABLE INDEXES
C
C 0080 C UNAPIT EQU 0 ; UNADJUSTED PITCH
C 0081 C OCTAVE EQU 1 ; OCTAVE
C 0082 C RHYBYTE EQU 2 ; RHYTHM BYTE
C 0083 C CPINCR EQU 3 ; CURRENT POSITION INCREMENT
C ;
C ; PARTITION TABLE INDEXES
C
C 0088 C PARADL EQU 0 ; LOW ORDER BYTE OF START ADDRESS
C 0081 C PARADH EQU 1 ; HIGH ORDER BYTE OF START ADDRESS
C 0082 C PSLR EQU 2 ; PARTITION LINE LENGTH
C 0083 C PSLR EQU 3 ; NUMBER OF LINES IN PARTITION
C 0084 C SFTORN EQU 4 ; STAFF ORIENTATION IN PARTITION
C 0085 C CURPOS EQU 5 ; CURRENT HORIZONTAL POSITION IN PARTITION
C 0086 C CLFCOD EQU 6 ; CURRENT CLEF USED IN PARTITION
C 0085 C XCOORD EQU 5 ; X COORDINATE FOR TEXT DISPLAY
C 0084 C YCOORD EQU 4 ; Y COORDINATE FOR TEXT DISPLAY
C ;
C 0088 C PARTEL EQU 8 ; PARTITION TABLE ENTRY LENGTH
C 0010 C MAXPAR EQU 10H ; MAXIMUM NUMBER OF PARTITIONS
C 
C SUBTL RCM DATA SECTION

#EJECT
MCP: JP SYSTEM ;MCP ENTRY POINT

0010 .RADIX 16

SCALES IN 'PLAREL' FORMAT

-major

C803 0F MISCAL: DEFB 0FH ;NUMBER OF NOTES IN SCALE
C804 08 00 08 00 DEFB 08,00,08,00,02,02,02,02,02,04
C808 08 02 02 02
C80C 02 04
C80E 04 04 04 04 DEFB 04,04,04,04,05,05,05,05,04
C812 04 05 05 05
C816 05 04
C818 07 07 07 07 DEFB 07,07,07,07,09,09,09,09,09,04
C81C 04 09 09 09
C820 09 04
C822 0B 0B 0B 0B DEFB 0B,0B,0B,0B,04,0C,0C,0C,0C,0B
C826 04 0C 0C 0C
C82A 0C 0B
C82C 0B 0B 0B 0B DEFB 0B,0B,0B,0B,04,09,09,09,09,09,04
C830 04 09 09 09
C834 09 04
C836 07 07 07 07 DEFB 07,07,07,07,05,05,05,05,04
C83A 04 05 05 05
C83E 05 04
C840 04 04 04 04 DEFB 04,04,04,04,02,02,02,02,04
C844 04 02 02 02
C848 02 04
C84A 08 00 08 00 DEFB 08,00,08,00,00,00,00,00,00
C84E 08
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-9
ROM DATA SECTION

C8B8 07 11 13 17 C DEFB 07H,11H,13H,17H,08H \v7
C8BF 08 C
C9C8 08 0F 13 18 C DEFB 08H,0FH,13H,18H,10H \i
C9C4 10 C
C
C ; ABSOLUTE PITCH TO INTEGER TRANSLATION TABLE
C ;
C C5 01 0D 19 25 C ABSPTT: DEFB 01H,0DH,19H,25H
C9C9 31 3D 49 55 C DEFB 31H,3DH,49H,55H
C ;
C ; RHYTHM TRANSLATION TABLE
C ;
C ; EACH VALUE REPRESENTS THE NUMBER OF 32ND NOTES CONTAINED
C ; WITHIN THAT NOTE VALUE.
C ;
C C9CD 40 C RHTBLT: DEFB 40H ;DOUBLE WHOLE NOTE
C C9CE 20 C DEFB 20H ;WHOLE NOTE
C C9CF 10 C DEFB 10H ;HALF NOTE
C C9D0 08 C DEFB 08H ;QUARTER NOTE
C C9D1 04 C DEFB 04H ;EIGHTH NOTE
C C9D2 02 C DEFB 02H ;SIXTEENTH NOTE
C C9D3 01 C DEFB 01H ;THIRTY-SECOND NOTE
C ;
C ; CURRENT POSITION INCREMENT DEFAULTS TABLE
C ;
C ; THE CURRENT POSITION (CURPOS) INCREMENT USED FOR THE VARIOUS
C ; RHYTHMIC VALUES WHEN THE INCREMENT IS NOT SPECIFIED. THE
C ; FIRST ENTRY IS FOR THE DOUBLE WHOLE NOTE, THE SECOND IS FOR
C ; THE WHOLE NOTE, ETC.
C ;
C C9D4 06 06 04 02 C CPIDEF: DEFB 06,06,04,02
C C9D8 01 01 01 01 C DEFB 01,01,01,01
C ;
C ; INCREMENT TABLE FOR ALL PLAYABLE PITCHES
C ; THE HIGH ORDER BYTE IS THE INTEGER AND THE LOW ORDER BYTE
C ; IS THE FRACTIONAL PART OF THE INCREMENT. THIS INCREMENT
C ; IS THE NUMBER OF SAMPLES TO SKIP THROUGH THE WAVEFORM TABLE
C ; IN ORDER TO PRODUCE THE DESIRED PITCH. THE FRACTIONAL PARTS
C ; ARE ACCUMULATED BUT ARE TRUNCATED WHEN EACH SAMPLE IS
C ; SELECTED.
C ;
C ; \DEPHASE
C ;
C C ORG DORG=0100H
C
  ; .PHASE BASE+0100H
C
C100  C  INCTBL: DEFS 100H
C  ;
C  ; WAVEFORM TABLE
C  ; THE VALUES IN THIS TABLE REPRESENT ONE CYCLE OF A SINE
C  ; WAVE WITH THE FIRST HARMONIC.
C
C200  C  NAVFRM: DEFS 100H
C  ;
C  ; TABLE OF PITCH CUTOFF FREQUENCIES
C  ; FREQUENCIES STORED ARE THE BOUNDARY FREQUENCIES BETWEEN
C  ; TWO ADJACENT PITCHES. ONLY ONE OCTAVE IS STORED. TABLE
C  ; MUST BEGIN ON PAGE BOUNDARY.
C

000A  C  ; .RADIX 10
C  ;
C  ; FROTBBL EQU $
C300  C  00D6  C  LOWNFQ: DEFW 214 ;G - A
C302  C  00E2  C  DEFW 226 ;A - A#
C304  C  00F0  C  DEFW 240 ;A# - B
C306  C  00FE  C  DEFW 254 ;B - C
C308  C  010D  C  DEFW 269 ;C - C#
C30A  C  011D  C  DEFW 285 ;C# - D
C30C  C  012E  C  DEFW 302 ;D - D#
C30E  C  0140  C  DEFW 320 ;D# - E
C310  C  0153  C  DEFW 339 ;E - F
C312  C  0167  C  DEFW 359 ;F - F#
C314  C  017D  C  DEFW 381 ;F# - G
C316  C  0193  C  DEFW 403 ;G - G#
C318  C  01AC  C  HIFREQ: DEFW 428 ;G# - A

0010  C  ; .RADIX 16
C  ;
C  ; LETTER NAME TRANSLATION TABLE
C  ;
C  ; USED FOR TRANSLATING A LETTER NAME INTO ITS UNADJUSTED PITCH
C  ; AND ABSOLUTE PITCH. THE FIRST BYTE OF EACH ENTRY IS THE
C  ; UNADJUSTED PITCH DISPLACEMENT AND LINE/SPACE INDICATOR.
C  ; THE SECOND BYTE IS THE ABSOLUTE PITCH FOR THAT WHITE NOTE
C  ; IN THE ZERO OCTAVE.
C
C31A  C  06 00  C  LTRTBL: DEFB 06H,00H ;A
C31C  C  16 02  C  DEFB 16H,02H ;B
### MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-11

### ROM DATA SECTION

```
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C31E</td>
<td>05 03</td>
<td>DEF B 05H,03H</td>
</tr>
<tr>
<td>C320</td>
<td>15 05</td>
<td>DEF B 15H,05H</td>
</tr>
<tr>
<td>C322</td>
<td>04 07</td>
<td>DEF B 04H,07H</td>
</tr>
<tr>
<td>C324</td>
<td>14 08</td>
<td>DEF B 14H,08H</td>
</tr>
<tr>
<td>C326</td>
<td>03 0A</td>
<td>DEF B 03H,0AH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KEY UNADJUSTED PITCH DEFINITION ADDRESSES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EACH ENTRY IS THE ADDRESS OF THE TABLE WHICH DEFINES ALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OF THE UNADJUSTED PITCHES FOR THE KEY IN QUESTION. THOSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WITH A VALUE OF ZERO ARE INVALID KEYS.</td>
</tr>
<tr>
<td>C328</td>
<td>C352</td>
<td>KEYUPD: DEFN AFLMAJ</td>
</tr>
<tr>
<td>C32A</td>
<td>C35E</td>
<td>DEFN AMAJOR</td>
</tr>
<tr>
<td>C32C</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td>C32E</td>
<td>C36A</td>
<td>DEFN BFLMAJ</td>
</tr>
<tr>
<td>C32B</td>
<td>C376</td>
<td>DEFN BMAJOR</td>
</tr>
<tr>
<td>C332</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td>C334</td>
<td>C382</td>
<td>DEFN CFLMAJ</td>
</tr>
<tr>
<td>C336</td>
<td>C38E</td>
<td>DEFN CMajor</td>
</tr>
<tr>
<td>C338</td>
<td>C39A</td>
<td>DEFN CSMAJOR</td>
</tr>
<tr>
<td>C33A</td>
<td>C3A6</td>
<td>DEFN DFLMAJ</td>
</tr>
<tr>
<td>C33C</td>
<td>C3B2</td>
<td>DEFN DMajor</td>
</tr>
<tr>
<td>C33E</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td>C340</td>
<td>C38E</td>
<td>DEFN EFLMAJ</td>
</tr>
<tr>
<td>C342</td>
<td>C3A0</td>
<td>DEFN EMajor</td>
</tr>
<tr>
<td>C344</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td>C346</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td>C348</td>
<td>C306</td>
<td>DEFN FMAJOR</td>
</tr>
<tr>
<td>C34A</td>
<td>C3E2</td>
<td>DEFN FSMAJOR</td>
</tr>
<tr>
<td>C34C</td>
<td>C3EE</td>
<td>DEFN GFLMAJ</td>
</tr>
<tr>
<td>C34E</td>
<td>C3FA</td>
<td>DEFN GMAJOR</td>
</tr>
<tr>
<td>C350</td>
<td>0000</td>
<td>DEFN 0000H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNADJUSTED PITCH ARRAYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EACH ARRAY IS THE UNADJUSTED PITCH TABLE FOR ALL OF THE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>POSSIBLE KEYS. THESE ARRAYS ARE COPIED INTO 'UNATBL'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WHEN A KEY IS ESTABLISHED.</td>
</tr>
<tr>
<td>C352</td>
<td>66 36 76 05</td>
<td>AFLMAJ: DEF B 66H,36H,76H,05H,35H,75H</td>
</tr>
<tr>
<td>C356</td>
<td>35 75</td>
<td>C</td>
</tr>
<tr>
<td>C35B</td>
<td>24 64 14 83</td>
<td>DEF B 24H,64H,14H,83H,03H,26H</td>
</tr>
<tr>
<td>C35C</td>
<td>03 26</td>
<td>C</td>
</tr>
<tr>
<td>C35E</td>
<td>86 46 16 86</td>
<td>AMAJOR: DEF B 06H,06H,16H,08H,45H,15H</td>
</tr>
<tr>
<td>C362</td>
<td>45 15</td>
<td>C</td>
</tr>
</tbody>
</table>
```
MUSIC CONTROL PROGRAM MACRO-88 3.44 09-Dec-81 PAGE 1-13

ROM DATA SECTION

C3F4 24 94 14 23 C DEFB 24H,94H,14H,23H,63H,26H
C3F8 63 26 C
C3FA 86 A6 16 05 C GMAJOR: DEFB 06H,0A6H,16H,05H,0A5H,15H ;6 MAJOR
C3FE A5 15 C
C400 B5 04 74 54 C DEFB 0B5H,04H,74H,54H,03H,0A3H
C404 03 A3 C
C406 86 96 16 05 C NOKEY: DEFB 06H,96H,16H,05H,95H,15H ;NO KEY
C40A 95 15 C
C40C 84 04 14 83 C DEFB 04H,04H,14H,03H,08H
C410 03 06 C
C412 42 45 41 44 C KSFLAT: DEFB 'B', 'E', 'A', 'D' ;FLATS IN KEY SIGNATURE
C416 47 43 46 C DEFB 'G', 'C', 'F'
C418 44 43 47 44 C KSSHARP: DEFB 'F', 'C', 'G', 'D' ;SHARPS IN KEY SIGNATURE
C410 41 45 42 C DEFB 'A', 'E', 'B'

;KEY DEFINITION TABLE

; THIS TABLE IS USED TO DETERMINE THE TYPE OF ACCIDENTAL
; (SHARP OR FLAT) AND THE NUMBER OF THOSE ACCIDENTALS IN
; A KEY SIGNATURE. THE LETTER NAME IS USED AS AN INDEX
; INTO THIS TABLE. THERE ARE 6 BYTES FOR EACH LETTER NAME;
; TWO EACH FOR THE UNALTERED LETTER NAME, THE LOWERED LETTER
; NAME AND THE RAISED LETTER NAME. THE FIRST BYTE IS FOR
; MAJOR KEYS; THE SECOND FOR MINOR KEYS. THE HIGH ORDER
; THREE BITS IN EACH BYTE REPRESENT THE ACCIDENTAL TYPE
; IN THE SAME FORMAT USED IN ADJUSTED PITCH. THE LOW
; ORDER 3 BITS REPRESENT THE NUMBER OF THOSE ACCIDENTALS
; TO BE PRINTED IN THIS KEY.

C420 84 87 C KEYTBL: DEFB 84H,87H ;A-FLAT
C422 A3 00 C DEFB 0A3H,00H ;A
C424 FF A7 C DEFB 0FFH,0A7H ;A-SHARP
C426 82 85 C DEFB 82H,05H ;B-FLAT
C428 A5 A2 C DEFB 0A5H,0A2H ;B
C42A FF FF C DEFB 0FFH,0FFH ;B-SHARP
C42C 87 FF C DEFB 87H,0FFH ;C-FLAT
C42E 80 83 C DEFB 00H,03H ;C
C430 A7 A4 C DEFB 0A7H,0A4H ;C-SHARP
C432 85 FF C DEFB 05H,0FFH ;D-FLAT
C434 A2 81 C DEFB 0A2H,01H ;D
C436 FF A6 C DEFB 0FFH,0A6H ;D-SHARP
C438 83 86 C DEFB 03H,06H ;E-FLAT
C43A A4 A1 C DEFB 0A4H,0A1H ;E
C43C FF FF C DEFB 0FFH,0FFH ;E-SHARP
C43E FF FF C DEFB 0FFH,0FFH ;F-FLAT
ROM DATA SECTION

C448 81 84  C  DEFB 81H,84H  ;F
C442 A6 A3  C  DEFB 8A6H,8A3H  ;F-SHARP
C444 86 FF  C  DEFB 86H,0FFH  ;G-FLAT
C446 A1 82  C  DEFB 8A1H,82H  ;G
C448 FF A5  C  DEFB 0FFH,8A5H  ;G-SHARP

; KEY SIGNATURES

; THESE ARRAYS HAVE THE KEY SIGNATURES FOR THE KEYS OF C-FLAT MAJOR AND C-SHARP MAJOR (7 FLATS AND 7 SHARPS, RESPECTIVELY). THE FIRST TWO NAMED ARRAYS CONTAIN THE ACCIDENTALS IN THE KEY SIGNATURES IN THE UNADJUSTED PITCH FORMAT (ZERO OCTAVE). THE FOLLOWING ARRAYS CONTAIN THE OCTAVES IN WHICH THESE ACCIDENTALS ARE PRINTED DEPENDING ON THE CLEF IN USE. EVERY VALID KEY SIGNATURE USES THESE ARRAYS BY PRINTING ONLY THE NUMBER OF ACCIDENTALS NEEDED.

C44A 96  C  KEYFLAT: DEFB 96H  ;B-FLAT
C44B 84  C  DEFB 84H  ;E-FLAT
C44C 86  C  DEFB 86H  ;A-FLAT
C44D 95  C  DEFB 95H  ;D-FLAT
C44E 83  C  DEFB 83H  ;G-FLAT
C44F 85  C  DEFB 85H  ;C-FLAT
C450 94  C  DEFB 94H  ;F-FLAT

C451 84  C  KEYSHARP: DEFB 84H  ;F-SHARP
C452 A5  C  DEFB 8A5H  ;C-SHARP
C453 A3  C  DEFB 8A3H  ;G-SHARP
C454 B5  C  DEFB 8B5H  ;D-SHARP
C455 A6  C  DEFB 8A6H  ;A-SHARP
C456 A4  C  DEFB 8A4H  ;E-SHARP
C457 B4  C  DEFB 8B4H  ;B-SHARP

; OCTAVES FOR SHARPS IN ALL CLEFS

C458 01 01 01 01  C  SHTOCT EQU $TRBSQCS: DEFB 01,01,01,01,01,01,01,01
C45C 01 01 01 00  C  TRBSQCS: DEFB 01,01,01,01,01,01,01,00
C468 FF FF FF FF  C  BASSOC: DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
C464 FF FF FF 00  C  ; OCTAVES FOR FLATS IN ALL CLEFS

C468 01 01 01 01  C  FLTOCT EQU $TRBFQCS: DEFB 01,01,01,01,01,01,01,00
C46C 00 01 00 00  C  TRBFQCS: DEFB 01,01,01,01,00,01,00,00
C470 FF FF FF 00  C  BASFQCS: DEFB 0FFH,0FFH,0FFH,0FFH,0FEH,0FFH,0FEH,00
ROM DATA SECTION

C474 00 FF 00 FF 00 C
 ; PATCH VECTOR ADDRESS TABLE
 ;
C478 C47C 00 C
PTCHAD: DEFN TRBPAT
C47A C481 00 C
DEFN BASPAT
 ;
 ; KEY SIGNATURE PATCHES
 ; ONLY SHARP KEY SIGNATURES NEED PATCHING. THE FIRST NUMBER IS
 ; THE MINIMUM NUMBER OF SHARPS IN THE KEY SIGNATURE TO REQUIRE
 ; THAT PATCH. THE SECOND NUMBER IS THE X-Y COORDINATE OF THE
 ; PATCH ADDRESS RELATIVE TO THE STARTING POSITION OF THE KEY
 ; SIGNATURE AT THE TOP LINE OF THE STAFF. THE HIGH ORDER 4 BITS
 ; ARE THE X COORDINATE; THE LOW ORDER 4 BITS ARE THE Y COORDINATE.
 ;
C47C 02 22 05 53 00 C
TRBPAT: DEFB 02H,22H,05H,53H,0FFH
C480 FF C
C481 02 23 05 54 00 C
BASPAT: DEFB 02H,23H,05H,54H,0FFH
C485 FF C
 ;
 ; CLEF CHARACTER VECTOR ADDRESS TABLE
 ;
C486 C48E 00 C
CLEFAD: DEFN TRBCLF
C488 C4AC 00 00 80 00 C
DEFN 8000H
C48A 00 00 C
DEFN 8000H
 ;
 ; CLEF CHARACTER DEFINITION ARRAYS
 ; GIVEN IN FORMAT AS USED BY THE SUBROUTINE 'DISPLA'.
 ;
C48E 0F 00 01 80 00 C
TRBCLF: DEFB 0FH,SPACE,01H,TRCL12,LINLEN
C482 00 00 C
C492 00 00 C
C493 01 02 3F 00 C
DEFB TRCL22,01H,TRCL23,LINLEN-1
C497 03 3F 00 C
DEFB TRCL32,LINLEN-1
C499 04 01 05 01 00 C
DEFB TRCL41,01H,TRCL42,01H,TRCL43,LINLEN-2
C49D 06 3E 00 C
C49F 07 01 88 11 00 C
DEFB TRCL51,01H,TRCL52,01H,TRCL53,LINLEN-2
C4A3 09 3E 00 C
C4A5 0A 01 8B 11 00 C
DEFB TRCL61,01H,TRCL62,01H,TRCL63,LINLEN-1
C4A9 0C 3F 00 C
C4AB 0D 00 C
DEFB TRCL72
C4AC 0A 20 40 00 C
BASCLF: DEFB 0AH,SPACE,LINLEN
C4AF 0E 01 08 01 00 C
DEFB BSCL11,01H,BSCL12,01H,BSCL13,LINLEN-2
C4B3 08 3E 00 C
C4B5 01 02 92 3F 00 C
DEFB BSCL21,02H,BSCL23,LINLEN-1
C4B9 03 01 94 3F 00 C
DEFB BSCL32,01H,BSCL33,LINLEN-1
C4BD 05 3F 00 C
DEFB BSCL42,LINLEN-1
ROM DATA SECTION

C4BF  96  C  ;  DEFB  BSCL51  
C4C0  80 FA  C  ;  CLFDIF: DEFB  0,-6  ;DIFERENTIAL FOR CLEF IN USE  
C4C2  FD FC  C  ;  DEFB  -3,-4  ;  TREBLE,BASS,ALTO,TENOR  

C4C4  ;  PITCH CHARACTERS SUBROUTINE JUMP TABLE  
C4C6  ;  EACH WORD REPRESENTS THE ADDRESS OF THE SUBROUTINE WHICH  
C4C8  ;  FINDS THE CHARACTERS FOR A SPECIFIC NOTE RHYTHMIC VALUE.  
C4CA  ;  THE FIRST ENTRY IS THE ADDRESS OF THE ROUTINE FOR THE  
C4CB  ;  DOUBLE WHOLE NOTE; THE SECOND ENTRY IS THE ROUTINE ADDRESS  
C4CC  ;  FOR THE WHOLE NOTE; ETC.  

C4C4  CHRTBL EQU 0  
C4C6  DBLNM1: DEFW WHLCHR  
C4C8  WHOLEN: DEFW WHLCHR  
C4CA  HALFNO: DEFW WHLCHR  
C4CB  QUARTR: DEFW WHLCHR  
C4CC  EIGHTH: DEFW WHLCHR  
C4CE  SIXTEN: DEFW WHLCHR  
C4D0  THRTY2: DEFW WHLCHR  
C4D2  BLKNOH: DEFW WHLCHR  

C4D4  ;  NOTE AND ACCIDENTAL CHARACTER VECTORS  
C4D6  ;  NATURAL ON LINE  
C4D8  NATLAN: DEFB 03H,ANLN,LINLEN,ANLNL,ANLEN,ANLENL  
C4DA  48 A1  C  
C4DE  NATLAN: DEFB 03H,ANLN,LINLEN,ANLNL,ANLEN,ANLENL  
C4E0  48 A2  C  
C4E4  NATLAN: DEFB 03H,ANLN,LINLEN,ANLNL,ANLEN,ANLENL  
C4E8  48 A1  C  
C4EC  NATLAN: DEFB 03H,ANLN,LINLEN,ANLNL,ANLEN,ANLENL  
C4F0  48 A2  C  
C4F4  NATLAN: DEFB 03H,ANLN,LINLEN,ANLNL,ANLEN,ANLENL  
C4F8  48 A1  C  
C4FC  40 A7  C  

C4B8  03 A3 48 A5  C  NATSAN: DEFB 03H,ANSTNL,LINLEN,ANSHNL,LINLEN,ANSENHL  
C4FC  40 A7  C
MUSIC CONTROL PROGRAM MACRO-88 3.44 09-Dec-81 PAGE 1-1

ROM DATA SECTION

C4FE 03 A3 40 A5 C NATSAB: DEFB 03H,ANSNL,LINLEN,ANSHNL,LINLEN,ANSEN
C582 40 A9 C
C584 03 A4 40 A5 C NATSBT: DEFB 03H,ANSNL,LINLEN,ANSHNL,LINLEN,ANSEN
C588 40 A7 C
C58A 03 A4 40 A6 C NATSML: DEFB 03H,ANSNL,LINLEN,ANSHNL,LINLEN,ANSEN
C58E 40 A8 C
C510 03 A3 40 A6 C NATSMT: DEFB 03H,ANSNL,LINLEN,ANSHNL,LINLEN,ANSEN
C514 40 A9 C
C516 03 A4 40 A6 C NATSMN: DEFB 03H,ANSNL,LINLEN,ANSHNL,LINLEN,ANSEN
C51A 40 A7 C
C51C 02 9D 40 A9 C FLTLAN: DEFB 02H,AFLTNL,LINLEN,AFLEN
C520 02 9E 40 A9 C FLTLAT: DEFB 02H,AFLTNL,LINLEN,AFLen
C524 02 9E 40 AA C FLTLNL: DEFB 02H,AFLTNL,LINLEN,AFLen
C528 02 9D 40 AA C FLTLMT: DEFB 02H,AFLTNL,LINLEN,AFLen
C52C 02 AB 40 AD C FLTSAN: DEFB 02H,AFSTNL,LINLEN,AFSBN
C530 02 AB 40 AE C FLTSAT: DEFB 02H,AFSTNL,LINLEN,AFSBN
C534 02 AC 40 AD C FLTSBT: DEFB 02H,AFSTNL,LINLEN,AFSBN
C538 02 AC 40 AE C FLTSN: DEFB 02H,AFSTNL,LINLEN,AFSBN
C53C 04 AF 3F B1 C SHPLAN: DEFB 04H,ASLTON,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C540 01 B3 40 B5 C
C544 04 AF 3F B1 C SHPLAL: DEFB 04H,ASLTON,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C548 01 B3 40 B6 C
C54C 04 B8 3F B1 C SHPLBL: DEFB 04H,ASLTOL,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C550 01 B3 40 B5 C
C554 04 B8 3F B2 C SHPLNL: DEFB 04H,ASLTOL,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C558 01 B4 40 B6 C
C55C 04 B8 3F B2 C SHPLMB: DEFB 04H,ASLTOL,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C560 01 B4 40 B5 C
C564 04 AF 3F B2 C SHPLMT: DEFB 04H,ASLTOL,LINLEN-1,ASMLN,01H,ASONLN,LINLEN,ASLBN
C568 01 B4 40 B6 C
C56C 04 B7 01 B9 C SHPSAN: DEFB 04H,ASSTLN,01H,ASSTRN,LINLEN-1,ASSBLN,01H,ASSRN
C570 3F BB 01 BD C
C574 04 B7 01 B9 C SHPSAT: DEFB 04H,ASSTLN,01H,ASSTRN,LINLEN-1,ASSBLN,01H,ASSRN
C578 3F BC 01 BE C
C57C 04 BB 01 BA C SHPSBL: DEFB 04H,ASSTLL,01H,ASSTRL,LINLEN-1,ASSBL,01H,ASSBN
C580 3F BB 01 BD C
ROM DATA SECTION

C584 84 B8 01 BA C SHPSML: DEFB 04H,ASSSTL,0IH,ASSSTSL,LINLEN-1,ASSBSLL,01H,ASSBSR
C588 3F BC 01 BE C C ; DOUBLE SHARP ON LINE
C58C 01 C8 C DSHLAN: DEFB 01H,ADSLON
C58E 01 C1 C DSHML: DEFB 01H,ADSLOL
C590 02 C2 40 C4 C ; DOUBLE SHARP ON SPACE
C594 02 C2 40 C5 C DSHSAB: DEFB 02H,ADSTN,LINLEN,ADSSBL
C598 02 C3 40 C4 C DSHSBT: DEFB 02H,ADSTN,LINLEN,ADSSBL
C59C 02 C3 40 C5 C DSHSM: DEFB 02H,ADSTN,LINLEN,ADSSBL
C5A0 01 C6 C WHOLIN: DEFB 01H,NHML
C5A4 02 CA 06 C WHOLIN: DEFB 02H,NHBL
C5A6 02 CA 06 C7 C WHOSTN: DEFB 02H,NHBSBL,NEGLEN,NHBSTN
C5AA 02 C9 06 C8 C WHOSBN: DEFB 02H,NHBSBL,NEGLEN,NHBSTN
C5AE 01 CB C BWHLIN: DEFB 01H,NHBL
C5B0 02 CF 08 CD C BWHML: DEFB 02H,NHBSBL,NEGLEN,NHBSTL
C5B4 02 CF 08 CC C BWHSTN: DEFB 02H,NHBSBL,NEGLEN,NHBSTN
C5B8 02 CE 08 CD C BWHSBN: DEFB 02H,NHBSBN,NEGLEN,NHBSTL
C5B8 02 CF 08 CD C BLACK WHOLE NOTE ON LINE
C5B8 02 CF 08 CD C BLACK WHOLE NOTE ON SPACE
C5B8 02 CF 08 CD C GRAPHIC CHARACTER DEFINITIONS
C5B8 02 CF 08 CD C CHARACTERS ARE DEFINED FOR USE ON THE EXIDY SORCERER COMPUTER
C5B8 02 CF 08 CD C WHICH ALLOWS THE DEFINITION OF AN 8 BY 8 PIXEL BLOCK FOR A
C5B8 02 CF 08 CD C CHARACTER. UP TO 128 CHARACTERS MAY BE USER DEFINED. EACH
C5B8 02 CF 08 CD C ROW OF PIXELS IN THE CHARACTER CELL IS DEFINED BY A BYTE IN
C5B8 02 CF 08 CD C RAM STARTING AT FC80H, ENDING AT FFFFH. THE BYTES AT FC80H
C5B8 02 CF 08 CD C THROUGH FC07H, INCLUSIVE, CORRESPOND TO THE 'ASCII' CHARACTER
C5B8 02 CF 08 CD C '00H'. THE BYTES AT FC08H THROUGH FC0FH DEFINE 'ASCII'
C5B8 02 CF 08 CD C CHARACTER '01H', ETC. THE FIRST BYTE IN EACH CHARACTER GROUP
MUSIC CONTROL PROGRAM
ROM DATA SECTION

MACRO-80 3.44 09-Dec-81 PAGE 1-21

C4EB 02 02 02 C
C4EC 42 42 4E 72 C  DEFB 42,42,4E,72,OFF,02,02,02 ;MIDDLE WITH LINE
C4F0 FF 02 02 C
C4F4 02 00 00 00 C  DEFB 02,00,00,00,00,00,00,00 ;BOTTOM NO LINE
C4F8 00 00 00 00 C
C4FC 02 00 00 00 C  DEFB 02,00,00,00,OFF,00,00,00 ;BOTTOM WITH LINE
C700 FF 00 00 00 C

; FLAT SIGN (A#h - A#h)
;
; ON LINE (TOP CHARACTERS SAME AS NATURAL ON LINE)

C704 40 5C 62 42 C  DEFB 40,5C,62,42,42,42,4C,70 ;BOTTOM NO LINE
C708 42 44 4C 70 C
C70C 40 5C 62 42 C  DEFB 40,5C,62,42,OFF,46,4C,70 ;BOTTOM WITH LINE
C710 FF 46 4C 70 C

; ON SPACE

C714 40 40 40 40 C  DEFB 40,40,40,40,5C,62,42 ;TOP NO LINE
C718 40 5C 62 42 C
C71C 40 40 40 40 C  DEFB 40,40,40,OFF,5C,62,42 ;TOP WITH LINE
C720 FF 5C 62 42 C
C724 42 46 4C 70 C  DEFB 42,46,4C,70,00,00,00,00 ;BOTTOM NO LINE
C728 00 00 00 00 C
C72C 42 46 4C 70 C  DEFB 42,46,4C,70,OFF,00,00,00 ;BOTTOM WITH LINE
C730 FF 00 00 00 C

; SHARP SIGN (A#h - B#h)
;
; ON LINE

C734 00 00 00 00 C  DEFB 00,00,00,00,00,00,00,04 ;TOP NO LINE
C738 00 00 04 04 C
C73C 00 00 00 00 C  DEFB 00,00,00,00,OFF,00,04,04 ;TOP WITH LINE
C740 FF 00 04 04 C
C744 00 00 00 00 C  DEFB 00,00,00,00,00,00,00,07,00 ;MIDDLE LEFT NO LINE
C748 00 00 00 00 C
C74C 00 00 00 00 C  DEFB 00,00,00,00,00,00,00,00,00,00,07,00 ;MIDDLE LEFT WITH LINE
C750 FF 00 00 00 C
C754 04 04 FF 04 C  DEFB 04,04,OFF,04,04,04,04,04,04 ;MIDDLE RIGHT NO LINE
C758 04 04 FF 04 C
C75C 04 04 FF 04 C  DEFB 04,04,OFF,04,04,OFF,04,OFF,04 ;MIDDLE RIGHT WITH LINE
C760 FF 04 FF 04 C
C764 04 04 04 00 C  DEFB 04,04,04,00,00,00,00,00,00,00,00,00,00,00 ;BOTTOM NO LINE
C768 00 00 00 00 C
C76C 04 04 04 00 C  DEFB 04,04,04,00,00,00,00,00,00,00,00,00,00,00 ;BOTTOM WITH LINE
C770 FF 00 00 00 C

; ON SPACE

C774 00 00 00 00 C  DEFB 00,00,00,00,00,00,00,07,00 ;TOP LEFT NO LINE
C778 00 00 07 00 C
C77C 00 00 00 00 C  DEFB 00,00,00,00,OFF,00,00,07,00 ;TOP LEFT WITH LINE
ROM DATA SECTION

C ; ON LINE
CB14 00 0E 35 6A C DEFB 00,0E,35,6A,0FF,56,0A,C,70
CB18 FF 56 AC 70 C

C ; ON SPACE
CB1C 00 00 00 00 C DEFB 00,00,00,00,00,0E,35,6B ;TOP NO LINE
CB20 00 0E 35 6B C
CB24 00 00 00 00 C DEFB 00,00,00,0FF,0E,35,6B ;TOP WITH LINE
CB28 FF 0E 35 6B C
CB2C D5 AA DC 70 C DEFB 0D5,0AA,0DC,70,00,00,00,00 ;BOTTOM NO LINE
CB30 00 00 00 00 C
CB34 D5 AA DC 70 C DEFB 0D5,0AA,0DC,70,0FF,00,00,00 ;BOTTOM WITH LINE
CB38 FF 00 00 00 C

C ; DEPASE
C
C ; SUBROUTINE ADDRESS TABLE
C ; EACH PAIR OF BYTES REPRESENTS THE ADDRESS OF THE SUBROUTINE
C ; WHICH HANDLES A SPECIFIC SYSTEM CALL. THE SYSTEM CALL NUMBER
C ; IS USED AS AN INDEX INTO THIS TABLE. A SYSTEM CALL OF 00
C ; IS HANDLED BY THE ROUTINE AT THE FIRST ADDRESS IN THIS TABLE;
C ; A SYSTEM CALL OF 01 IS HANDLED BY THE ROUTINE AT THE SECOND
C ; ADDRESS IN THE TABLE, ETC. THE ENTRY POINT ROUTINE LOADS THE
C ; APPROPRIATE ROUTINE ADDRESS AND THEN CALLS IT.
C
C ; ORG DORG+0B00H
C
C ; .PHASE BASE+0B00H
C
CB00 C66E C JMP:BL: DEFN DEFPAR ;DEFINE PARITION
CB02 CBEE C DEFN ACTPAR ;ACTIVATE PARITION
CB04 CCEE C DEFN CLPAR ;CLEAR PARITION
CB06 C05A C DEFN CURPAR ;GET CURRENT ACTIVE PARITION NUMBER
CB08 C05E C DEFN STFCLF ;DISPLAY STAFF AND CLEF
CB0A CB60 C DEFN STAFF ;DISPLAY BLANK STAFF
CB0C CB6E C DEFN CLEF ;DISPLAY CLEF
CB0E C6FA C DEFN CLOSTF ;CLOSE STAFF
CB10 CE65 C DEFN BARTCF ;DISPLAY BAR LINE
CB12 CE65 C DEFN SETKEY ;SET KEY
CB14 CEE8 C DEFN KEYSIG ;DISPLAY KEY SIGNATURE
CB16 CF40 C DEFN GETKEY ;GET NUMBER/TYPUE ACCIDENTALS IN KEY
CB18 CF4C C DEFN RSTKEY ;RESET KEY TO NO KEY
CB1A D0BD C DEFN TIMSIG ;DISPLAY TIME SIGNATURE
CB1C D011 C DEFN SETXPO ;SET TRANSPOSITION INDICATORS
CB1E D0C6 C DEFN NERTUN ;DISPLAY TUNE BY LETTER/SCALE DEGREE
CB20 D12D C DEFN XLATUN ;TRANSLATE LETTER/SCALE TUNE
CB22 D17A C DEF NW PRTSCR ;DISPLAY TUNE TILL SCREEN FILLED
CB24 D1AB C DEF NW PRTSTF ;DISPLAY TUNE TILL END OF STAFF
CB26 D1E2 C DEF NW PRTBAR ;DISPLAY TUNE TILL END OF MEASURE
CB28 D27F C DEF NW PITCHST ;DISPLAY PITCH (SCALE DEGREE)
CB2A D281 C DEF NW PITCHLT ;DISPLAY PITCH (LETTER/ACCIDENTAL)
CB2C D288 C DEF NW PITCHUT ;DISPLAY PITCH (UNADJUSTED)
CB2E D319 C DEF NW PITCHI ;DISPLAY PITCH (INTEGER)
CB30 D54F C DEF NW ERANOT ;ERASE PREVIOUS NOTE
CB32 D592 C DEF NW ERACOL ;ERASE COLUMNS IN PARTITION
CB34 D5D7 C DEF NW ACONTL ;DISPLAY ACCIDENTAL
CB36 D7B4 C DEF NW TRNPOL ;TRANSPOSE LETTER NAME/ACCIDENTAL
CB38 D881 C DEF NW CVTSTL ;CONVERT SCALE DEGREE TO LETTER
CB3A D822 C DEF NW CVTLS ;CONVERT LETTER TO SCALE DEGREE
CB3C D86C C DEF NW CVLTA ;CONVERT LETTER TO ABSOLUTE PITCH
CB3E D89F C DEF NW CVUTL ;CONVERT UNADJUSTED TO LETTER
CB40 D930 C DEF NW CVTATI ;CONVERT ABSOLUTE TO INTEGER
CB42 D957 C DEF NW CVTITU ;CONVERT INTEGER TO UNADJUSTED
CB44 D9F1 C DEF NW MONOPH ;PLAY TUNE TABLE IN MONOPHONIC
CB46 DA48 C DEF NW PLAKEY ;PLAY KEY IDENTIFICATION
CB48 DA11 C DEF NW PAREL ;PLAY RELATIVE TO GIVEN ROOT
CB4A DA5C C DEF NW SNGTUN ;SING TUNE
CB4C DB8E C DEF NW SETOFS ;SET INTEGER OFFSET
CB4E DBE3 C DEF NW NXPIT ;LISTEN FOR NEXT PITCH
CB50 DC8F C DEF NW GENTOT ;GET NOTE
CB52 DC40 C DEF NW GETPIT ;GET PITCH
CB54 DC9A C DEF NW GETFRQ ;GET FREQUENCY
CB56 DCAD C DEF NW FRPIT ;CONVERT FREQUENCY TO PITCH
CB58 DD8E C DEF NW INITPIT ;INITIALIZE PITCH SAMPLE AREA
CB5A DD8E C DEF NW HSFRNT ;NUMBER HALF STEPS BETWEEN NOTES
CB5C D22F C DEF NW HILO ;HIGHEST AND LOWEST NOTE IN TUNE
CB5E DE18 C DEF NW ADMKEY ;GET ACCIDENTAL IN KEY FOR NOTE
CB60 DE48 C DEF NW SCRT ;SET CURRENT POSITION
CB62 DE52 C DEF NW GETCP ;GET CURRENT POSITION
CB64 DE59 C DEF NW INHIBT ;INHIBIT STORE IN TUNE TABLE
CB66 DE5F C DEF NW ENASTO ;ENABLE STORE IN TUNE TABLE
CB68 DD1E C DEF NW PRTXT ;PRINT TEXT IN PARTITION

CB6A C ENDTAG EQU $
C:
C: ; .DEPHASE
C:
C:
C: SUBTTL SYSTEM ENTRY POINTS
C: EJECT

MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-24
ROM DATA SECTION
NOTE THAT THE FOLLOWING MACRO DEFINES OP-CODES FOR THE 288 PROCESSOR THAT WERE NOT ORIGINALLY INTENDED OR RECOGNIZED. BECAUSE OF THE WAY THE PROCESSOR IS MICRO-PROGRAMMED THEY DO FUNCTION PROPERLY, HOWEVER, AND ARE VERY VALUABLE FOR PERFORMING FUNCTIONS WITH THE IX AND IY REGISTERS IN A MANNER THAT IS MUCH FASTER.

EXIT Y MACRO Y ; THIS MACRO IS USED TO EITHER INCREMENT OR DECREMENT THE HIGH OR LOW BYTE OF THE IY REGISTER AND SET THE ZERO FLAG EXCEPT WHERE NOTED, EACH SUBROUTINE RETURNS ALL REGISTERS WITH THE VALUES THEY HAD UPON ENTRY. THE EXCEPTION TO THIS IS THE (A) REGISTER WHICH IS ONLY OCCASIONALLY RETURNED UNCHANGED. THESE EXCEPTIONS ARE NOTED.

NOTE THAT ONLY REGISTERS A, BC AND DE CAN BE RETURNED WHEN CALLING MCP VIA THE ENTRY point BELOW SINCE REGISTERS HL, IX AND IY ARE 'STACKED' UPON ENTRY AND EXIT. SOME SYSTEM ROUTINES RETURN VALUES IN THESE REGISTERS FOR INTERNAL USE.
SYSTEM ENTRY POINTS

COLD START INITIALIZATION ROUTINE.

COLD: LD HL, TUNTBL
ZERO OUT ALL FIELDS

COLD START INITIALIZATION ROUTINE.

CC08 21 0000X C COLD: LD HL, TUNTBL
ZERO OUT ALL FIELDS

CC03 11 0001X C LD DE, TUNTBL+1
IN MCP RAM AREA

CC06 01 08F7 C LD BC, 08F7H
LENGTH OF RAM DATA

CC09 36 00 C LD (HL), 00H
START FIRST BYTE

CC0B ED 00 C LDIR
MOVE IT TO REST

CC0D 21 0000X C LD HL, SAFBAR
PUT IN SAFETY BAR LINE

CC10 3E FE C LD A, TTSBAR
SINGLE BAR LINE

CC12 06 00 C LD B, 08H
NUMBER TO PLACE

CC14 77 C COLD01: LD (HL), A
STORE IT

CC15 23 C INC HL
POINT TO NEXT

CC16 10 FC C DNNZ COLD01

CC18 01 0000C C LD BC, 0000CH
INITIALIZE UNADJUSTED

CC1A 11 0000X C LD DE, UNATBL
PITCH TABLE WITH

CC1E 21 C406 C LD HL, NOKEY
NO KEY

CC21 ED 00 C LDIR

CC23 21 FFF8 C LD HL, TUNTBL-TTENLEN

CC26 22 0000X C LD (SAVEIX), HL
START TUNE POINTER

CC29 21 C58C C LD HL, GRPHCS
MOVE GRAPHIC CHARACTER

CC2C 11 FC00 C LD DE, 0FC00H
DEFINITIONS

CC2F 01 0200 C LD BC, 0200H
LENGTH

CC32 ED 00 C LDIR

CC34 C3 8FFD C JP BASE-3
JUMP TO OTHER COLD START

SYSTEM ENTRY POINT

USES THE SYSTEM FUNCTION CODE GIVEN IN (A) AS AN
INDEX INTO THE SUBROUTINE ADDRESS TABLE AND PASSES
CONTROL TO THE ROUTINE AT THE ADDRESS FOUND.

CC37 FE 35 C SYSTEM: CP MAXFNC
TEST FOR VALID FUNCTION CALL

CC39 D0 C RET NC
IGNORE IT IF INVALID

CC3A E5 C PUSH HL

CC3B DD E5 C PUSH IX
SAVE INDEX REGISTERS

CC3D FD E5 C PUSH IY

CC3F 21 C800 C LD HL, JMPTBL
POINT TO START OF JUMP TABLE

CC42 D5 C PUSH DE
SAVE (DE)

CC43 9F C LD E, A
USE FUNCTION CALL NUMBER AS

CC44 16 00 C LD D, 00H
INDEX INTO TABLE

CC46 19 C ADD HL, DE
ADD TWICE SINCE EACH ENTRY

CC47 19 C ADD HL, DE
IS 2 BYTES LONG

CC48 5E C LD E, (HL)
GET ADDRESS OF ROUTINE THAT

CC49 23 C INC HL
HANDLES THIS FUNCTION CALL
SYSTEM ENTRY POINTS

CC4A  56  C  LD  D,(HL) ;SAVE UNTIL READY
CC4B  ED 53 0000  C  LD  (TEMP01),DE ;RESTORE ORIGINAL (DE)
CC4F  D1  C  POP  DE ;SAME VALUE AS BEFORE
CC50  DD 2A 0000  C  LD  IX,(SAVEIX) ;SET IX AND IY REGISTERS TO
CC54  FD 2A 0000  C  LD  IY,(SAVEIY) ;SAME VALUE AS BEFORE
CC5B  CC60  C  LD  HL,RETURN ;AFTER EACH FUNCTION CALL
CC5C  21  CC60  C  PUSH  HL ;RETURN BACK HERE
CC5D  FD 2A 0000  C  LD  IX,(SAVEIX) ;SAME VALUE AS BEFORE
CC5F  E9  C  JP  (HL) ;CALL ROUTINE
CC60  DD 22 0000  C  RETURN:  LD  (SAVEIX),IX ;SAVE IX AND IY REGISTERS AFTER
CC64  FD 22 0000  C  LD  (SAVEIY),IY ;EVERY FUNCTION CALL
CC68  CC6A  C  POP  IX
CC6C  CC6D  C  POP  HL
CC6E  C5  C  RET

SUBTTL SCREEN PARTITION CONTROL

; DEFPAR
;
; DEFINE PARTITION
; SETS UP PARTITION TABLE ENTRY FOR THE PARTITION NUMBER
; SPECIFIED. THE PARTITION NUMBER MAY REFER TO A NEW
; PARTITION OR REDEFINE AN OLD ONE. THE PARTITION IS
; DEFINED BY SUPPLYING AN ADDRESS TO A PARTITION DEFINITION
; BLOCK WHICH IS IN THE FOLLOWING FORMAT:
;
; BYTE  MEANING
; 1  NUMBER OF THE PARTITION BEING DEFINED
; 2  X COORDINATE OF TOP LEFT CORNER OF PARTITION
; RELATIVE TO 1
; 3  Y COORDINATE OF TOP LEFT CORNER OF PARTITION
; RELATIVE TO 1
; 4  LENGTH OF LINES IN THE PARTITION
; 5  NUMBER OF LINES IN THE PARTITION
; 6  LINE NUMBER FROM THE TOP OF THE PARTITION
; ON WHICH THE TOP LINE OF THE STAFF IS DISPLAYED
;
; GIVEN:  (BC) - ADDRESS OF PARTITION DEFINITION BLOCK
; RETURNS:  (A) - COMPLETION CODE
;          = 00 - SUCCESSFUL
;          = FF - INVALID PARTITION NUMBER
; (DE) - BASE ADDRESS OF PARTITION ON SCREEN
; (IX) - BASE ADDRESS OF PARTITION TABLE ENTRY
;          = 0000 IF PARTITION NUMBER IS INVALID

CC6E  C5  C  DEFPAR: PUSH BC
SCREEN PARTITION CONTROL

CC6F 68 C LD H,B ;SET (HL) TO PDB
CC70 69 C LD L,C
CC71 46 C LD B,(HL) ;GET PARTITION NUMBER
CC72 23 C INC HL ;POINT TO X COORDINATE
CC73 EB C EX DE,HL ;SAVE HL
CC74 CD CD0E C CALL PARMSK ;GET PARTITION MASKS
CC77 FD 21 0000 C LD IY,0000H ;SET TO ZERO IF INVALID
CC7B FE FF C CP 8 ;TEST FOR INVALID NUMBER
CC7D 28 3D C JR Z,D FPRI83 ;EXIT IF INVALID
CC7F 81 C OR C ;SET PARTITION NUMBER BIT HIGH
CC80 77 C LD (HL),A ;IN MASK FOR DEFINED FLAG
CC81 EB C EX DE,HL ;RESTORE PCB ADDRESS
CC82 78 C LD A,B ;GET PARTITION NUMBER
CC83 32 0000X C LD (PARACT),A ;ACTIVATE PARTITION
CC84 FD 21 0000X C LD IY,PARO1 ;POINT TO START OF PARTITION TABLE
CC89 3D C DEC A ;MAKE NUMBER RELATIVE TO 0
CC8B 28 08 C JR Z,DPFR02 ;JUMP IF FIRST PARTITION
CC8D 47 C LD B,A ;ELSE COUNT PARTITIONS IN (B)
CC8E 11 0008 C LD DE,PARTL ;PARTITION TABLE ENTRY LENGTH
CC91 FD 19 C DFPR01: ADD IY,DE ;POINT TO NEXT ENTRY IN TABLE
CC93 10 FC C DJNZ DFPR01 ;CONTINUE UNTIL FOUND
CC95 46 C DFPR02: LD B,(HL) ;GET X COORDINATE
CC96 23 C INC HL ;POINT TO Y COORDINATE
CC97 4E C LD C,(HL) ;GET Y COORDINATE
CC98 23 C INC HL ;POINT TO LINE LENGTH
CC99 11 F000 C LD DE,SCREEN ;X-Y COORDINATE OF ENTIRE SCREEN
CC9C EB C EX DE,HL ;SEND BASE ADDRESS OF SCREEN IN (HL)
CC9D CD DE01 C CALL XYCORD ;CALCULATE ADDRESS OF COORDINATE
CCA0 FD 75 00 C LD (IY+PARADL),L ;STORE BASE ADDRESS OF PARTITION
CCA3 FD 74 01 C LD (IY+PARADH),H
CCA6 EB C EX DE,HL ;SET (HL) TO PDB AGAIN
CCA7 7E C LD A,(HL) ;GET LINE LENGTH
CCA8 FD 77 02 C LD (IY+PARLH),A ;PUT IT IN PARTITION TABLE
CCA9 23 C INC HL ;POINT TO NUMBER OF LINES
CCAC 7E C LD A,(HL) ;GET NUMBER OF LINES
CCAD FD 77 03 C LD (IY+PARLIN),A ;STORE IT IN PARTITION TABLE
CC80 23 C INC HL ;POINT TO STAFF ORIENTATION
CC81 7E C LD A,(HL) ;GET STAFF ORIENTATION
CCB2 FD 77 04 C LD (IY+STFORD),A ;STORE IT
CCB5 AF C XOR A ;ZERO OUT NEXT 2 FIELDS
CCB6 FD 77 05 C LD (IY+CURPOS),A
CCB9 FD 77 06 C LD (IY+CLFCD),A
CCBC C1 C DFPR03: POP BC
CCBD C9 C RET
  C ;
  C ; ACTPAR
  C ;
SCREEN PARTITION CONTROL

C  ; ACTIVATE PARTITION
C  ; ACTIVATES THE GIVEN PARTITION SO THAT ALL OTHER FUNCTIONS
C  ; WITH IMPLICIT PARTITION REFERENCES WILL NOT AFFECT THE
C  ; NEWLY ACTIVATED ONE. NOTE THAT DEFINING A PARTITION VIA
C  ; 'DEFPAR' ALSO ACTIVATES THAT PARTITION, SO IT IS NOT
C  ; NECESSARY TO ACTIVATE A PARTITION AFTER IT IS DEFINED.
C  ; IT IS ONLY NECESSARY TO ACTIVATE A PARTITION WHEN A
C  ; DIFFERENT ACTIVE PARTITION IS DESIRED.
C  ;
C  ; GIVEN:  (B) - PARTITION NUMBER (RELATIVE TO 1)
C  ; RETURNS: (A) - COMPLETION CODE
C  ;   = 00H - SUCCESSFUL
C  ;   = FFH - PARTITION NOT DEFINED OR PARTITION NUMBER IS INVALID
C  ; (HL) - BASE ADDRESS OF PARTITION
C  ; (UNCHANGED IF PARTITION NUMBER INVALID)
C  ; (IY) - ADDRESS OF NEW PARTITION TABLE ENTRY
C  ;   = 0000 IF PARTITION NUMBER INVALID OR NOT DEFINED
C  ;

CCBE  C5  C ACTPAR: PUSH BC
CCBF  D5  C PUSH DE
CCBG  CD CD3E C CALL PARAMSK ;GET PARTITION MASKS
CCCH  FE FF  C CP  0FFH ;TEST FOR INVALID NUMBER
CCCI  20 03 C JR Z,ACTP01 ;EXIT IF INVALID
CCCJ  A1  C AND C ;TEST UNDER MASK
CCCS  20 08 C JR NZ,ACTP02 ;JUMP IF PARTITION IS DEFINED
CCCT  3E FF C ACTP01: LD A,0FFH ;ELSE LOAD NOT DEFINED FLAG
CCCU  FD 01 0000 C LD IY,0000H ;SET TO 0 IF INVALID
CCCV  18 19 C JR ACTP05 ; AND EXIT
CCCW  21 0000X C ACTP02: LD HL,PAR01 ;POINT TO BEGINNING OF TABLE
CCCX  78  C LD A,B ;GET PARTITION NUMBER
CCCY  32 0000X C LD (PARACT),A ;STORE PARTITION NUMBER
CCCZ  30  C DEC A ;MAKE RELATIVE TO 0
CCCa  28 07 C JR Z,ACTP04 ;JUMP IF FIRST PARTITION
CCCb  47  C LD B,A
CCCc  11 0088 C LD DE,PARTEL ;PARTITION TABLE ENTRY LENGTH
CCCd  19  C ACTP03: ADD HL,DE ;POINT TO NEXT ENTRY
CCCe  18 FD C DJNZ ACTP03
CCCf  E5  C ACTP04: PUSH HL ;NEW TABLE ENTRY ADDRESS
CCCg  F0 F1  C POP IY
CCCh  5E  C LD E,(HL) ;LOAD BASE ADDRESS USING
CCCi  23  C INC HL ;(HL) SINCE IT IS FASTER
CCCj  56  C LD D,(HL)
CCCk  EB  C EX DE,HL ;RETURN ADDRESS IN (HL)
CCCl  AF  C XOR A ;SET SUCCESSFUL CODE
CCCm  D1  C ACTP05: POP DE
CCCn  C1  C POP BC
CCCp  C9  C RET
SCREEN PARTITION CONTROL

CLEAR PARTITION

BLANKS OUT GIVEN PARTITION ON SCREEN. NOTE THAT BLANKS HAVE
TO BE ACTUALLY WRITTEN TO EACH BYTE OF THE PARTITION SINCE
THE EXECUTION OF A SIMPLE FORM FEED (0CH) CAUSES THE SORCERER
STANDARD GRAPHIC CHARACTER SET TO BE RE-INITIALIZED. ALSO
NOTE THAT THE PARTITION CLEARED DOES NOT NECESSARILY HAVE TO
BE THE ACTIVE ONE. THIS ROUTINE MAY ALSO BE USED TO CLEAR
THE ENTIRE SCREEN BY PASSING THE VALUE 0FFH IN LIEU OF THE
PARTITION NUMBER.

GIVEN: (B) - PARTITION TO BE CLEARED (RELATIVE TO 1)
01 - PARTITION NUMBER 1, ETC
00H - CLEAR ENTIRE SCREEN

RETURNS: (A) - COMPLETION CODE
00H - SUCCESSFUL
FFH - PARTITION NOT DEFINED OR PARTITION
NUMBER IS INVALID

CLRPAR: PUSH BC
PUSH DE
PUSH HL
LD A,B
LD DE,SCREEN
LD BC,LINLEN-1
CLRPAR01: CALL PARMSK
LD A,NUMLIN
JR CLRP81
LD & ,PART01
LD B ,A
LD DE,PARTEL
ADD HL,DE
LD A,0FFH
JR CLRP04
JR CLRP07
LD (& ,PART01)
ADD HL,DE
LD A,0FFH
JR CLRP04
LD A,B
DEC A
LD B,A
JR CLRP07
JR CLRP02
JR CLRP02
LD HL,PART01
LD HL,PARTEL
LD DE,PARTEL
LD E,(HL)
CLRP03: ADD HL,DE
CLRP05: CALL PARMSK
CLRP06: ADD HL,DE
ADD HL,DE
SCREEN PARTITION CONTROL

GIVEN:  
(B) - PARTITION NUMBER

RETURNS:  
(A) - TEST MASK FOR THE GIVEN PARTITION NUMBER
        = FF - INVALID PARTITION NUMBER

(C) - PARTITION NUMBER

(C) - BIT MAP FOR THE SET OF PARTITIONS TO WHICH
       THE GIVEN PARTITION BELONGS

(HL) - ADDRESS OF BIT MAP IN (C)
SCREEN PARTITION CONTROL

C

PARMSK: LD A,B ;GET PARTITION NUMBER
C3F 48 C

LD C,B ;SAVE IT IN (C)
C40 FE 11 C

CP MAXPAR+1 ;TEST FOR INVALID NUMBER
C42 30 03 C

JR C,PRMK01 ;JUMP IF VALID
C44 3E FF C

LD A,OFFH ;ELSE LOAD INVALID FLAG
C46 C9 C

RET ;AND EXIT
C47 21 0081 C

PRMK01: LD HL,PARBIT+1 ;BIT MAP FOR PARTITIONS 1 TO 8
C4A FE 09 C

CP 09H ;IS IN 2ND BYTE
C4C 30 03 C

JR C,PRMK02 ;JUMP IF IN GROUP 1
C4E D6 08 C

SUB 08H ;ELSE ADJUST FOR GROUP 2
C50 28 C

DEC HL ;POINT TO BIT MAP FOR GROUP 2
C52 47 C

PRMK02: LD B,A ;GET (ADJUSTED) PARTITION NUMBER
C54 AF C

XOR A ;ZERO MASK
C56 37 C

SCF ;SET CARRY FLAG FOR MASK BIT
C58 17 C

PRMK03: RLA ;MOVE BIT TO THE POSITION FOR THE
C5A 10 FD C

DJNZ PRMK03 ;CORRESPONDING PARTITION NUMBER
C5C 41 C

LD B,C ;RESTORE PARTITION NUMBER
C5E 4E C

LD C,(HL) ;GET BIT MAP MASK
C60 C9 C

RET

C GET CURRENT ACTIVE PARTITION
C ;RETURNS THE PARTITION NUMBER THAT IS CURRENTLY ACTIVE.
C C GIVEN: NONE
C C RETURNS: (A) - PARTITION NUMBER (RELATIVE TO 1)
C C
C CURPAR
C C
C GET CURRENT ACTIVE PARTITION
C ;RETURNS THE PARTITION NUMBER THAT IS CURRENTLY ACTIVE.
C C C GIVEN: NONE
C C C RETURNS: (A) - PARTITION NUMBER (RELATIVE TO 1)
C C
C STFCLF
C C CLEAR ACTIVE PARTITION AND DISPLAY STAFF WITH CLEF.
C C C GIVEN: (D) - STAFF ORIENTATION
C C C C - CLEF CODE
C C C (IY) - ADDRESS OF PARTITION TABLE ENTRY
C C C RETURNS: NONE
C C
C C STFCLF: PUSH BC
MUSIC CONTROL PROGRAM  MACRO-80 3.44  09-Dec-81  PAGE  1-33

MUSIC SYMBOL DISPLAY

CD5F  D5  C  PUSH  DE
CD60  ED  C  PUSH  HL
CD61  FD  C+  DEFB  0FDH  ; INCREMENT OR DECREMENT THE
CD62  24  C+  DEFB  INCYH  ; HIGH OR LOW BYTE OF THE IY
CD63  FD  C+  DEFB  0FDH  ; INCREMENT OR DECREMENT THE
CD64  25  C+  DEFB  DECYH  ; HIGH OR LOW BYTE OF THE IY
CD65  28  15  C  JR  Z,STC01 : RETURN IF INVALID
CD67  FD  70  04  C  LD  (IY+STFORM),B  ; STORE NEW STAFF ORIENTATION
CD6A  3A  0000X  C  LD  A,(PARTACT)  ; GET PARTITION NUMBER
CD6B  47  C  LD  B,A  ; SEND IN (B)
CD6E  CD  CCEE  C  CALL  CLRPAR  ; CLEAR THE PARTITION
CD71  CD  CD00  C  CALL  STAFF  ; DISPLAY STAFF
CD74  FD  36  05  02  C  LD  (IY+CURPOS)+2H  ; SET CURRENT POSITION FOR CLEF
CD78  41  C  LD  B,C  ; LOAD CLEF CODE
CD79  CD  CDBE  C  CALL  CLEF  ; DISPLAY CLEF
CD7C  E1  C  STC01:  POP  HL
CD7D  D1  C  POP  DE
CD7E  C1  C  POP  BC
CD7F  C9  C  RET

C ;
C ; STAFF
C ;
C ; DISPLAY BLANK STAFF
C ; ASSUMES (IY) IS SET TO PARTITION TABLE ENTRY FOR WHICH STAFF
C ; IS TO BE DISPLAYED AND SETS CURRENT POSITION TO ZERO.
C ; THE STAFF IS DRAWN STARTING ON THE LINE INDICATED AT IY+STFORM.
C ;
C ; GIVEN:  (IY) - ADDRESS OF PARTITION TABLE ENTRY
C ; RETURNS:  NONE
C ;
CD80  C5  C  STAFF:  PUSH  BC
CD81  D5  C  PUSH  DE
CD82  E5  C  PUSH  HL
CD83  FD  C+  DEFB  0FDH  ; INCREMENT OR DECREMENT THE
CD84  24  C+  DEFB  INCYH  ; HIGH OR LOW BYTE OF THE IY
CD85  FD  C+  DEFB  0FDH  ; INCREMENT OR DECREMENT THE
CD86  25  C+  DEFB  DECYH  ; HIGH OR LOW BYTE OF THE IY
CD87  28  31  C  JR  Z,STAFF02  ; JUMP IF INVALID
CD89  AF  C  XOR  A  ; SET CURRENT POSITION
CD9A  FD  77  05  C  LD  (IY+CURPOS),A  ; TO BEGINNING OF LINE
CD9D  FD  7E  04  C  LD  A,(IY+STFORM)  ; FIND TOP OF STAFF
CD98  CD  CD64  C  CALL  SACALC
CD93  E5  C  PUSH  HL  ; SAVE STAFF ADDRESS
MUSIC CONTROL PROGRAM
MUSIC SYMBOL DISPLAY

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode</th>
<th>Operation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD94</td>
<td>36 97</td>
<td>LD (HL),VLINE</td>
<td>PRINT FIRST BAR</td>
</tr>
<tr>
<td>CD96</td>
<td>03</td>
<td>INC HL</td>
<td></td>
</tr>
<tr>
<td>CD97</td>
<td>36 98</td>
<td>LD (HL),HLINE</td>
<td>PRINT FIRST CHAR OF LINE</td>
</tr>
<tr>
<td>CD99</td>
<td>66 00</td>
<td>LD B,00H</td>
<td>SET UP TO PRINT</td>
</tr>
<tr>
<td>CD9B</td>
<td>FD 4E 02</td>
<td>LD C,(IY+PARLEN)</td>
<td>WHOLE TOP LINE</td>
</tr>
<tr>
<td>CD9E</td>
<td>C5</td>
<td>PUSH BC</td>
<td>SAVE WHOLE LENGTH</td>
</tr>
<tr>
<td>CD9F</td>
<td>0D</td>
<td>DEC C</td>
<td></td>
</tr>
<tr>
<td>CD90</td>
<td>0D</td>
<td>DEC C</td>
<td></td>
</tr>
<tr>
<td>CD91</td>
<td>54</td>
<td>LD D,H</td>
<td></td>
</tr>
<tr>
<td>CD92</td>
<td>50</td>
<td>LD E,L</td>
<td></td>
</tr>
<tr>
<td>CD93</td>
<td>13</td>
<td>INC DE</td>
<td></td>
</tr>
<tr>
<td>CD94</td>
<td>ED B8</td>
<td>LDIR</td>
<td>PRINT TOP LINE</td>
</tr>
<tr>
<td>CD96</td>
<td>36 97</td>
<td>LD (HL),VLINE</td>
<td>PRINT LAST BAR</td>
</tr>
<tr>
<td>CD98</td>
<td>C1</td>
<td>POP BC</td>
<td>FULL LINE LENGTH</td>
</tr>
<tr>
<td>CD99</td>
<td>D1</td>
<td>POP DE</td>
<td>BEGINNING OF FIRST LINE</td>
</tr>
<tr>
<td>CD9A</td>
<td>3E 04</td>
<td>LD A,04H</td>
<td>FOUR MORE LINES TO DO</td>
</tr>
<tr>
<td>CD9C</td>
<td>19</td>
<td>ADD HL,DE</td>
<td>POINT TO NEXT LINE</td>
</tr>
<tr>
<td>CD9D</td>
<td>EB</td>
<td>EX DE,HL</td>
<td></td>
</tr>
<tr>
<td>CD9E</td>
<td>D5</td>
<td>PUSH DE</td>
<td>SAVE START ADDRESS</td>
</tr>
<tr>
<td>CD9F</td>
<td>C5</td>
<td>PUSH BC</td>
<td>SAVE LENGTH</td>
</tr>
<tr>
<td>CD9A</td>
<td>ED B0</td>
<td>LDIR</td>
<td>PRINT NEXT LINE</td>
</tr>
<tr>
<td>CD9B</td>
<td>C1</td>
<td>POP BC</td>
<td>RESTORE FULL LENGTH</td>
</tr>
<tr>
<td>CD9C</td>
<td>D1</td>
<td>POP DE</td>
<td>RESTORE START ADDRESS</td>
</tr>
<tr>
<td>CD9E</td>
<td>3D</td>
<td>DEC A</td>
<td></td>
</tr>
<tr>
<td>CD9F</td>
<td>20 F2</td>
<td>JR NZ,STAF01</td>
<td></td>
</tr>
<tr>
<td>CD9A</td>
<td>E1</td>
<td>STAF02: POP HL</td>
<td></td>
</tr>
<tr>
<td>CD9B</td>
<td>D1</td>
<td>POP DE</td>
<td></td>
</tr>
<tr>
<td>CD9C</td>
<td>C1</td>
<td>POP BC</td>
<td></td>
</tr>
<tr>
<td>CD9D</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
</tbody>
</table>

CLEF
DISPLAY CLEF AT CURRENT POSITION ON STAFF IN ACTIVE PARTITION AND INCREMENT CURRENT POSITION. TYPE OF CLEF TO DISPLAY IS GIVEN AND STORED IN PARTITION TABLE.

GIVEN: (B) - CLEF CODE
(C02) - ADDRESS OF PARTITION TABLE ENTRY
RETURNS: NONE

CLEF: PUSH BC
PUSH HL
EXITY INCYH ; TEST FOR VALID PARTITION
DEFB 0F0H ; INCREMENT OR DECREMENT THE
MUSIC CONTROL PROGRAM

MUSIC SYMBOL DISPLAY

CDC2 24 C4 DEFB INCYH ; HIGH OR LOW BYTE OF THE IY
C EXTY DECYH
CDC3 FD C4 DEFB 0FDH ; INCREMENT OR DECREMENT THE
CDC4 25 C4 DEFB DECYH ; HIGH OR LOW BYTE OF THE IY
CDC5 28 2F C JR Z,CLEFO3 ; JUMP IF INVALID
CDC7 FD 7E 04 C LD A,(IY+STFORM) ; FIND STAFF ADDRESS
CDCA CD DE64 C CALL SACALC
CDE0 11 0040 C LD DE,LINELEN
CD0 27 C AND A ;RESET CARRY FLAG
CD01 ED 52 C SBC HL,DE ; START 1 LINE ABOVE STAFF
CD03 EB C EX DE,HL ; KEEP ADDRESS IN (DE)
CD04 21 C486 C LD HL,CLEFAD ; POINT TO CLEF ADDRESS TABLE
CD07 78 C LD A,B ; GET CLEF CODE
CD08 FE 02 C CP MAXCLF ; TEST FOR INVALID CLEF
CD0A 30 1A C JR NC,CLEF03 ; EXIT IF INVALID
CD0C FD 77 06 C LD (IY+CFCOD),A ; STORE IT IN PARTITION TABLE
CD0F B7 C OR A
CDE8 28 04 C JR Z,CLEFO2
CDE2 23 C CLEFO1: INC HL ; USE CLEF CODE AS INDEX INTO
CDE3 23 C INC HL ; TABLE OF CLEF CHARACTER
CDE4 10 FC C DJNZ CLEFO1 ; VECTOR ADDRESSES
CDE6 4E C CLEFO2: LD C,(HL) ; GET ADDRESS OF CLEF CHARACTER
CDE7 23 C INC HL ; VECTOR
CDE8 46 C LD B,(HL)
CDE9 68 C LD H,B ; PUT ADDRESS IN (HL) FOR
CDEA 69 C LD L,C ; CALL TO DISPLAY
CDEB CD DEA1 C CALL DISPLA ; DISPLAY THE CLEF
CDEF FD 7E 05 C LD A,(IY+CURPOS) ; UPDATE CURRENT POSITION
CDF1 C6 04 C ADD A,04H
CDF3 FD 77 05 C LD (IY+CURPOS),A
CDF6 E1 C CLEFO3: POP HL
CDF7 D1 C POP DE
CDF8 C1 C POP BC
CDF9 C9 C RET
C ;
C ; CLOSTF
C ;
C ; CLOSE STAFF
C ; ERASE UNUSED PORTION OF STAFF AND CLOSE IT WITH EITHER A
C ; SINGLE OR DOUBLE BAR LINE.
C ;
C ; GIVEN: (B) - SINGLE/DOUBLE BAR FLAG
C ; (IY) - ADDRESS OF PARTITION TABLE ENTRY
C ; RETURNS: NONE
C ;
CDFA C CLOSTF: EXTY INCYH ; TEST FOR VALID PARTITION
CDF0A FD C DEFB 0FDH ; INCREMENT OR DECREMENT THE
MUSIC SYMBOL DISPLAY

CDF8 24 C+ DEFB INCYH ; HIGH OR LOW BYTE OF THE IY
CDFC FD C+ DEFB OFDH ; INCREMENT OR DECREMENT THE
CDFD 25 C+ DEFB DECYH ; HIGH OR LOW BYTE OF THE IY
CDEF C8 C RET Z ;RETURN IF NOT VALID
CDFS CD CE06 C CALL ERASE ; ERASE EXCESS STAFF
CE02 CD CE2F C CALL ENDSTF ; DISPLAY BAR
CE05 C9 C RET

CE06 C5 C ERASE: PUSH BC
CE07 D5 C PUSH DE
CE08 E5 C PUSH HL
CE09 FD 7E 04 C LD A, (IY*STFO RN) ; COMPUTE STAFF ADDRESS
CE0C CD DE44 C CALL SACALC
CE0F FD 7E 02 C LD A, (IY+PARLEN) ; GET PARTITION LENGTH
CE12 FD 46 05 C LD B, (IY+CURPOS) ; GET CURRENT POSITION
CE15 90 C SUB B ; GET DIFFERENCE
CE16 28 13 C JR Z, ERAS03 ; EXIT IF NONE
CE18 38 11 C JR C, ERAS03
CE1A 0E 05 C LD C, 05 ; NUMBER OF LINES TO ERASE
CE1C 11 0040 C LD D E, LINLEN
CE1F 47 C ERAS01: LD B, A ; NUMBER OF CHARACTERS TO ERASE
CE20 E5 C PUSH HL
CE21 36 20 C ERAS02: LD (HL), SPACE ; ERASE CHAR
CE23 23 C INC HL
CE24 10 FB C DJNZ ERAS02 ; B=0 AT END OF EACH LINE
CE26 E1 C POP HL
CE27 19 C ADD HL, DE ; POINT TO NEXT LINE
CE28 60 C DEC C ; ERASE NUMBER OF
CE29 20 F4 C JR NZ, ERAS01 ; LINES IN C
CE2B E1 C ERAS03: POP HL
CE2C D1 C POP DE
CE2D C1 C POP BC
CE2E C9 C RET

CE ;
CE ; ENDSTF
CE ;
CE ; END OF STAFF
MUSIC CONTROL PROGRAM

MUSIC SYMBOL DISPLAY

; PRINT SINGLE OR DOUBLE BAR AT END OF STAFF AND PUT
; BAR LINE/DRAW BAR LINE IN THE TUNE TABLE.
; GIVEN: (ED) - SINGLE/DRAW BAR FLAG
; = 00 - DRAW BAR
; = NON-ZERO - SINGLE BAR
; (IX) - ADDRESS OF CURRENT NOTE IN TUNE TABLE
; RETURNS: (IX) - ADDRESS OF BAR LINE IN TUNE TABLE

CE2F C5 C ENDFST: PUSH BC
CE30 D5 C PUSH DE
CE31 E5 C PUSH HL
CE32 FD 7E 04 C LD A,(IY+STFORN) ;GET SCREEN ADDRESS
CE35 CD DE64 C CALL SACALC
CE38 0E 97 C LD C,VLINE
CE3A 78 C LD A,8 ;GET SINGLE/DRAW BAR FLAG
CE3B A7 C AND A
CE3C 20 02 C JR NZ,NDST01 ; JUMP IF SINGLE
CE3E 0E 9C C LD C,DBLBAR
CE40 11 0040 C NDST01: LD DE,LINLEN
CE43 06 05 C LD B,05H ;NUMBER OF LINES
CE45 71 C NDST02: LD (HL),C ;PRINT BAR CHARACTER
CE46 19 C ADD HL,DE ;POINT TO NEXT LINE
CE47 10 FC C DJNZ NDST02
CE49 06 04 C LD B,TTELEN ;TUNE TABLE ENTRY LENGTH
CE4B 79 C LD A,C ;SINGLE OR DOUBLE BAR?
CE4C 16 FE C LD D,TTSBAR ;FIRST ASSUME SINGLE BAR
CE4E FE 9C C CP DBLBAR ;TEST FOR DOUBLE BAR
CE50 20 02 C JR NZ,NDST03 ;JUMP IF NOT DOUBLE BAR
CE52 16 FF C LD D,TDBAR ;LOAD DOUBLE BAR FLAG
CE54 3A 0000X C NDST03: LD A,(INKSTO) ;INHIBIT STORE IN TUNE TABLE?
CE57 A7 C AND A ;DON'T STORE IF NON-ZERO
CE58 20 07 C JR NZ,NDST05
CE5A DD 72 04 C NDST04: LD (IX+TTELEN),D ;PUT BAR LINE IN TUNE TABLE
CE5D DD 23 C INC IX ;EXIT WITH (IX) POINTING TO
CE5F 10 F9 C DJNZ NDST04 ;BEGINNING OF BAR LINE
CE61 E1 C NDST05: POP HL
CE62 D1 C POP DE
CE63 C1 C POP BC
CE64 C9 C RET

; BARL

; PRINT A BAR LINE ACROSS THE STAFF AND BAR INDICATOR IN
; THE TUNE TABLE. ALSO INCREMENTS CURRENT POSITION.
; GIVEN: (IY) - ADDRESS OF PARTITION TABLE ENTRY
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-38

MUSIC SYMBOL DISPLAY

C ; (IX) - ADDRESS OF CURRENT NOTE IN TUNE TABLE
C ; RETURNS: (IX) - ADDRESS OF BAR LINE IN TUNE TABLE
C ;
CE65  C5  C  BARLIN: PUSH BC
CE66  D5  C  PUSH DE
CE67  E5  C  PUSH HL
C  EXTIY INCYH ; TEST FOR VALID PARTITION
CE68  FD  C+  DEFB 0FH ; INCREMENT OR DECREMENT THE
CE69  24  C+  DEFB INCYH ; HIGH OR LOW BYTE OF THE IV
C  EXTIY DECYH
CE6A  FD  C+  DEFB 0FH ; INCREMENT OR DECREMENT THE
CE6B  25  C+  DEFB DECYH ; HIGH OR LOW BYTE OF THE IV
CE6C  28 33  C  JR  Z, BLIN03 ; JUMP IF INVALID
CE6D  FD 7E 05  C  LD  A),(IX+CURPOS) ; TEST FOR END OF STAFF
CE6E  FD 5E 02  C  LD  E),(IX+PARLEN) ; GET LENGTH OF LINES
CE6F  BB  C  OP  E ; COMPARE
CE70  30 2A  C  JR  NC, BLIN03 ; EXIT IF NO ROOM
CE71  CD  DE64  C  CALL SACALC ; FIND STAFF ADDRESS
CE72  36 99  C  LD  (HL),BARLN1 ; PRINT TOP CHAR
CE73  11 0040  C  LD  DE, LINLEN
CE74  06 83  C  LD  B, 83
CE75  3E 9A  C  LD  A, BARLN2 ; MID CHAR
CE76  19  C  BLIN01: ADD HL, DE ; POINT TO NEXT LINE
CE77  77  C  LD  (HL), A ; PRINT MID CHAR
CE78  10  FC  C  DJNZ BLIN01 ; 3 TIMES
CE79  19  C  ADD HL, DE
CE7A  36 9B  C  LD  (HL), BARLN3 ; PRINT BOTTOM CHAR
CE7B  FD 34 05  C  INC (IX+CURPOS) ; INCREMENT CURRENT POSITION
CE7C  3A  0000  C  LD  A,(INHST0) ; TEST INHIBIT STORE FLAG
CE7D  A7  C  AND A ; DON'T STORE IF NON-ZERO
CE7E  20 0B  C  JR  NZ, BLIN03
CE7F  3E FE  C  LD  A, TTSBAR ; LOAD SINGLE BAR LINE FLAG
CE80  06 84  C  LD  B, TTLEN ; TUNE TABLE ENTRY LENGTH
CE81  DD 77 84  C  BLIN02: LD (IX+TTLEN), A ; PUT BAR LINE IN TUNE TABLE
CE82  DD 23  C  INC IX
CE83  E9  C  DJNZ BLIN02
CE84  18  F9  C  POP HL
CE85  E1  C  BLIN03: POP DE
CE86  D1  C  POP BC
CE87  C1  C  POP HL
CE88  C9  C  RET
C ;
C ; SUBTTL KEY SIGNATURE CONTROL
C MACLIB B:MCP02.MAC
C ;
C ; SETKEY
### MUSIC CONTROL PROGRAM

**KEY SIGNATURE CONTROL**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEAD</td>
<td>28 35</td>
<td>SET KEY SIGNATURE WITHOUT DISPLAYING IT AND MOVE APPROPRIATE UNADJUSTED PITCH DEFINITION TABLE TO THE UNADJUSTED PITCH TABLE (UNATBL).</td>
</tr>
<tr>
<td>CEAF</td>
<td>7A</td>
<td>GIVEN: (B) - KEY LETTER NAME</td>
</tr>
<tr>
<td>CEB8</td>
<td>FE 82</td>
<td>(C) - ACCIDENTAL (-1 TO +1)</td>
</tr>
<tr>
<td>CEBB</td>
<td>3C</td>
<td>(D) - MODE (09H - MAJOR; 08H - MINOR)</td>
</tr>
<tr>
<td>CECC</td>
<td>7A</td>
<td>RETURNS: (A) - COMPLETION CODE</td>
</tr>
<tr>
<td>CECE</td>
<td>3E FF</td>
<td>= 00H IF SUCCESSFUL</td>
</tr>
<tr>
<td>CEDF</td>
<td>77</td>
<td>= FFH IF INVALID DATA OR KEY</td>
</tr>
</tbody>
</table>

- **SETKEY**: PUSH BC
  - PUSH DE
  - PUSH HL
- **CALL ALCHEK**: CHECK DATA FOR VALIDITY
- **CP 0FFH**: CHECK ERROR FLAG
- **JR Z, STK Y82**: JUMP IF INVALID DATA OR KEY
- **LD A, D**: TEST MODE
- **JR NC, STKY02**: JUMP IF UPPER BOUND
- **LD HL, KEYLET**: POINT TO KEY DATA
- **INC HL**: POINT TO ACCIDENTAL
- **INC HL**: STORE ACCIDENTAL
- **INC HL**: STORE MODE
- **CALL KEYNDX**: GET INDEX INTO KEYUPD
- **SLA A**: EACH ENTRY IS TWO BYTES LONG
- **LD HL, KEYUPD**: POINT TO START OF TABLE
- **ADD A, L**: ADD INDEX
- **INC H**: ELSE ADJUST FOR OVERFLOW
- **STKY01**: LD E, (HL): LOAD ADDRESS OF UNADJUSTED PITCH TABLE FOR THE GIVEN KEY
- **INC HL**: TABLE FOR THE GIVEN KEY
- **INC HL**: CHECK FOR VALID KEY
- **OR E**: INVALID IF ZERO
- **JR Z, STKY02**: JUMP IF INVALID
- **LD HL, UNATBL**: DESTINATION
- **LD DE, HL**: SOURCE IN (HL), DESTINATION IN (DE)
- **LD BC, 0008H**: TABLE IS 12 BYTES LONG
- **LDHL, XPOSNS**: POINT TO TRANPOSITION INDICATORS
- **LD (HL), A**: RESET TO NO TRANPOSITION

---

**MACRO-88 3.44 09-Dec-81 PAGE 1-39**
KEY SIGNATURE CONTROL

DISPLAYS THE KEY SIGNATURE OF THE KEY PREVIOUSLY SET VIA THE SETKEY ROUTINE. THIS ROUTINE ASSUMES THAT THE FIELDS KEYLET, KEYACD, AND KEYMOD ARE INITIALIZED TO THE KEY DESIRED. THE CURRENT POSITION IS INCREMENTED TO ACCOUNT FOR THE SPACE OCCUPIED BY THE KEY SIGNATURE.

GIVEN: NONE

RETURNS: (A) - COMPLETION CODE

- 00H IF SUCCESSFUL
- FFH IF INVALID OR NO KEY ESTABLISHED

KEYSIG: PUSH BC
PUSH DE
PUSH HL
LD HL,KEYLET ;POINT TO KEY DATA
LD A,(HL) ;GET LETTER OF KEY
LD B,A ;PUT IT IN (B)
CP 00H ;TEST FOR NO KEY SET
LD A,0FFH ;LOAD NO-KEY-SET FLAG
JR Z,KYSG07 ;EXIT IF NO KEY ESTABLISHED
INC HL ; POINT TO ACCIDENTAL
LD C,(HL) ; GET ACCIDENTAL
INC HL ; POINT TO MODE
LD D,(HL) ; GET MODE
CALL GETKEY ;GET ACCIDENTAL NUMBER AND TYPE
CP 0FFH ;TEST FOR INVALID KEY
JR Z,KYSG07 ;JUMP IF INVALID
LD C,A ;SAVE NUMBER AND TYPE IN (E)
AND 07H ;GET NUMBER ONLY
JR Z,KYSG06 ;JUMP IF NO ACCIDENTALS TO PRINT
LD HL,KEYFLT ;FIRST ASSUME FLAT KEY
LD DE,FLT0CT
BIT FLTSH»,C ;TEST SHARP/FLAT BIT
JR Z,KYSG01 ;JUMP IF FLAT KEY
LD HL,KEYSHP ;ELSE SET UP FOR SHARPS
KEY SIGNATURE CONTROL

CF14 11 C458  C  LD  DE,SHPOCT
CF17 FD 7E 06  C  KYSG01: LD  A, (IY+CLFCOD)  ; USE CLEF CODE AS INDEX
CF1A CB 27  C  SLA  A  ; INTO OCTAVE TABLE FOR
CF1C CB 27  C  SLA  A  ; THE OCTAVES OF THE
CF1E CB 27  C  SLA  A  ; ACCIDENTALS
CF20 83  C  ADD  A,E  ; (DE) POINTS TO THE OCTAVE
CF21 5F  C  LD  E,A  ; TABLE
CF22 30 01  C  JR  NC,KYSG02  ;TEST FOR ADDRESS OVERFLOW
CF24 14  C  INC  D  ; ADJUST FOR OVERFLOW
CF25 FD 7E 05  C  KYSG02: LD  A, (IY+CURPOS)  ; SAVE CURRENT POSITION FOR
CF28 32 0000X  C  LD  (TEMPO1),A  ; PATCH ROUTINE LATER
CF2B C5 C  CALL ADJUST  ; ADJUST PITCH
CF34 CD D507  C  CALL ACONTL  ; DISPLAY ACCIDENTAL
CF37 C1  C  POP  BC  ; RESTORE NUMBER/TYPEx ACCIDENTALS
CF38 CB 69  C  BIT  FLTSHP,C  ; TEST FLAT/SHARP BIT
CF3A 26 03  C  JR  Z,KYSG04  ; JUMP IF FLAT KEY
CF3C FD 35 05  C  DEC  (IY+CURPOS)  ; COMPRESS SHARPS IN SIGNATURE
CF3F 19 EA  C  KYSG04: DJNZ  KYSG03  ; COUNT ACCIDENTALS
CF41 CB 69  C  BIT  FLTSHP,C  ; SEE IF PATCH NEEDED
CF43 28 0A  C  JR  Z,KYSG05  ; NO PATCH NEEDED ON FLAT KEYS
CF45 3A 0000X  C  LD  A,(TEMPO1)  ; GET STARTING POSITION
CF48 47  C  LD  B,A  ; AND SEND TO PATCH ROUTINE
CF4C CD CF57  C  CALL PATCH  ; CLEAN UP THE DISPLAY
CF4F FD 34 05  C  INC  (IY+CURPOS)  ; LEAVE SPACE AFTER KEY SIGNATURE
CF52 AF  C  KYSG05: INC  (IY+CURPOS)  ; JUST FOR GOOD LOOKS
CF53 E1  C  KYSG06: XOR  A  ; SET SUCCESSFUL CODE
CF54 D1  C  KYSG07: POP  HL
CF55 C1  C  POP  DE
CF56 C9  C  POP  BC
CF57 C  ;
CF58 C  ; PATCH
CF59 C  ; PATCH KEY SIGNATURE
CF60 C  ; SHARP KEY SIGNATURES WITH MORE THAN 1 SHARP NEED TO BE PATCHED
CF61 C  ; WITH A SPECIAL CHARACTER. SHARPS ARE DISPLAYED IN A FORMAT
CF62 C  ; THAT IS 2 CHARACTERS WIDE, BUT THE KEY SIGNATURE LOOKS BETTER
CF63 C  ; IF THE SHARPS ARE DISPLAYED ONLY 1 CHARACTER APART. THIS
CF64 C  ; WORKS FINE EXCEPT BETWEEN F# AND C# AND BETWEEN D# AND A#.
CF65 C  ; IN THESE 2 CASES THE SHARPS OVERLAP SO ANOTHER CHARACTER WAS
CF66 C  ; DEFINED TO ACCOMODATE THIS SITUATION. THIS ROUTINE SIMPLY
DISPLAYS THIS SPECIAL CHARACTER IN THE APPROPRIATE PLACE.

GIVEN: (B) - THE COLUMN IN WHICH THE SIGNATURE HAS BEGUN
(C) - THE NUMBER AND TYPE OF ACCIDENTALS IN THE KEY SIGNATURE

RETURNS: NONE

{USE CLEF CODE AS INDEX INTO PATCH VECTOR TABLE
DOUBLE INDEX SINCE EACH ENTRY IS 2 BYTES LONG
TEST FOR ADDRESS OVERFLOW
ADJUST FOR OVERFLOW
GET ADDRESS OF PATCH LEADER
GET CURRENT POSITION AND SAVE IT
GET ADDRESS OF THE STARTING POSITION OF THE KEY SIGNATURE
GET NUMBER OF ACCIDENTALS IN THE KEY SIGNATURE
GET CUT OFF NUMBER TO SEE IF PATCH IS NEEDED
JUMP IF NO MORE PATCHES; ELSE LOAD X-Y COORDINATE
POINT TO NEXT PATCH
SAVE GIVEN DATA
CALCULATE ADDRESS OF X-Y COORDINATE RELATIVE TO THE START OF THE KEY SIGNATURE
PUT X COORDINATE IN LOWER 4 BITS
SAVE BASE ADDRESS
CALCULATE ADDRESS
DISPLAY PATCH CHARACTER
RESTORE BASE ADDRESS
RESTORE GIVEN DATA
CONTINUE TILL ALL DONE
RESET CURRENT POSITION

PATCH: LD A,(IY+CLFCD)
LD HL,PTCHAD
SLA A
ADD A,L
LD L,A
JR NC,PTCH01
INC K
PATC81: LD E,(HL)
INC HL
LD D,(HL)
LD A,(IY+CURPOS)
LD (TEMP01),A
LD (IY+CURPOS),B
LD A,(IY+STFORM)
CALL SACALC
LD A,C
AND 07H
LD C,A
PATC02: LD A,(DE)
DEC A
CP C
JR NC,PATC03
INC DE
LD A,(DE)
INC DE
PUSH BC
LD B,A
SRL B
SRL B
SRL B
SRL B
PUSH HL
CALL XYCORD
LD (HL),ASSPC1
POP HL
POP BC
JR PATC02
PATC03: LD A,(TEMPO1)
LD (IY+CURPOS),A
KEY SIGNATURE CONTROL

KEY INDEX
; GET INDEX INTO KEYBL AND KEYUPD FOR THE KEY GIVEN.
; GIVEN:  (B) - ASCII LETTER NAME
; (C) - ACCIDENTAL (-1 TO +1)
; (D) - MODE (8-MAJOR; 1-MINOR)
; RETURNS:  (A) - TABLE ENTRY NUMBER

CF81  C5  C KEYINDEX:  PUSH  BC
CF82  D5  C  PUSH  DE
CF83  78  C  LD  A,B  ;GET LETTER
CF84  D6  41  C  SUB  41H  ;ADJUST TO 0 - 6
CF86  5F  C  LD  E,A  ;SAVE IT
CF87  47  C  LD  B,A  ;USE IT FOR COUNTER
CF88  3E  01  C  LD  A,01H  ;START AT A-NATURAL
CF8A  28  04  C  JR  Z,KYNX02  ;JUMP IF KEY OF A
CF8C  C6  03  C  KYNX01:  ADD  A,03H  ;POINT TO NEXT WHITE NOTE
CF8E  10  FC  C  DJNZ  KYNX01
CF90  47  C  KYNX02:  LD  B,A  ;SAVE PARTIAL RESULT
KEY SIGNATURE CONTROL

CFC1 79 C LD A,C ;GET ACCIDENTAL
CFC2 A7 C AND A ;TEST FOR NATURAL
CFC3 28 08 C JR Z,KYNX04 ;JUMP IF NATURAL
CFC5 CB 79 C BIT 7,C ;TEST SIGN FOR FLAT/SHARP
CFC7 20 03 C JR NZ,KYNX03 ;JUMP IF FLAT
CFC9 84 C INC B ;ELSE POINT TO SHARP LETTER
CFCA 18 01 C JR KYNX04
CFCB 05 C KYNX03: DEC B ;POINT TO FLAT LETTER
CFCD 7A C KYNX04: LD A,D ;GET MODE
CFCE A7 C AND A ;TEST FOR MAJOR
CFCF 78 C LD A,B ;GET RESULT IF DONE
CFD0 28 17 C JR Z,KYNX07 ;DONE IF MAJOR KEY
CFD2 7B C LD A,E ;GET LETTER
CFD7 30 02 C JR C,KYNX06 ;JUMP IF KEYS OF A OR B
CFD9 28 04 C JR Z,KYNX05 ;JUMP IF KEY OF C
CFDB FE 05 C CP 05H ;ELSE TEST FOR D AND E
CFD9 30 02 C JR C,KYNX06 ;JUMP IF KEYS OF D OR E
CFD9 1E 05 C KYNX05: LD E,05H ;ELSE F AND G ADD 5
CFE1 78 C KYNX06: LD A,B ;GET PARTIAL RESULT
CFE2 83 C ADD A,E ;ADD RELATIVE MAJOR OFFSET
CFE3 FE 15 C CP 15H ;TEST FOR WRAP-AROUND
CFE5 30 02 C JR C,KYNX07 ;JUMP IF NONE
CFE7 06 15 C SUB 15H ;ELSE ADJUST FOR WRAP-AROUND
CFE9 D1 C KYNX07: POP DE
CFEC C9 C RET
C ; RESTKEY
C ; RESET KEY TO NO KEY AND TRANSPOSITION INDICATOR TO NONE.
C ; GIVEN: NONE
C ; RETURNS: NONE

CFE1 C5 C RSTKEY: PUSH BC
CFE2 D5 C PUSH DE
CFE3 E5 C PUSH HL
CFE5 21 0000 C LD HL,KEYLET ;POINT TO KEY DATA
CFE6 AF C XOR A ;ZERO ACCUMULATOR
CFE7 06 05 C LD B,05H ;NUMBER OF FIELDS TO RESET
CFE8 77 C RSKY01: LD (HL),A ;ZERO KEY FIELD
CFE9 23 C INC HL ;POINT TO NEXT KEY FIELD
CFEA 10 FC C DJNZ RSKY01
CFEB 21 0000 C LD HL,XPOSYS ;POINT TO TRANSPOSITION CODES
CFEC 06 03 C LD B,03H ;NUMBER OF TRANSPOSE FIELDS
KEY SIGNATURE CONTROL

CFFE 77 C RSKY02: LD (HL),A ;ZERO TRANSPOSE FIELD
CFFF 23 C INC HL ;POINT TO NEXT TRANSPOSE FIELD
D000 10 FC C DJNZ RSKY02
D002 21 C406 C LD HL,NOKEY ;POINT TO UNADJUSTED PITCH SOURCE
D005 11 0000x C LD DE,UNATBL ;POINT TO DESTINATION TABLE
D008 01 0000C C LD BC,000CH ;LENGTH OF TABLE
D00B ED 80 C LDIR
D00D E1 C POP HL
D00E D1 C POP DE
D00F C1 C POP BC
D018 C9 C RET

; SETXPO
;
; SET TRANSPOSITION INDICATORS
; CALCULATES THE DISPLACEMENT AND DIRECTION FOR TRANSPOSING
; GIVEN LETTERS/SCALE DEGREES TO THE GIVEN KEY FROM THE
; CURRENT KEY. THE HALF-STEP DISPLACEMENT FIELD (XPOSLS)
; IS EQUAL TO 00H FOR NO TRANSPOSITION OR IS EQUAL TO THE
; NUMBER OF HALF STEPS THE NEW KEY IS FROM THE OLD KEY.
; THE UNADJUSTED PITCH TABLE FOR THE NEW KEY IS COPIED
; INTO THE UNADJUSTED PITCH TABLE. NOTE THAT NO MODE IS
; GIVEN SINCE THE MODE IS ASSUMED TO BE THE SAME AS THE
; CURRENT KEY. THE DIRECTION INDICATOR IDENTIFIES THE
; DIRECTION OF TRANSPOSITION BY ITS SIGN; POSITIVE VALUES
; TRANSPOSE UPWARD, NEGATIVE VALUES TRANSPOSE DOWNWARD.
; THE VALUE OF THE DIRECTION INDICATES THE OCTAVE TO WHICH
; THE TRANSPOSITION IS DONE. A VALUE OF 1 INDICATES THAT
; THE TRANSPOSITION WILL BE KEPT TO LESS THAN 1 OCTAVE AWAY
; FROM THE ORIGINAL KEY. A VALUE OF 2 INDICATES THAT THE
; TRANSPOSITION WILL BE LESS THAN 2 OCTAVES, BUT MORE THAN,
; OR EQUAL TO, 1 OCTAVE AWAY, ETC.
;
; GIVEN: (B) - LETTER OF KEY TO TRANPOSE TO
; (C) - ACCIDENTAL OF KEY TO TRANPOSE TO
; (D) - DIRECTION/OCTAVE
; RETURNS: (A) - COMPLETION CODE
; = 00H IF SUCCESSFUL
; = FFH IF NO CURRENT KEY ESTABLISHED OR GIVEN
; KEY IS INVALID
;
; D011 C5 C SETXPO: PUSH BC
D012 D5 C PUSH DE
D013 E5 C PUSH HL
D014 21 0000x C LD HL,KEYLET ;POINT TO CURRENT KEY
D017 7E C LD A, (HL) ;GET KEY LETTER
D018 A7 C AND A ;TEST FOR NO KEY
KEY SIGNATURE CONTROL

M019 28 3E C JR Z,STXP04 ;EXIT IF NO KEY IS SET
M01B C5 C PUSH BC ;SAVE GIVEN LETTER/ACCIDENTAL
M01C D5 C PUSH DE ;SAVE GIVEN OCTAVE/DIRECTION
M01D 16 00 C LD D,00H ;SET AT ZERO OCTAVE
M01F 5F C LD E,A ;SET CURRENT LETTER
M020 23 C INC HL ;POINT TO CURRENT ACCIDENTAL
M021 66 C LD H,(HL) ;GET CURRENT LETTER
M022 2E 00 C LD L,00H ;SET AT ZERO OCTAVE
M024 CD DD05 C CALL HLFSFP ;NUMBER OF HALF STEPS BETWEEN
M027 D1 C POP DE ;RESTORE OCTAVE/DIRECTION
M028 FE FF C CP 0FFH ;TEST FOR INVALID DATA
M02A 20 03 C JR NZ,STXP01 ;JUMP IF DATA IS VALID
M02C 8F C POP BC ;ELSE RESTORE STACK POINTER
M02D 18 2A C JR STXP84 ! AS® EXIT
M02F 78 C STXP01: LD A,B ;GET NUMBER OF HALF STEPS
M030 CB 7A C BIT 7,D ;TEST FOR UP/DOWN
M032 20 0E C JR NZ,STXP02 ;JUMP IF DOWN
M034 CB 7F C BIT 7,A ;TEST FOR ABOVE/BELLO
M036 28 8D C JR Z,STXP03 ;JUMP IF TO-LETTER > FROM
M038 C6 8C C ADD A,0CH ;ELSE ADJUST NUMBER HALF STEPS
M03A 18 09 C JR STXP03 ;WHEN OCTAVE BOUNDARY CROSSED
M03C CB 7F C STXP02: BIT 7,A ;TEST FOR ABOVE/BELLO
M03E 20 05 C JR NZ,STXP03 ;JUMP IF TO-LETTER < FROM
M040 A7 C AND A ;TEST FOR ZERO
M041 28 02 C JR Z,STXP03 ;JUMP IF EXACT OCTAVE
M043 C6 F4 C ADD A,0F4H ;ELSE ADJUST NUMBER HALF STEPS
M045 6F C STXP03: LD L,A ;SAVE NUMBER OF HALF STEPS
M046 62 C LD H,D ;SAVE DIRECTION/OCTAVE
M047 C1 C POP BC ;RESTORE GIVEN LETTER/ACCIDENTAL
M048 CD DD0A C CALL KEYSNP ;MOVE CURRENT KEY TO OLD KEY
M04B 3A 0000X C LD A,(KEYMOD) ;SET MODE OF CURRENT KEY
M04E 57 C LD D,A ;SEND AS NEW MODE
M04F CD CEA5 C CALL SETKEY ;SET KEY TO GIVEN KEY
M052 FE FF C CP 0FFH ;CHECK FOR INVALID DATA
M054 20 07 C JR NZ,STXP05 ;JUMP IF VALID
M056 CD DD0A C CALL KEYSNP ;ELSE RESET CURRENT KEY
M059 3E FF C STXP04: LD A,OFFH ;SET ERROR FLAG
M05B 18 4F C JR STXP15 ; AND EXIT
M05D 7C C STXP05: LD A,H ;GET DIRECTION/OCTAVE
M05E 32 0000X C LD (XPOS0D),A ;STORE IT
M061 0E 8C C LD C,0CH ;NUMBER OF HALF STEPS/OCTAVE
M063 CB 7C C BIT 7,H ;TEST UP/DOWN
M065 28 84 C JR Z,STXP06 ;JUMP IF UP
M067 ED 44 C NEG ;ELSE MAKE OCTAVE POSITIVE
M069 0E F4 C LD C,0F4H ;12 HALF STEPS DOWN PER OCTAVE
M06B 47 C STXP06: LD B,A ;COUNT OCTAVES IN (8)
M06C 7D C LD A,L ;GET NUMBER OF HALF STEPS
M06D 05 C DEC B ;TEST FOR LESS THAN 1 OCTAVE


KEY SIGNATURE CONTROL

D06E 28 03 C JR Z,STXP08 ;JUMP IF DONE
D070 81 C STXP07: ADD A,C ;ADD +/-12 FOR EACH OCTAVE
D071 10 FD C DJNZ STXP07
D073 32 0000X C STXP08: LD (XPOSHS),A ;STORE NUMBER OF HALF STEPS
D076 3A 0000X C LD A,(KEYLET) ;GET NEW KEY LETTER
D079 5F C LD E,A
D07A 3A 0000X C LD A,(OLDKYL) ;GET OLD KEY LETTER
D07D 93 C SUB E ;LETTER NAME DIFFERENCE
D07E 38 02 C JR NC,STXP09 ;JUMP IF RESULT IS POSITIVE
D080 C6 07 C ADD A,07H ;ELSE MAKE IT POSITIVE
D082 CB 7C C STXP09: BIT 7,H ;TEST FOR DOWN/UP TRANSPOSE
D084 28 0B C JR Z,STXP10 ;JUMP IF TRANSPOSE IS UP
D086 A7 C AND A ;TEST FOR SAME LETTERS
D087 28 16 C JR N2,STXP14 ;JUMP IF NOT SAME
D089 7D C LD A,L ;GET HALF STEPS IN 1 OCTAVE
D08A 30 C DEC A ;PUT IN RANGE FD TO FF
D08B FE FD C CP 0FDH ;UNISON, 2ND OR 7TH INTERVAL?
D08D 38 0A C JR NC,STXP11 ;JUMP IF UNISON OR 2ND
D08F 18 0B C JR STXP12 ;JUMP IF 7TH
D091 A7 C STXP10: AND A ;TEST FOR SAME LETTERS
D092 28 0A C JR N2,STXP13 ;JUMP IF NOT SAME
D094 7D C LD A,L ;GET HALF STEPS IN 1 OCTAVE
D095 FE 03 C CP 03H ;UNISON, 2ND OR 7TH?
D097 38 03 C JR C,STXP12 ;JUMP IF UNISON OR 2ND
D099 AF C STXP11: XOR A ;ZERO FOR MIN/MAX CUTOFF LETTER
D09A 18 03 C JR STXP14
D09C 3E 07 C STXP12: LD A,07H ;8 FOR MIN/MAX CUTOFF LETTER
D09E 3C C STXP13: INC A ;ADJUST FOR UPWARD TRANSPOSE
D09F 1E 40 C STXP14: LD E,40H ;'A' MINUS 1
D0A1 03 C ADD A,E ;CALCULATE CUT OFF LETTER
D0A2 32 0000X C LD (XPOSCH),A ;STORE IT
D0A5 AF C XOR A ;SET SUCCESSFUL COMPLETION CODE
D0A6 E1 C STXP15: POP HL
D0A7 D1 C POP DE
D0A8 C1 C POP BC
D0A9 C9 C RET

; KEYSHIP
; KEY SNAP
; EXCHANGES THE CURRENT KEY DATA WITH THE OLD KEY DATA.
; GIVEN:  NONE
; RETURNS: NONE

D0AA D5 C KEYSHP: PUSH DE
D0AB E5 C PUSH HL
MACRO-80 3.44  09-Dec-81  PAGE 1-48

MUSIC CONTROL PROGRAM

MACLIB  B:MACP03.MAC

KEY SIGNATURE CONTROL

D0AC  2A 0000 C  LD  HL,(KEYLET) ;GET CURRENT KEY LETTER AND
D0AF  EB C  EX  DE,HL ;ACCIDENTAL AND SAVE IT
D0B0  2A 0000 C  LD  HL,(OLDKYL) ;PUT OLD KEY LETTER AND
D0B3  22 0000 C  LD  (KEYLET),HL ;ACCIDENTAL IN NEW POSITION
D0B6  EB C  EX  DE,HL ;PUT NEW KEY DATA IN THE OLD
D0B7  22 0000 C  LD  (OLDKYL),HL ;POSITION
D0BA  E1 C  POP  HL
D0BB  D1 C  POP  DE
D0BC  C9 C  RET

;TIMSIG

;DISPLAY TIME SIGNATURE.

D0BD  FD 7E 05 C  TIMSIG: LD  A,(IY+CURPOS)
D0C0  C6 02 C  ADD  A,02H
D0C2  FD 77 05 C  LD  (IY+CURPOS),A
D0C5  C9 C  RET

;SUBTL MELODY DISPLAY AND MANIPULATION

;NEWTON

;NEWTUN

;NEWTUNE

;TRANSLATES EITHER A SCALE DEGREE/MODIFIER OR LETTER NAME/
;ACCIDENTAL FORMAT TUNE INTO THE UNADJUSTED (TUNE TABLE)
;FORMAT AND DISPLAYS THE TUNE USING THE PARTITIONS GIVEN.

;GIVEN: (BC) - ADDRESS OF PARTITION LIST (ENDING WITH FFH)
; (DE) - ADDRESS OF TUNE HEADER
;RETURNS: (A) - COMPLETION CODE

;EOT - END OF TUNE FOUND
;EOP - END OF PARTITION LIST FOUND BEFORE END OF TUNE
;FDH - INVALID TUNE TYPE
;FEH - INVALID KEY IN HEADER
;FFH - INVALID NOTE IN TUNE

D0C6  C5 C  NEWTUN: PUSH  BC
D0C7  D5 C  PUSH  DE
D0C8  E5 C  PUSH  HL
D0C9  AF C  XOR  A ;RESET INHIBIT STORE FIELD
D0CA  32 0000 C  LD  (INHISTO),A
D0CD  DD 21 FFFC C  LD  IX,TUNTBL-TTELEN ;POINT TO START OF TUNE TABLE
MELODY DISPLAY AND MANIPULATION

D0D1 C5 C PUSH BC ;SAVE PARTITION LIST ADDRESS
D0D2 EB C EX DE,HL ;PUT TUNE ADDRESS IN (HL)
D0D3 58 C LD D,B ; AND PARTITION ADDRESS
D0D4 59 C LD E,C ; IN (DE)
D0D5 23 C INC HL ;SKIP OVER IDENTIFIER
D0D6 7E C LD A,(HL) ;GET TUNE TYPE
D0D7 32 0000% C LD (TEMP02),A ;SAVE TUNE TYPE
D0D8 23 C INC HL ;POINT TO CLEF
D0D9 4E C LD C,(HL) ;GET CLEF
D0DA 23 C INC HL ;POINT TO KEY DATA
D0DB E5 C PUSH HL ;SAVE TUNE ADDRESS
D0DE 1A C PAGE 1-49: LD A,(DE) ;GET PARTITION NUMBER
D0DF 13 C INC DE ;POINT TO NEXT PARTITION NUMBER
D0E0 FE FF C CP 0FFH ;TEST FOR END OF LIST
D0E2 28 8D C JR Z,NNTN02 ;JUMP IF END OF LIST
D0E4 47 C LD B,A ;SEND PARTITION NUMBER IN (B)
D0E5 CD CCBE C CALL ACTPAR ;ACTIVATE THE PARTITION
D0E8 FE FF C CP 0FFH ;TEST FOR ERROR
D0EA 28 F2 C JR Z,NNTN01 ;IGNORE ERRORS
D0EC FD 71 06 C LD (IY+CLFCOD),C ;STORE CLEF IN PARTITION ENTRY
D0EF 1B ED C JR NNTN01 ;GO GET NEXT PARTITION
D0F1 E1 C PAGE 2-49: POP HL ;RESTORE TUNE ADDRESS
D0F2 46 C LD B,(HL) ;GET KEY LETTER NAME
D0F3 23 C INC HL ;POINT TO ACCIDENTAL
D0F4 4E C LD C,(HL) ;GET ACCIDENTAL
D0F5 23 C INC HL ;POINT TO MODE
D0F6 56 C LD D,(HL) ;GET MODE
D0F7 CD CEA5 C CALL SETKEY ;ESTABLISH KEY SIGNATURE
D0FA A7 C C CP A ;TEST FOR ERROR
D0FB 28 84 C JR Z,NNTN03 ;JUMP IF NO ERROR
D0FD 3E FE C LD A,0FEH ;ELSE LOAD INVALID KEY FLAG
D0FF 18 27 C JR NNTN06 ; AND EXIT
D101 3E 18 C NNTN03: LD A,HDRLN ;SKIP OVER THE REST OF THE
D103 D6 05 C SUB 05H ; HEADER INFORMATION
D105 85 C ADD A,L
D106 6F C LD L,A
D107 30 01 C JR NC,NNTN04 ;TEST FOR OVERFLOW
D109 24 C INC H ;ADJUST FOR OVERFLOW
D10A EB C NNTN04: EX DE,HL ;START OF TUNE IN (DE)
D10B 3A 0000% C LD A,(TEMP02) ;GET TUNE TYPE
D10E 47 C LD B,A ;SEND IT IN (B)
D10F FE A0 C CP TYPSC; SCALE DEGREE/MODIFIER FORMAT?
D111 28 88 C JR Z,NNTN05 ;JUMP IF SO
D113 FE 80 C CP TUPLET ;LETTER NAME/ACCIDENTAL FORMAT?
D115 28 84 C JR Z,NNTN05 ;JUMP IF SO
D117 3E FD C LD A,0FDH ;ELSE INVALID TUNE TYPE
D119 18 8D C JR NNTN06 ;EXIT
D11B CD D12D C NNTN05: CALL XLATUN ;TRANSLATE TUNE TO TUNE TABLE FORMAT
MELODY DISPLAY AND MANIPULATION

D11E  FE FF  C  CP  0FFH  ;TEST FOR ERROR IN TRANSLATION
D120  28 06  C  JR  Z,NNTN6 ;EXIT IF ERROR FOUND
D122  D1  C  POP  DE  ;RESTORE PARTITION LIST ADDRESS
D123  CD D17A  C  CALL  PRTRCN  ;DISPLAY TUNE
D126  18 01  C  JR  NNTN7
D128  C1  C  NNTN6:  POP  BC  ;RESET STACK POINTER
D129  E1  C  NNTN7:  POP  HL
D12A  D1  C  POP  DE
D12B  C1  C  POP  BC
D12C  C9  C  RET

D12D  C5  C  XLATUN:  PUSH  BC
D12E  D5  C  PUSH  DE
D12F  E5  C  PUSH  HL
D138  DD 21  FFFCX  C  LD  IX,TUNTBL-TTEL1EN  ;POINT TO START OF TUNE TABLE
D134  EB  C  EX  DE,HL  ;PUT TUNE ADDRESS IN (HL)
D135  78  C  LD  A,B  ;GET TUNE TYPE
D136  32 0000X  C  LD  (TEMP02),A  ;STORE IT
D139  7E  C  XLTN01:  LD  A,(HL)  ;GET SCALE DEGREE/LETTER NAME
D13A  FE FF  C  CP  TTDBAR  ;TEST FOR DOUBLE BAR LINE
D13C  28 27  C  JR  Z,XLTN03  ;JUMP IF SO
D13E  FE FE  C  CP  TTSBARR  ;TEST FOR SINGLE BAR LINE
D140  28 23  C  JR  Z,XLTN03  ;JUMP IF SO
D142  47  C  LD  B,A  ;ELSE ASSUME NOTE
D143  23  C  INC  HL  ;POINT TO MODIFIER/ACCIDENTAL
D144  4E  C  LD  C,(HL)  ;GET MODIFIER/ACCIDENTAL
D145  23  C  INC  HL  ;POINT TO OCTAVE
D146  3A 0000X  C  LD  A,(TEMP02)  ;GET TUNE TYPE
D149  FE A8  C  CP  TYPSCM  ;SCALE DEGREE TYPE?
D14B  28 05  C  JR  H2,XLTN02  ;JUMP IF LETTER NAME TYPE
D14D  CD D861  C  CALL  CVTSL  ;CONVERT SCALE DEGREE TO LETTER NAME
D150  42  C  LD  B,D  ;GET LETTER NAME AND ACCIDENTAL
D151  48  C  LD  C,E
D152  CD D84D  C  XLTN02:  CALL  CVTTLT  ;CONVERT LETTER TO UNADJUSTED

XLATUN:  PUSHBC
PUSHDE
PUSHHL
LDIX,TUNTBL-TTEL1EN;POINT TO START OF TUNE TABLE
EXDE,HL;PUT TUNE ADDRESS IN (HL)
LDAB;GET TUNE TYPE
LD(TEMP02),A;STORE IT
XLTN01:LDAA,(HL);GET SCALE DEGREE/LETTER NAME
CPTTDBAR;TEST FOR DOUBLE BAR LINE
JRZ,XLTN03;JUMP IF SO
CP TTSBARR;TEST FOR SINGLE BAR LINE
JRZ,XLTN03;JUMP IF SO
LB,A;ELSE ASSUME NOTE
INC HL;POINT TO MODIFIER/ACCIDENTAL
LDC,(HL);GET MODIFIER/ACCIDENTAL
INC HL;POINT TO OCTAVE
LDAA,(TEMP02);GET TUNE TYPE
CP TYPSCM;SCALE DEGREE TYPE?
JR H2,XLTN02;JUMP IF LETTER NAME TYPE
CALL CVTSL;CONVERT SCALE DEGREE TO LETTER NAME
LDB,D;GET LETTER NAME AND ACCIDENTAL
LDC,E
CALL CVTTLT;CONVERT LETTER TO UNADJUSTED
MELODY DISPLAY AND MANIPULATION

D155 FE FF C CP 0FFH ;TEST FOR ERROR
D157 28 ID C JR Z,XLTN04 ;EXIT IF INVALID DATA
D159 47 C LD B,A ;ELSE SAVE UNADJUSTED PITCH
D15A 4E C LD C, (HL) ;GET OCTAVE
D15B 23 C INC HL ;POINT TO RHYTHM
D15C 56 C LD D, (HL) ;GET RHYTHM
D15D 23 C INC 11 ;POINT TO CPI
D15E 5E C LD E, (HL) ;GET CPI
D15F 23 C INC HL ;POINT TO NEXT NOTE
D160 CD 052A C CALL STORE ;STORE NOTE IN TUNE TABLE
D163 18 D4 C JR XLTN01 ;GO TRANSLATE NEXT NOTE
D165 47 C XLTN03: LD B,A ;PUT SINGLE/DENT BAR
D166 4F CD C, A ;LINE IN ALL 4 REGISTERS
D167 57 C LD D,A ;TO STORE IN TUNE TABLE
D168 5F C LD E,A
D169 CD 052A C CALL STORE ;STORE BAR LINE
D16C 11 0005 C LD DE,05H ;LENGTH OF NOTE ENTRY
D16F 19 C ADD HL,DE ;POINT TO NEXT NOTE
D170 78 C LD A,B ;GET TBE BAR LINE
D171 FE FE C CP TSBAR ;TEST FOR SINGLE/DENT BAR
D173 2B C4 C JR Z,XLTN81 ;CONTINUE IF SINGLE BAR
D175 AF C XOR A ;ELSE SET SUCCESSFUL FLAG
D176 E1 C XLTN04: POP HL
D177 D1 C POP DE
D178 C1 C POP BC
D179 C9 C RET

;PRINT TUNE UNTIL END OF SCREEN
;DISPLAYS THE TUNE IN THE TUNE TABLE USING ALL OF THE PARTITIONS
;GIVEN IN THE ORDER THEY ARE GIVEN. STARTING WITH THE FIRST
;PARTITION GIVEN, THE TUNE IS DISPLAYED UNTIL THE END OF THAT
;PARTITION IS REACHED. THE NEXT PARTITION, IF ANY, IN THE GIVEN
;LIST IS THEN ACTIVATED AND THE TUNE IS CONTINUED IN THAT
;PARTITION. THIS CONTINUES UNTIL EITHER THE ENTIRE TUNE IS
;DISPLAYED OR NO MORE PARTITIONS ARE AVAILABLE IN WHICH TO
;CONTINUE.

;GIVEN: (DE) - ADDRESS OF PARTITION NUMBER LIST
;RETURNS: (A) - COMPLETION CODE
; = EOT - END OF TUNE
; (DE) - NEXT PARTITION TO BE USED
; (IX) - DOUBLE BAR LINE
; = EOP - END OF PARTITIONS
; (DE) - END OF PARTITION LIST
; (IX) - NEXT NOTE TO BE DISPLAYED
MUSIC CONTROL PROGRAM MACRO-88 3.44 09-Dec-81 PAGE 1-52

MELODY DISPLAY AND MANIPULATION

D17A C5 C PRTSCR: PUSH BC
D17B ES C PUSH HL
D17C DD 21 00000X C LD IX,TUNBL ; INITIALIZE TUNE TABLE ADDRESS
D17E 1A C PRSC01: LD A,(DE) ; GET PARTITION NUMBER
D181 13 C INC DE ; POINT TO NEXT PARTITION NUMBER
D182 FE FF C CP 8FFH ; TEST FOR END OF PARTITION LIST
D184 28 04 C JR NZ,PRSC02 ; JUMP IF NOT END OF LIST
D186 3E 19 C LD A,EOP ; ELSE LOAD END OF PARTITION LIST
D188 18 1E C JR PRSC03 ; FLAG AND EXIT
D18A 47 C PRSC02: LD B,A ; SEND PARTITION NUMBER IN (B)
D18C CD C8E C CALL ACTPARN ; ACTIVATE THE PARTITION
D18E FE FF C CP 8FFH ; TEST FOR ERROR
D190 28 16 C JR NZ,PRSC03 ; EXIT IF ERROR FOUND
D192 FD 48 04 C LD B,(IY+STFORN) ; GET STAFF ORIENTATION
D195 FD 48 06 C LD C,(IY+CLFCOD) ; GET CLEF CODE
D199 CD C0SE C CALL STFCLF ; DISPLAY STAFF AND CLEF
D19B CD C0E8 C CALL KEYSIG ; PRINT KEY SIGNATURE
D19D CD D08D C CALL TIMSIG ; PRINT TIME SIGNATURE
D1A1 CD D1AB C CALL PRTSTF ; DISPLAY TUNE UNTIL END OF STAFF
D1A4 FE 18 C CP EOS ; TEST FOR END OF STAFF
D1A6 28 D8 C JR NZ,PRSC01 ; CONTINUE IN NEXT PARTITION
D1A8 C1 C PRSC03: POP HL
D1A9 C1 C POP BC
D1AA C9 C RET

; PRTSTF
; PRINT TUNE UNTIL END OF STAFF
; DISPLAYS A STORED TUNE IN THE ACTIVE PARTITION UNTIL THE
; END OF THE STAFF IS REACHED OR THE END OF THE TUNE IS FOUND.

; GIVEN: (IX) - ADDRESS OF TUNE IN TUNE TABLE FORMAT
; RETURNS: (A) - COMPLETION CODE
; = EOT - END OF TUNE;
; = EOS - END OF STAFF;
; = FFH - ERROR;
; = DOUBLE BAR LINE ENTRY
; = NEXT NOTE TO DISPLAY
; = NOTE IN ERROR

D1AB C5 C PRTSTF: PUSH BC
D1AC D5 C PUSH DE
D1AD E5 C PUSH HL
MELODY DISPLAY AND MANIPULATION

D1AE CD D1E2 C PRST01: CALL PRTBAR ; PRINT ONE MEASURE
D1B1 FE 17 C CP EOM ; TEST FOR END OF MEASURE
D1B3 28 8D C JR Z, PRST02 ; JUMP IF SO
D1B5 FE 1A C CP EOT ; TEST FOR END OF TUNE
D1B7 20 25 C JR H2, PRST04 ; EXIT IF ANY OTHER RETURN CODE
D1B9 06 00 C LD B, 00H ; ELSE CLOSE THE STAFF WITH
D1BB CD D1FA C CALL CLOSTF ; A DOUBLE BAR AND RETURN
D1BE 3E 1A C LD A, EOT ; WITH END OF TUNE FLAG
D1C8 18 1C C JR PRST04
D1C2 11 0004 C PRST02: LD DE, TTELEN ; TUNE TABLE ENTRY LENGTH
D1C5 DD 19 C ADD IX, DE ; POINT TO NEXT ENTRY
D1C7 FD 46 05 C LD B, (IY+CURPOS) ; TEST FOR ROOM FOR 1 MORE
D1CA FD 7E 02 C LD A, (IY+PARLEN) ; MEASURE
D1C0 90 H ; NEXT NOTE TO BE DISPLAYED
D1CE FE 09 C CP MIMEAR
D1D0 38 05 C JR C, PRST03 ; JUMP IF NOT ENOUGH ROOM
D1D2 CD D1E5 C CALL BARLIN ; ELSE PRINT BAR LINE AND CONTINUE
D1D5 1B 17 C JR PRST01
D1D7 06 FF C PRST03: LD B, 0FFH ; CLOSE STAFF WITH SINGLE BAR
D1D9 CD D1FA C CALL CLOSTF
D1DC 3E 18 C LD A, EOS ; RETURN WITH END OF STAFF FLAG
D1DE E1 C PRST04: POP HL
D1DF D1 C POP DE
D1E8 C1 C POP BC
D1E1 C9 C RET

; PRINT TUNE UNTIL END OF BAR
; DISPLAYS A STORED TUNE UNTIL A MEASURE IS COMPLETED OR THE END
; OF THE STAFF OR TUNE IS REACHED.
; GIVEN: (IX) - ADDRESS OF TUNE IN TUNE TABLE FORMAT
; RETURNS: (A) - COMPLETION CODE
; = EOM - END OF MEASURE;
; = EOT - END OF TUNE
; = EOS - END OF STAFF REACHED BEFORE END OF MEASURE;
; = FFH - ERROR;
; = IX - NEXT NOTE TO BE DISPLAYED
; = IX - NOTE IN ERROR

D1E2 C5 C PRTBAR: PUSH BC
D1E3 D5 C PUSH DE
D1E4 E5 C PUSH HL
D1E5 3E FF C LD A, 0FFH ; INHIBIT STORE FUNCTIONS
MUSIC CONTROL PROGRAM  MACRO-80 3.44  09-Dec-81  PAGE 1-54

MELODY DISPLAY AND MANIPULATION

```
D1E7  32 0000H  C   LD  (INHSTO),A
D1EA  DD 7E 80  C   PRBR01:  LD  A,(IX+UNAPIT) ;GET UNADJUSTED PITCH
D1ED  FE FE  C   CP  TTDBAR ;TEST FOR SINGLE BAR LINE
D1EF  28 34  C   JR  Z,PRBR05 ;JUMP IF END OF MEASURE
D1F1  FE FF  C   CP  TTDBAR ;TEST FOR DOUBLE BAR LINE
D1F3  28 34  C   JR  Z,PRBR06 ;EXIT IF END OF TUNE
D1F5  47  C   LD  B,A ;ELSE PUT PITCH IN (B)
D1F6  DD 4E 01  C   LD  C,(IX+OCTAVE) ;GET OCTAVE
D1F9  DD 56 82  C   LD  D,(IX+RHYBYT) ;GET RHYTHM
D1FC  DD 5E 03  C   LD  E,(IX+CPINCR) ;GET CURRENT POSITION INCREMENT
D1FF  CD 032A  C   CALL  PITCHU ;DISPLAY THE PITCH
D202  FE 00  C   CP  00H ;TEST FOR NO ERROR
D204  28 04  C   JR  Z,PRBR82 ;CONTINUE IF NO ERROR
D296  FE 16  C   CP  GORANS ;TEST FOR SINGLE BAR LINE
D208  DD 11 0004  C   PRBR82:  LDDE,HELEN ;TUNE TABLE ENTRY LENGTH
D20D  DD 19  C   ADD  IX,DE ;POINT TO NEXT NOTE
D20F  18 09  C   JR  PRBR01 ;GO GET NEXT NOTE
D211  FE 18  C   PRBR03:  CP  EOS ;TEST FOR END OF STAFF
D213  DD 28 16  C   JR  NZ,PRBR02 ;EXIT IF NOT END OF STAFF
D215  DD 7E FC  C   LD  A,(IX+TELTC) ;GET PREVIOUS PITCH OR BAR LINE FLAG
D218  FE FE  C   CP  TTDBAR ;TEST FOR SINGLE BAR LINE
D21A  28 04  C   JR  Z,PRBR04 ;EXIT IF END OF MEASURE
D21C  3E 18  C   LD  A,EOS ;END OF STAFF REACHED AND MEASURE
D21E  18 08  C   JR  PRBR07 ;IS NOT COMPLETE
D220  DD 11 FFFC  C   PRBR04:  LDDE,TELTC ;MINUS TUNE TABLE ENTRY LENGTH
D223  DD 19  C   ADD  IX,DE ;POINT TO PREVIOUS BAR LINE
D225  3E 17  C   PRBR05:  LD  A,EDM ;ELSE LOAD COMPLETED MEASURE FLAG
D227  18 02  C   JR  PRBR07 ;AND EXIT
D229  3E 1A  C   PRBR06:  LD  A,ETN ;LOAD END OF TUNE FLAG AND EXIT
D22B  E1  C   PRBR07:  POP  HL
D22C  D1  C   POP  DE
D22D  C1  C   POP  BC
D22E  C9  C   RET

C
C ; HILO
C
C ; HIGHEST/LOWEST NOTE
C ; SCANS A TUNE AND FINDS THE HIGHEST AND LOWEST PITCHES USED.
C ; THE PITCHES ARE RETURNED IN THE INTEGER FORMAT. INVALID
C ; PITCHES ARE IGNORED.
C
C ; GIVEN: (B) - TYPE TUNE
C ; (DE) - ADDRESS OF FIRST NOTE OF TUNE
C ; RETURNS: (B) - HIGHEST PITCH IN INTEGER FORMAT
C ; (C) - LOWEST PITCH IN INTEGER FORMAT
C
```
MELODY DISPLAY AND MANIPULATION

```
D22F  D5  C  HILO:  PUSH  DE
D230  E5  C  PUSH  HL
D231  7B  C  LD  A,B   ;GET TUNE TYPE
D232  32  0000X  C  LD  (TEMP02),A   ;STORE IT
D235  EB  C  EX  DE,HL   ;PUT TUNE ADDRESS IN (HL)
D236  11  00FF  C  LD  DE,00FFH  ;INITIALIZE HIGH/LOW PITCHES
D239  7E  C  HILO01:  LD  A,(HL)  ;GET FIRST NOTE
D23A  FE  FC  C  CP  0FCH   ;TEST FOR PITCHES
D23C  38  0D  C  JR  C,HILO02  ;JUMP IF PITCH
D23E  FE  FF  C  CP  TTDBAR  ;ELSE TEST FOR DOUBLE BAR
D240  28  38  C  JR  2,HILO06  ;EXIT ON DOUBLE BAR
D242  7D  C  LD  A,L   ;ELSE SKIP THIS ENTRY
D243  C6  05  C  ADD  A,85H  ;ADD ENTRY LENGTH
D245  4F  C  LD  L,A
D246  38  F1  C  JR  NC,HILO01  ;TEST FOR OVERFLOW
D248  24  C  INC  H   ;ADJUST FOR OVERFLOW
D249  18  EE  C  JR  HILO01  ; AND CONTINUE
D24B  47  C  HILO02:  LD  B,A  ;PUT NOTE IN (B)
D24C  23  C  INC  HL   ;POINT TO MODIFIER/ACCIDENTAL
D24D  4E  C  LD  C,(HL)  ;GET IT
D24E  23  C  INC  HL   ;POINT TO OCTAVE
D24F  D5  C  PUSH  DE   ;SAVE HIGHEST/LOWEST NOTES
D250  3A  0000X  C  LD  A,(TEMP02)  ;GET TUNE TYPE
D253  FE  A0  C  CP  TYPSCL   ;TEST FOR SCALE DEGREE TYPE
D255  28  05  C  JR  NC,HILO01  ;IF NOT, ASSUME LETTER NAME
D257  CD  D001  C  CALL  CVTSTL  ;CONVERT SCALE DEGREE TO LETTER
D25A  42  C  LD  B,D  ;PUT LETTER NAME AND ACCIDENTAL
D25B  4B  C  LD  C,E  ; IN SAME REGISTERS
D25C  56  C  HILO03:  LD  D,(HL)  ;NOW GET OCTAVE
D25D  23  C  INC  HL   ;POINT TO NEXT BYTE
D25E  CD  D04C  C  CALL  CVTIELA  ;CONVERT LETTER TO ABSOLUTE
D261  D1  C  POP  DE   ;RESTORE HIGHEST/LOWEST PITCH
D262  FE  FF  C  CP  0FFH   ;TEST FOR ERROR
D264  28  10  C  JR  2,HILO05  ;IGNORE NOTE IF IN ERROR
D266  47  C  LD  B,A  ;SEND ABSOLUTE PITCH IN (B)
D267  CD  D030  C  CALL  CVTIATA  ;ELSE CONVERT TO INTEGER
D26A  FE  FF  C  CP  0FFH  ;TEST FOR ERROR
D26C  28  08  C  JR  2,HILO05  ;IGNORE NOTE IF IN ERROR
D26E  8A  C  CP  D   ;NOTE HIGHER THAN HIGHEST NOTE?
D26F  38  01  C  JR  C,HILO04  ;JUMP IF NOT
D271  57  C  LD  D,A  ;ELSE STORE NEW HIGHEST NOTE
D272  BB  C  HILO04:  CP  E   ;NOTE LOWER THAN LOWEST NOTE?
D273  38  01  C  JR  NC,HILO05  ;JUMP IF NOT
D275  5F  C  LD  E,A  ;ELSE STORE NEW LOWEST NOTE
D276  23  C  HILO05:  INC  HL  ;ADJUST POINTER TO NEXT
D277  23  C  INC  HL   ; NOTE ENTRY
D278  18  BF  C  JR  HILO01  ;CONTINUE WITH NEXT NOTE
D27A  42  C  HILO06:  LD  B,D  ;RETURN HIGHEST NOTE
```
MELODY DISPLAY AND MANIPULATION

D27B 4B C  LD   C,E ;RETURN LOWEST NOTE
D27C  E1 C   POP  HL
D27D  D1 C   POP  DE
D27E  C9 C   RET

D27F  C5 C   PUSH BC
D280  D5 C   PUSH DE
D281  E5 C   PUSH HL
D282  CD D39B C   CALL MOUNT ;MOVE NOTE TO SAFE PLACE
D285  3A 0000H C   LD   A,(XPOSQD) ;TEST FOR TRANSPOSITION FLAG
D288  A7 C   AND  A ;NON-ZERO MEANS TRANSPOSE
D289  20 05 C   JR   NZ,PTST01 ;JUMP TO TRANSPOSE
D288  CD D2EB C   CALL PITCHS ;ELSE DISPLAY PITCH
D28E  1B 1D C   JR   PTST02 ; AND EXIT
D298  CD D0AA C   PTST01: CALL KEYSNP ;CHANGE TO ORIGINAL KEY
D293  EB C   EX   DE,HL ;SAVE ADDRESS OF DATA
D294  46 C   LD   B,(HL) ;GET SCALE DEGREE
D295  23 C   INC  HL ;POINT TO MODIFIER
D296  4E C   LD   C,(HL) ;GET MODIFIER
D297  23 C   INC  HL ;POINT TO OCTAVE
D298  CD D001 C   CALL CUTSTL ;LETTER NAME IN ORIGINAL KEY
D29B  42 C   LD   B,D ;LOAD LETTER NAME
D29C  4B C   LD   C,E ;LOAD ACCIDENTAL
D29D  56 C   LD   D,(HL) ;GET OCTAVE
D29E  CD D0AA C   CALL KEYSNP ;RESET TO NEW KEY
D2A1  CD D7B4 C   CALL TRNPOL ;TRANSPOSE THE LETTER NAME
D2A4  72 C   LD   (HL),D ;REPLACE WITH NEW OCTAVE
D2A5  2B C   DEC  HL ;POINT TO ACCIDENTAL POSITION
PITCH AND ACCIDENTAL DISPLAY

D2A6 71 C LD (HL),C ;REPLACE MODIFIER WITH ACCIDENTAL
D2A7 2B C DEC HL ;POINT TO LETTER POSITION
D2A8 70 C LD (HL),B ;REPLACE SCALE DEGREE WITH LETTER
D2A9 EB C EX DE,HL ;PUT ADDRESS OF DATA IN (DE)
D2AA CD D2FC C CALL PITCHL ;DISPLAY PITCH BY LETTER NAME
D2AD E1 C PTST02: POP HL
D2AE D1 C POP DE
D2AF C1 C POP BC
D2B0 C9 C RET

D2B1 C5 C PITCHLT: PUSH BC
D2B2 D5 C PUSH DE
D2B3 E5 C PUSH HL
D2B4 3A 0000X C LD A,(XPOSOD) ;TEST TRANSPOSITION INDICATOR
D2B7 A7 C AND A ;NON-ZERO MEANS TRANSPOSE
D2B8 20 05 C JR NZ,PTLT01 ;JUMP TO TRANSPOSE
D2B9 CD D2FC C CALL PITCHL ;ELSE DISPLAY PITCH BY LETTER
D2BA 18 15 C JR PTLT02 ;EXIT
D2BF CD D398 C PTLT01: CALL KMOVN ;MOVE NOTE TO SAFE PLACE
D2C2 EB C EX DE,HL ;SAVE DATA ADDRESS
D2C3 46 C LD B,(HL) ;GET LETTER NAME
D2C4 23 C INC HL ;POINT TO ACCIDENTAL
D2C5 4E C LD C,(HL) ;GET ACCIDENTAL
D2C6 23 C INC HL ;POINT TO OCTAVE
D2C7 36 C LD D,(HL) ;GET OCTAVE
D2C8 CD D7B4 C CALL TRNPOL ;TRANSPOSE LETTER NAME
D2C9 72 C LD (HL),D ;REPLACE OCTAVE
D2CC 2B C DEC HL ;POINT TO ACCIDENTAL POSITION
D2CD 71 C LD (HL),C ;REPLACE ACCIDENTAL
D2CE 28 C DEC HL ;POINT TO LETTER NAME
D2CF 70 C LD (HL),B ;REPLACE LETTER NAME
D2D0 EB C EX DE,HL ;PUT DATA ADDRESS IN (DE)
D2D1 CD D2FC C CALL PITCHL ;DISPLAY TRANSPOSED PITCH
PITCH AND ACCIDENTAL DISPLAY

D204 E1 C PTLT02: POP HL
D205 D1 C POP DE
D206 C1 C POP BC
D207 C9 C RET

; PITCHUT
; CONTROLS THE DISPLAY OF A PITCH BY UNADJUSTED PITCH VALUE
; TRANSPOSING IT IF THE TRANSPOSITION INDICATORS ARE SET.
; GIVEN: (B) - UNADJUSTED PITCH
; (C) - OCTAVE DESIGNATION
; (D) - RHYTHMIC VALUE
; (E) - CURRENT POSITION INCREMENT
; RETURNS: (A) - COMPLETION CODE
; = 08H - SUCCESSFUL
; = EOS - END OF STAFF; NOTE NOT DISPLAYED
; = OQRANG - OUT OF RANGE
; = FFH - INVALID DATA

D208 C5 C PITCHUT: PUSH BC
D209 D5 C PUSH DE
D20A E5 C PUSH HL
D20B 3A 0000 C LD A,(XPOSQD) ;TEST TRANSPOSITION INDICATOR
D20C A7 C AND A ;ZERO MEANS NO TRANSPOSE
D20D 28 03 C JR Z,PTUT01 ;JUMP IF NO TRANSPOSE
D20E CD D7F1 C CALL TRNPOU ;ELSE TRANSPOSE UNADJUSTED PITCH
D20F CD D32A C PTUT01: CALL PITCHU ;DISPLAY PITCH
D210 E1 C POP HL
D211 D1 C POP DE
D212 C1 C POP BC
D213 C9 C RET

; PITCH
; DISPLAY PITCH FROM GIVEN SCALE DEGREE
; TAKES THE SCALE DEGREE AND MODIFIER AND CONVERTS IT TO THE
; LETTER NAME/ACCIDENTAL FORMAT FOR DISPLAY BY PITCHL.
; GIVEN: (DE) - ADDRESS OF DATA STREAM IN THE FORMAT:
; SCALE DEGREE, MODIFIER, OCTAVE, RHYTHM,
; CURRENT POSITION INCREMENT
; RETURNS: (A) - COMPLETION CODE
; = 08H - SUCCESSFUL
; = EOS - END OF STAFF; NOTE NOT DISPLAYED
; = OQRANG - OUT OF RANGE
PITCHE: PUSH HL
D2EC EB C
D2ED 46 C
D2EE 23 C
D2EF 4E C
D2F0 CD D801 C
D2F3 73 C
D2F4 2B C
D2F5 72 C
D2F6 EB C
D2F7 CD D2FC C
D2FA E1 C
D2FB C9 C

PITCHL
D2FC E5 C
D2FD EB C
D2FE 46 C
D2FF 23 C
D300 4E C

PITCH AND ACCIDENTAL DISPLAY

C ; = FFH - INVALID DATA
C ; (B) - UNADJUSTED PITCH
C ; (C) - OCTAVE
C ; (D) - RHYTHM
C ; (E) - CURRENT POSITION INCREMENT

C;

PITCHS: PUSH HL
D2EB E5 C
D2EC EB C
D2ED 46 C
D2EE 23 C
D2EF 4E C
D2F0 CD D801 C
D2F3 73 C
D2F4 2B C
D2F5 72 C
D2F6 EB C
D2F7 CD D2FC C
D2FA E1 C
D2FB C9 C

; PITCHL
; DISPLAY PITCH FROM GIVEN LETTER NAME/ACCIDENTAL
; TAKES THE ASCII LETTER NAME AND ACCIDENTAL (PLUS/MINUS FORMAT)
; AND CONVERTS IT TO THE UNADJUSTED PITCH FORMAT FOR DISPLAY
; BY PITCHL. A CHECK IS MADE TO DETERMINE IF THE GIVEN LETTER
; NAME AND ACCIDENTAL ARE VALID AND IF THE PITCH IS A DIATOMIC
; NOTE.
;
; GIVEN: (DE) - ADDRESS OF DATA STREAM IN THE FORMAT:
; LETTER NAME, ACCIDENTAL, OCTAVE, RHYTHM,
; CURRENT POSITION INCREMENT
; RETURNS: (A) - COMPLETION CODE
; = 0CH - SUCCESSFUL
; = EOS - END OF STAFF; NOTE NOT DISPLAYED
; = OORANG - OUT OF RANGE
; = FFH - INVALID DATA
; (B) - UNADJUSTED PITCH
; (C) - OCTAVE
; (D) - RHYTHM
; (E) - CURRENT POSITION INCREMENT

;
PITCH AND ACCIDENTAL DISPLAY

D381 23 C INC HL ;POINT TO OCTAVE
D382 56 C LD D,(HL) ;GET OCTAVE
D383 CD D84D C PITL01: CALL CVLTU ;CONVERT TO UNADJUSTED PITCH
D386 FE FF C CP 8FFH ;CHECK FOR ERROR
D388 28 0D C JR Z,PITL02 ;JUMP IF DATA INVALID
D38A 47 C LD B,A ;SAVE UNADJUSTED PITCH
D38B 4A C LD C,D ;GET OCTAVE IN (C)
D38C 23 C INC HL ;POINT TO RHYTHM
D38D 7E C LD A,(HL) ;GET RHYTHM
D38E CD D9C5 C CALL CVTRHY ;CONVERT RHYTHM
D391 57 C LD D,A ;SAVE RHYTHM
D392 23 C INC HL ;POINT TO INCREMENT
D393 5E C LD E,(HL) ;GET CURRENT POSITION INCREMENT
D394 CD D32A C CALL PITCHU ;DISPLAY THIS UNADJUSTED PITCH
D397 E1 C PITL02: POP HL
D398 C9 C RET

; PITCHI
; DISPLAY PITCH FROM INTEGER.
; GIVEN: (B) - INTEGER PITCH
; (C) - RHYTHM
; (D) - CURRENT POSITION INCREMENT
; RETURNS: (A) - COMPLETION CODE
; = 88H - SUCCESSFUL
; = EOS - END OF STAFF; NOTE NOT DISPLAYED
; = OQRANG - OUT OF RANGE
; = FFH - INVALID DATA

D399 C5 C PITCHI: PUSH BC
D39A D5 C PUSH DE
D39B 5A C LD E,D ;MOVE RHYTHM AND CPI
D39C 51 C LD D,C
D39D CD D957 C CALL CVLTU ;CONVERT INTEGER TO UNADJUSTED
D3A0 FE FF C CP 8FFH ;TEST FOR ERROR
D3A2 28 03 C JR Z,PITI01 ;JUMP ON ERROR
D3A4 CD D32A C CALL PITCHU ;ELSE DISPLAY UNADJUSTED PITCH
D3A7 D1 C PITH01: POP DE
D3A8 C1 C POP BC
D3A9 C9 C RET

; PITCHU
; DISPLAY UNADJUSTED PITCH.
PITCH AND ACCIDENTAL DISPLAY

C ; GIVEN: (B) - UNADJUSTED PITCH
C ; (C) - OCTAVE DESIGNATION
C ; (D) - RHYTHMIC VALUE
C ; (E) - CURRENT POSITION INCREMENT
C ;
C ; RETURNS: (A) - COMPLETION CODE
C ;
C ; = 00H - SUCCESSFUL
C ;
C ; = EOS - END OF STAFF; NOTE NOT DISPLAYED
C ;
C ; = OORANG - OUT OF RANGE
C ;
C ; = FFH - INVALID DATA

D32A C5 C PITCHU: PUSH BC
D32B D5 C PUSH DE
D32C E5 C PUSH HL
D32D 78 C LD A, B ; TEST UNADJUSTED PITCH
D32E E6 0F C AND 0FH ; LOW-ORDER BYTE FOR
D330 D6 03 C SUB 03H ; VALID VALUES IN RANGE:
D332 FE 04 C CP 04H ; 3 TO 6
D334 3E FF C LD A, 0FFH ; LOAD ERROR FLAG
D336 38 12 C JR NZ, PITU81 ; DON'T STORE IF NON-ZERO
D338 A7 C AND A ; CHECK INHIBIT STORE FLAG
D33C 28 83 C JR N 2, PITU81
D33E CD D52A C CALL STORE ; STORE IN TUNE TABLE
D341 CD D3AB C PITU81: CALL ADJUST ; ADJUST PITCH
D344 47 C LD B, A ; ADJUSTED PITCH
D345 4A C LD C, D ; RHYTHMIC VALUE
D346 53 C LD D, E ; CURRENT POSITION
D347 CD D34E C CALL PITCHA ; DISPLAY PITCH
D34A E1 C PITU82: POP HL
D34B D1 C POP DE
D34C C1 C POP BC
D34D C9 C RET

C ;
C ; PITCHA
C ;
C ; CONTROLS DISPLAY OF NOTE, ACCIDENTAL AND LEDGER LINES.
C ; BEFORE EACH NOTE IS DISPLAYED, THE MEASURE IS SCANNED
C ; BACKWARDS TO SEE IF AN ACCIDENTAL IS NEEDED FOR THE
C ; CURRENT NOTE.
C ;
C ; GIVEN: (B) - ADJUSTED PITCH
C ; (C) - RHYTHMIC VALUE
C ; (D) - CURRENT POSITION INCREMENT
C ;
C ; RETURNS: (A) - COMPLETION CODE
C ;
C ; = 00H - SUCCESSFUL
C ;
C ; = EOS - END OF STAFF; NOTE NOT DISPLAYED
C ;
C ; = OORANG - OUT OF RANGE
PITCH AND ACCIDENTAL DISPLAY

D34E C5 C PITCHA: PUSH BC
D34F D5 C PUSH DE
D350 E5 C PUSH HL
D351 FD 5E 05 C LD E,(IY+CURPOS) ; TEST FOR END OF STAFF
D354 FD 7E 02 C LD A,(IY+PARLEN)
D357 93 C SUB E ; HOW MUCH ROOM LEFT ON STAFF?
D358 38 04 C JR C,PITA01 ; JUMP IF ALREADY GONE TOO FAR
D35A FE 04 C CP 04H ; MINIMUM SPACE FOR 1 PITCH
D35C 38 04 C JR NC,PITA02 ; JUMP IF OK
D35E 3E 10 C PITA01: LD A,EOS ; ELSE LOAD END OF STAFF FLAG
D360 18 35 C PITA02: LD A,B ; GET PITCH
D362 78 C AND A ; TEST FOR TOO HIGH
D364 28 04 C JR Z,PITA03 ; JUMP IF TOO HIGH
D366 FE 04 C CP 0FFH ; TEST FOR TOO LOW
D368 28 0E C JR NZ,PITA04 ; JUMP IF IN RANGE
D36A CD D41D C PITA03: CALL OUTRNG ; ELSE OUT OF RANGE
D36D FD 7E 05 C LD A,(IY+CURPOS) ; ADJUST CURRENT POSITION
D370 82 C ADD A,D
D371 FD 77 05 C LD (IY+CURPOS),A
D374 3E 16 C LD A,0000H ; LOAD OUT OF RANGE CODE
D376 18 1F C JR PITA08
D378 3A 0000X C PITA04: LD A,(UNCADO) ; UNCONDITIONAL ACCIDENTAL FLAG
D37B A7 C AND A ; SET ZERO FLAG
D37C 78 C LD A,B ; RESTORE PITCH
D37D 20 0E C JR NZ,PITA05 ; JUMP IF UNCONDITIONAL
D37F CD D42E C CALL A0ARRY ; SEE IF A0NTL NEEDED
D382 FE 01 C CP 01H ; JUMP IF EXACT PITCH
D384 28 0A C JR Z,PITA06 ; ALREADY PRINTED
D386 78 C LD A,B ; RESTORE PITCH
D387 38 04 C JR NC,PITA05 ; JUMP IF MUST PRINT
D389 FE 60 C CP 60H ; JUMP AROUND IF NO
D38B 38 04 C JR C,PITA07 ; A0NTL NEEDED
D38D CD D507 C PITA05: CALL A0NTL ; PRINT ACCIDENTAL
D390 78 C PITA06: LD A,B ; SET UP REGISTERS FOR
D391 41 C PITA07: LD B,C ; CALL TO DISPLAY NOTE
D392 4A C LD C,D
D393 CD D466 C CALL DNOTE ; DISPLAY NOTE
D396 AF C XOR A ; SET SUCCESSFUL RETURN CODE
D397 E1 C PITA08: POP HL
D398 D1 C POP DE
D399 C1 C POP BC
D39A C9 C RET
PITCH AND ACCIDENTAL DISPLAY

; MOVE THE NOTE TO A TEMPORARY WORK AREA SO THAT THE
; DATA MAY BE OVERWRITTEN AS IT IS TRANSPOSED AND
; CONVERTED FROM SCALE DEGREE TO LETTER NAME.
; GIVEN: (DE) - POINTER TO NOTE TO BE MOVED
; RETURNS: (DE) - POINTER TO SAME NOTE IN NEW LOCATION

MOVNUT: PUSH BC
PUSH HL
LD HL,WRKNOT ;DESTINATION
EX DE,HL ;EXCHANGE SOURCE AND DESTINATION
PUSH DE ;SAVE DESTINATION
LD BC,0005H ;LENGTH IS 5
LDIR ;MOVE IT
POP DE ;RESTORE NEW ADDRESS
POP HL
POP BC
RET

ADJUST

; ADJUST UNADJUSTED PITCH FOR OCTAVE AND STAFF ORIENTATION.
; ADJUSTED PITCH IS IN THE FORMAT:
; BIT VALUE MEANING
; 7-5 000-111 ACCIDENTAL IN THE SAME FORM
; 4 0 NOTE ON LINE
; 1 0 NOTE ON SPACE
; 3-0 0000-1111 LINE NUMBER FROM TOP OF PARTITION

GIVEN: (B) - UNADJUSTED PITCH
(C) - OCTAVE DESIGNATION (+/- 8-BIT VALUE)

RETURNS: (A) - ADJUSTED PITCH

ADJUST:
PUSH BC
PUSH DE
PUSH HL
LD A,B ;SET UP FOR CALL
LD B,C
CALL OCTADJ ;ADJUST FOR OCTAVE
CALL STFADJ ;ADJUST FOR STAFF AND CLEF
POP HL
POP DE
POP BC
RET
MUSIC CONTROL PROGRAM MACRO-88 3.44 89-Dec-81 PAGE 1-64

PITCH AND ACCIDENTAL DISPLAY

\[ \text{D38A D5 C OCTADJ: PUSH DE} \]
\[ \text{D38B E5 C PUSH HL} \]
\[ \text{D38C 4F C LD C,A} \quad \text{;SAVE UNADJUSTED PITCH} \]
\[ \text{D38D E6 87 C AND 87H} \quad \text{;SET LOW ORDER 3 BITS} \]
\[ \text{D38E 5F C LD E,A} \quad \text{;SAVE UNADJUSTED PITCH} \]
\[ \text{D38F 78 C LD A,B} \quad \text{;GET OCTAVE} \]
\[ \text{D390 A7 C AND A} \quad \text{;TEST FOR ZERO OCTAVE} \]
\[ \text{D391 7B C LD A,E} \quad \text{;RESTORE UNADJUSTED DISPLACEMENT} \]
\[ \text{D392 20 1E C JR Z,OCA04} \quad \text{;EXIT IF ZERO OCTAVE} \]
\[ \text{D393 21 FC6D C LD HL,0FCFDH} \quad \text{; -4 AND -3 OCTAVE ADJUSTMENT} \]
\[ \text{D394 CB 78 C BIT 7B} \quad \text{;OCTAVE BELOW OR ABOVE ZERO OCTAVE?} \]
\[ \text{D395 28 88 C JR Z,OCA01} \quad \text{;JUMP IF ABOVE ZERO OCTAVE} \]
\[ \text{D396 78 C LD A,B} \quad \text{;ABSOLUTE VALUE OF OCTAVE} \]
\[ \text{D397 ED 44 C NEG} \]
\[ \text{D398 47 C LD B,A} \quad \text{;RESTORE UNADJUSTED DISPLACEMENT} \]
\[ \text{D399 7B C LD A,E} \quad \text{;+3 AND +4 OCTAVE ADJUSTMENT} \]
\[ \text{D39A 21 8304 C LD HL,8304H} \]
\[ \text{D39B CB 61 C OCA01: BIT LINSPC,C} \quad \text{;TEST LINE/SPACE INDICATOR} \]
\[ \text{D39C 28 05 C JR Z,OCA02} \quad \text{;JUMP IF ON LINE ELSE ON SPACE} \]
\[ \text{D39D 54 C LD D,H} \quad \text{;OCTAVE ADJUSTMENT FOR NOTE ON SPACE} \]
\[ \text{D39E 80 A1 C RES LINSPC,C} \quad \text{;NOTE IS NON ON LINE} \]
\[ \text{D39F 18 83 C JR OCA03} \quad \text{;60 ADD ADJUSTMENT} \]
\[ \text{D3A0 55 C OCA02: LD D,L} \quad \text{OCTAVE ADJUSTMENT FOR NOTE ON LINE} \]
\[ \text{D3A1 CB E1 C SET LINSPC,C} \quad \text{;NOTE IS NON ON SPACE} \]
\[ \text{D3A2 82 C OCA03: ADD A,D} \quad \text{ADD OCTAVE ADJUSTMENT} \]
\[ \text{D3A3 10 F1 C DJNZ OCA01} \quad \text{CONTINUE FOR EACH OCTAVE} \]
\[ \text{D3A4 C9 C OCA04: POP HL} \]
\[ \text{D3A5 D1 C POP DE} \]
\[ \text{D3A6 C9 C RET} \]
STFADJ: STAFF ADJUST

STAFF ADJUST Adjusts octave-adjusted pitch for the staff orientation and clef in use. The 8-bit octave-adjusted pitch is adjusted to a positive 4-bit value and is concatenated with the accidental and line/space indicator to form the adjusted pitch.

Given:
- (A) - octave-adjusted pitch
- (C) - original unadjusted pitch with adjusted line/space indicator
- (IY) - address of active partition table entry

Returns:
- (A) - adjusted pitch

```
D3E6 C5 C STFADJ: PUSH BC
D3E7 E5 C PUSH HL
D3E8 47 C LD B, A ; save octave-adjusted pitch
D3E9 FD 7E 04 C LD A, (IY + STFORM) ; load staff orientation
D3EC 80 C ADD A, B ; adjust for it
D3ED 47 C LD B, A ; save result
D3EE 21 C4C0 C LD HL, CLFDIF ; point to clef differential table
D3F1 FD 7E 06 C LD A, (IY + CLFCOND) ; get clef code
D3F4 85 C ADD A, L ; point to differential for clef
D3FA 6F C LD L, A
D3F6 30 01 C JR NC, STD01 ; test for overflow
D3F8 24 C INC H ; adjust for overflow
D3F9 7E C STD01: LD A, (HL) ; get differential
D3FA 80 C ADD A, B ; adjust for it
D3FB 28 10 C JR Z, STD05 ; pitch is too high to be
D3FD CB 7F C BIT 7, A ; displayed if result is
D3FF 28 03 C JR Z, STD02 ; zero or negative
D401 6F C XOR A ; set too-high flag
D402 18 16 C JR STD05 ; and exit
D404 47 C STD02: LD B, A ; save adjusted displacement
D405 FD 7E 03 C LD A, (IY + PARLIND) ; get number of lines in partition
D408 D6 02 C SUB 02H ; no print on bottom line allowed
D40A B0 C CP B ; check for too low
D40B 38 05 C JR C, STD03 ; jump if too low
D40D 78 C LD A, B ; else restore displacement and
D40E FE 10 C CP 10H ; check for overflow
D410 38 04 C JR C, STD04 ; jump if in range
D412 3E FF C STD03: LD A, 0FFH ; load too low flag
D414 18 04 C JR STD05 ; and exit
D416 79 C STD04: LD A, C ; load unadjusted pitch
D417 E6 F0 C AND 0FFH ; get high order 4 bits only
D419 88 C OR B ; put with adjusted displacement
```
PITCH AND ACCIDENTAL DISPLAY

STAD85: POP HL
POP BC
RET

OUTRNG
Routine for handling pitches that are out of the pitch range of the staff as it is currently displayed. An asterisk is displayed in lieu of the actual pitch.

Given: (A) - Too high/too low flag

= 80H - Too high
= FFH - Too low

Returns: None

OUTRNG: PUSH HL
AND A ;Test for too high
JR Z,OTRNG01 ;Jump if so
LD A,(IY+PARLIN) ;Find bottom of partition
INC A
CALL SACALC ;Find address
LD (HL),ASTRSK ;Display asterisk
POP HL
RET

ACARRY
Accidental carry
Checks to see which accidental is in effect in the current measure for the current note before it is displayed. If the accidental has already been printed there is no need to repeat it. If an accidental has been used on a note that is normally diatonic, the accidental needs to be canceled. Assumes the note has already been stored in the tune table.

Given: (IX) - Pointer to current note in tune table
Returns: (A) = 80H if pitch letter not in measure i.e., print the accidental if the note calls for one
= 01H if exact pitch already displayed i.e., do not print any accidental
= 02H if pitch letter found in measure but with unequal accidental
MUSIC CONTROL PROGRAM  MACRO-80  3.44  09-Dec-81  PAGE  1-67

PITCH AND ACCIDENTAL DISPLAY

C ;  
C ;  i.e., ACCIDENTAL MUST BE DISPLAYED
C ;
D42E  C5  C ;  ACARRY: PUSH BC
D42F  D5  C ;  PUSH DE
D438  DD  E5  C ;  PUSH IX
D432  DD  A6  00  C ;  LD $B,(IX+UNAPIT) ; CURRENT UNADJUSTED PITCH
D433  DD  4E  01  C ;  LD $C,(IX+OCTAVE) ; CURRENT PITCH OCTAVE
D438  11  FFFC  C ; ACRY01: LD DE,TELTC ; MINUS TUN-TBL ENTRY LENGTH
D43B  DD  7F  00  C ; ADD IX,DE ; POINT TO PREVIOUS PITCH
D440  FE  FE  C ; CP TTSBAR ; CHECK FOR BAR LINE
D442  28  03  C ; JR NZ,ACRY82 ; JUMP IF BAR NOT FOUND
D444  AF  C ; XOR A ; ZERO (A) WHEN LETTER NOT FOUND
D445  18  1A  C ; JR ACRY04 ; AND EXIT
D447  57  C ; ACRY02: LD D,A ; SAVE PREVIOUS NOTE
D448  E6  IF  C ; AND IFH ; GET LETTER NAME ONLY
D44A  5F  C ; LD E,A ; SAVE IT
D44B  78  C ; LD A,B ; GET CURRENT PITCH
D44C  E6  IF  C ; AND IFH ; GET ITS PITCH LETTER
D44E  BB  C ; CP E ; ARE LETTER NAMES THE SAME?
D44F  28  E7  C ; JR NZ,ACRY01 ; JUMP IF NOT SAME
D451  78  C ; LD A,B ; ELSE COMPARE ACCIDENTALS
D452  BA  C ; CP D
D453  28  04  C ; JR 2,ACRY03 ; JUMP IF EQUAL
D455  3E  02  C ; LD A,02H
D457  18  08  C ; JR ACRY04
D459  79  C ; ACRY03: LD A,C ; COMPARE OCTAVES
D45A  DD  BE  01  C ; CP (IX+OCTAVE)
D45D  28  D9  C ; JR NZ,ACRY01 ; JUMP IF OCTAVES UNEQUAL
D45F  3E  01  C ; LD A,01H ; ELSE EXACT PITCH FOUND
D461  DD  E1  C ; ACRY04: POP IX
D463  D1  C ; POP DE
D464  C1  C ; POP BC
D465  C9  C ; RET

C ;
C ; DNOTE
C ;
C ; DISPLAY NOTE ON SCREEN
C ; CONTROLS DISPLAY OF NOTE. LEDGER LINES ARE PRINTED FIRST.
C ; THE RHYTHM BYTE IS USED TO DETERMINE WHICH NOTE CHARACTER
C ; VECTOR IS USED FOR DISPLAY. THE CURRENT POSITION IS THEN
C ; INCREMENTED USING EITHER THE DEFAULT FOR THAT NOTE VALUE
C ; OR THE EXPLICIT INCREMENT GIVEN.
C ;
C ; GIVEN:  (A) - ADJUSTED PITCH
C ;  (B) - RHYTHMIC VALUE
C ;  (C) - CURRENT POSITION INCREMENT
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-68

PITCH AND ACCIDENTAL DISPLAY

C ; (IY) - ADDRESS OF ACTIVE PARTITION TABLE
C ; RETURNS: NONE
C ;
D466 C5 C DNOTE: PUSH BC
D467 D5 C PUSH DE
D468 E5 C PUSH HL
D469 CD D4F8 C CALL LEDGER ;PRINT LEDGER LINES IF NEEDED
D46C CD D494 C CALL PCHARS ;GET CHARSET VECTOR IN (HL)
D46F EB C EX DE,HL ;SAVE IT IN DE
D470 E6 0F C AND 0FH ;GET PITCH DISPLACEMENT
D472 CD DE64 C CALL SACALC ;GET SCREEN ADDRESS IN (HL)
D475 EB C EX DE,HL ; (DE)-SCREEN : (HL)-CHAR
D476 CD DEA1 C CALL DISPLAY ;DISPLAY NOTE
D479 79 C LD A,C ;CURRENT POSITION INCREMENT
D47A A7 C AS5 A ;CHECK FOR DEFAULT REQUEST
D47B 28 0C C JR NZ,DNOT02 ;JUMP IF EXPLICIT
D47D 21 C8D4 C LD HL,CPIDEF ;RHYTHMIC DEFAULTS TABLE
D480 78 C LD A,B ;GET RHYTHMIC VALUE
D481 E6 07 C AND 07H ;USE AS INDEX INTO TABLE
D483 85 C ADD A,L
D484 6F C LG L,A
D485 38 01 C JR NC,DNOT01 ;TEST FOR OVERFLOW
D487 24 C INC H ;ADJUST FOR OVERFLOW
D488 4E C DNOT01: LD C,(HL) ;GET DEFAULT
D489 FD 7E 05 C DNOT02: LD A, (IY+CURPOS) ;GET CURRENT POSITION
D48C 81 C ADD A,C ;INCREMENT IT
D48D FD 77 05 C LD (IY+CURLPOS),A ;STORE IT
D498 E1 C POP HL
D499 D1 C POP DE
D49C C1 C POP BC
D49D C9 C RET
C ;
C ; PCHARS
C ;
C ; PITCH CHARACTERS
C ; DETERMINES WHICH SUBROUTINE IS TO BE USED TO FIND THE
C ; CHARACTER VECTOR FOR THE NOTE TO BE DISPLAYED. USING
C ; THE RHYTHMIC VALUE AS AN INDEX INTO A SUBROUTINE JUMP
C ; TABLE, THE ADDRESS OF THE APPROPRIATE SUBROUTINE IS
C ; JUMPED TO. THE SUBROUTINE THEN RETURNS DIRECTLY TO
C ; THE ROUTINE WHICH CALLED THIS ROUTINE.
C ;
C ; GIVEN: (A) - ADJUSTED PITCH
C ; (B) - RHYTHMIC VALUE
C ; RETURNS: (HL) - ADDRESS OF CHARACTER VECTOR
C ; (A) - UNCHANGED
C ;
PITCH AND ACCIDENTAL DISPLAY

WHLCHR: PUSH BC

WHCHR: PUSH BC

WHLCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: PUSH BC

WHCHR: P
BLACK WHOLE NOTE CHARACTERS
FINDS THE CHARACTER VECTOR FOR BLACK WHOLE NOTES ACCORDING TO THE POSITION OF THE NOTE ON THE STAFF.

GIVEN: (A) - ADJUSTED PITCH
(IY) - CURRENT ACTIVE PARTITION TABLE
RETURNS: (HL) - ADDRESS OF WHOLE NOTE CHARACTER VECTOR
(A) - UNCHANGED

LEDGER
DETERMINES IF LEDGER LINES ARE NEEDED AND PRINTS THEM IF SO.
GIVEN: (A) - ADJUSTED PITCH
(IY) - ACTIVE PARTITION TABLE
RETURNS: (A) - UNCHANGED
(DE) - CHANGED

LEDGER: PUSH AF
PUSH BC
LD C,A ;SAVE PITCH
D4F3 E6 0F C AND #FH ;GET DISPLACEMENT
D4F5 47 C LD B,A
D4F6 FD 7E 04 C LD A,(IX+5TFORN) ;GET STAFF ORIENTATION
D4F9 5F C LD E,A ;SAVE IT
D4FA D4 02 C SUB 02H ;MUST BE MINIMUM OF 2 LINES
D4FC 90 C SUB B ; ABOVE STAFF TO PRINT
D4FF 68 C INE A ;NUMBER OF LINES TO PRINT
D500 47 C LD B,A ;SAVE IT
D501 7B C LD A,E ;GET STAFF ORIENTATION
D502 CD DE64 C CALL SACALC ;GET SCREEN ADDRESS
D505 11 FFC0 C LD DE,NEGLEN ;NEGATIVE LINE LENGTH
D508 18 18 C JR LDGR01 ;GO PRINT
D50A 7B C LDGR01: LD A,E ;GET STAFF ORIENTATION
D50B 86 C ADD A,B ;POINT TO BOTTOM OF STAFF + 2
D50D CB 61 C BIT LINSPEC,C ;TEST LINE/SPACE BIT
D50F 20 01 C JR Z,LDGR02 ;JUMP IF ON LINE
D511 85 C DEC B ;ADJUST FOR NOTE ON SPACE
D512 4F C LDGR02: LD C,A ;SAVE BOTTOM OF STAFF
D513 7B C LD A,B ;GET NOTE DISPLACEMENT
D514 91 C SUB C ;NOTE MinUS BOTTOM OF STAFF
D515 30 10 C JR C,LDGR04 ;JUMP IF NONE TO PRINT
D517 3C C INC A ;NUMBER OF LINES TO PRINT
D518 47 C LD B,A ;PUT IN (B)
D519 7B C LD A,E ;GET STAFF ORIENTATION
D51A 04 84 C ADD A,(IX) ;BOTTOM LINE OF STAFF
D51B CD DE64 C CALL SACALC ;ADDRESS OF BOTTOM LINE
D51F 11 0040 C LD DE,LINELEN ;GET LINE LENGTH
D522 19 C LDGR03: ADD HL,DE ;POINT TO NEXT LINE
D523 3E 98 C LD (HL),LINE ;PRINT LINE
D525 10 FB C DJNZ LDGR03
D527 C1 C LDGR04: POP BC
D528 F1 C POP AF
D529 C9 C RET

; ; ; STORE
C ; ; STORE NOTE IN TUNE TABLE
C ; ; GIVEN: (B) - UNADJUSTED PITCH
C ; ; (C) - OCTAVE
C ; ; (D) - RHYTHM BYTE
C ; ; (E) - CURRENT POSITION INCREMENT
C ; ; (IX) - ADDRESS OF PREVIOUS NOTE IN TUNE TABLE
C ; ; RETURNS: NONE
C ;

D52A C5 C STORE: PUSH BC
PITCH AND ACCIDENTAL DISPLAY

PUSH DE
LD DE,TLELEN ;ADJUST (IX) TO POINT TO NEXT
ADD IX,DE ;NOTE IN TUNE TABLE
PUSH IX
POP BC
LD DE,TLELEN ;END OF TUNE TABLE ADDRESS
CALL DCOMP ;TEST FOR END OF TABLE
JR NC,STOR1 ;JUMP IF NOT END OF TABLE
LD IX,TUNBL ;ELSE POINT BACK TO BEGINING
STOR01: POP DE ;RESTORE REGISTERS
POP BC
PUSH DE
PUSH HL
JR NC,ERAN01 ;JUMP IF NOT END OF TABLE
LD IX+UNAPIT);GET UNADJUSTED PITCH
CP 60H ;ACCIDENTAL DIATONIC OR NOT?
JR NC,ERAN02 ;JUMP IF NON-DIATONIC
ADD A,00H ;ELSE MAKE IT NON-DIATONIC

GIVEN: (IX) - POINTER TO CURRENT NOTE IN TUNE TABLE
(IX) - CURRENT PARTITION TABLE ENTRY
RET
ERAN82: INC C  
CP 0E0H  ;DOUBLe SHARP AND SINGLE FLAT  
JR NC,ERAN83  ;USE ONLY 1 MORE COLUMN WHILE  
CP 0A0H  ;SINGLE SHARP AND DOUBLE FLAT  
JR C,ERAN83  ;NEED 2 MORE COLUMNS  
INC C  ;ADD 1 MORE COLUMN  
LD A,(IY+CURPOS)  ;ADJUST CURRENT POSITION TO  
LD D,(IX+CPINCR)  ;THAT OF PREVIOUS NOTE  
INC C  ;NUMBER OF COLUMNS  
LD (IY+CURPOS),A  
LD B,A  ;STARTING COLUMN NUMBER  
CALL ERACOL  ;ERASE THE COLUMNS  
LD DE,TELTC  ;MINUS TUNE TABLE ENTRY LENGTH  
ADD IX,DE  ;POINT TO PREVIOUS NOTE  
PUSH BC  
PUSH DE  
PUSH HL  
LD A,(IY+CURPOS)  ;GET CURRENT POSITION  
LD (TEMP01),A  ;SAVE IT  
LD (IY+CURPOS),B  ;GET CURRENT POSITION  
XOR A  ;GET ADDRESS OF TOP LINE  
CALL SACALC  ;OF PARTITION  
LD B,(IY+PARLIN)  ;GET NUMBER OF LINES IN PARTITION  
LD A,C  ;NUMBER OF COLUMNS TO ERASE  
LD DE,LINELEN  ;LENGTH OF PHYSICAL LINE  
ERAC01:  ;SAVE START ADDRESS  
PUSH HL  
ERAC02:  ;BLANK OUT POSITION  
LD (HL),SPACE
PITCH AND ACCIDENTAL DISPLAY

D5AC 19 C ADD HL,DE ;POINT TO NEXT ROW
D5AD 18 FB C DJNZ ERAC02 ;CONTINUE DOWN ENTIRE PARTITION
D5AF E1 C POP HL ;RESTORE STARTING ADDRESS
D5B0 23 C INC HL ;POINT TO NEXT COLUMN
D5B1 FD 46 03 C LD B,(IY+PARLIN) ;RESTORE NUMBER OF LINES
D5B4 3D C DEC A ;COUNT NUMBER OF COLUMNS
D5B5 20 F2 C JR NZ,ERAC01 ;JUMP IF MORE COLUMNS TO DO
D5B7 FD 7E 04 C LD A,(IY+STFORD) ;GET ADDRESS OF TOP OF STAFF
D5BA CD DE64 C CALL SACALC ;AT SAME COLUMN
D5BD 06 05 C LD B,05H ;NUMBER OF LINES IN STAFF
D5BE 3E 98 C LD A,HLINE ;STAFF LINE CHARACTER
D5C1 E5 C ERAC03: PUSH HL ;SAVE STARTING ADDRESS
D5C2 77 C ERAC04: LD (HL),A ;REPLACE STAFF CHARACTER
D5C3 19 C ADD HL,DE ;POINT TO NEXT LINE
D5C4 10 FC C DJNZ ERAC04 ;UNTIL BOTTOM OF STAFF
D5C6 E1 C POP HL ;RESTORE STARTING ADDRESS
D5C7 23 C INC HL ;POINT TO NEXT COLUMN
D5C8 06 05 C LD B,05H ;RESET NUMBER OF LINES
D5CA 80 C DEC C ;COUNT NUMBER OF COLUMNS
D5CB 20 F4 C JR NZ,ERAC03 ;JUMP IF NOT DONE
D5CD 3A 0080X C LD A,(TEMPS 1) ;RESET CURRENT POSITION
D5D0 FD 77 05 C LD (IY+CURPOS),A
D5D3 E1 C POP HL
D5D4 01 C POP DE
D5D5 C1 C POP BC
D5D6 C9 C RET

; MACLIB B:HCMP05.MAC

; ACNTL

; DISPLAY ACCIDENTAL AT CURRENT POSITION. TYPE OF
; ACCIDENTAL IS DETERMINED FROM THE CURRENT PITCH.

; GIVEN: (A)  - ADJUSTED PITCH
; RETURNS: NONE

ACNTL: PUSH BC
D5D7 C5 C PUSH DE
D5D8 D5 C PUSH HL
D5DA 47 C LD B,A ;SAVE PITCH
D5DB E6 8F C AND 0FH ;GET SCREEN ADDRESS OF
D5DC 3D C DEC A ; ACCIDENTAL WHICH IS
D5DE CD DE64 C CALL SACALC ;1 LINE ABOVE NOTE ADDRESS
D5E1 EB C EX DE,HL ;PASS ADDRESS IN (DE)
D5E2 78 C LD A,B ;RESTORE PITCH
PITCH AND ACCIDENTAL DISPLAY

D5E3 FE 60 C CP 60H ; JUMP IF PITCH IS DIATONIC
D5E5 38 03 C JR NC,ACDT01 ; NON-DIATONIC
D5E7 C6 60 C ADD A,60H ; MAKE DIATONIC PITCH APPEAR
D5E9 47 C LD B,A ; NON-DIATONIC
D5EA FE 80 C ACDT01: CP 80H ; COMPARISON WITH FLAT VALUE
D5EC 38 05 C JR NC,ACDT02 ; JUMP IF FLAT OR SHARP
D5EE CD D611 C CALL NATURL ; ELSE DISPLAY NATURAL
D5F1 18 1A C JR ACDT85
D5F3 CB 6F C ACDT02: BIT FLTSHP,A ; BIT = 0 IF FLAT, 1 IF SHARP
D5F5 20 13 C JR NZ,ACDT04 ; JUMP IF SHARP
D5F7 3E 01 C LD A,01H ; ASSUME ONLY ONE FLAT
D5F9 32 0000% C LD (DBLFLT),A
D5FC CB 70 C BIT DOUBLE,B ; IS IT A DOUBLE FLAT?
D5FE 28 04 C JR Z,ACDT83 ; JUMP IF NOT
D600 3C C INC A ; ELSE SET DOUBLE FLAT FLAG
D601 32 0000% C LD (DBLFLT),A
D604 7D C ACDT03: LD A,B ; RESTORE PITCH
D605 CD D61B C CALL FLAT ; DISPLAY FLAT
D608 18 03 C JR ACDT05
D60A CD D63B C ACDT04: CALL SHARP ; DISPLAY SHARP
D60D E1 C ACDT05: POP HL
D60E D1 C POP DE
D60F C1 C POP BC
D610 C9 C RET

; NATURAL
; PRINTS NATURAL SIGN STARTING AT ADDRESS GIVEN IN (DE) AND UPDATES CURRENT POSITION.
; GIVEN: (A) - ADJUSTED PITCH
; (DE) - SCREEN ADDRESS
; RETURNS: NONE

D611 CD D65B C NATURL: CALL NCCHARS ; GET DISPLAY CHARACTERS
D614 CD DEA1 C CALL DISPLAY ; DISPLAY NATURAL
D617 FD 34 05 C INC (IY+CURPOS) ; INCREMENT CURRENT POSITION
D61A C9 C RET

; FLAT
; PRINT FLAT SIGN AT POSITION GIVEN IN (DE) AND UPDATE CURRENT POSITION. IF DOUBLE FLAT, PRINT BOTH.
; GIVEN: (A) - ADJUSTED PITCH
; (DE) - CURRENT SCREEN ADDRESS
PITCH AND ACCIDENTAL DISPLAY

C; RETURNS: NONE

C;

D61B CD D6CA C FLAT: CALL FCHARS ;GET DISPLAY CHARACTERS
D61E 1B C DEC DE
D61F 13 C FLAT01: INC DE ;POINT TO NEXT COLUMN
D620 CD DEA1 C CALL DISPLAY ;PRINT FLAT
D623 FD 34 05 C INC (IY+CURPOS) ;INCREMENT CURRENT POSITION
D626 3A 0080X C LD A,(DBLFLT) ;GET DOUBLE FLAT FLAG
D629 3D C DEC A
D62A 32 0000X C LD (DBLFLT),A ;RESET IT
D62D 28 F0 C JR NZ,FLAT01
D62F C9 C RET

C;

C;

D630 CB 77 C SHARP: BIT DOUBLE,A ;TEST DOUBLE SHARP BIT
D632 2B 13 C JR Z,SHRPO2 ;JUMP IF SINGLE SHARP
D634 CB 67 C BIT LINSPC,A ;TEST LINE/SPACE BIT
D636 20 05 C JR NZ,SHRPO1 ;JUMP IF ON SPACE
D638 21 0040 C LD HL,LINLEN ;DOUBLE SHARP ON LINE IS
D63B 19 C ADD HL,DE ;ONLY ACCIDENTAL THAT STARTS
D63C EB C EX DE,HL ;PRINT AT SAME LINE AS NOTE
D63D CD 0776 C SHRPO1: CALL DSCHAR ;GET DOUBLE SHARP CHAR ADDR
D640 CD DEA1 C CALL DISPLAY ;PRINT DOUBLE SHARP
D643 3E 01 C LD A,01H ;NUMBER OF CHAR POSITIONS TO
D645 18 8D C JR SHRPO4 ;UPDATE
D647 CB 67 C SHRPO2: BIT LINSPC,A ;SHARPS ON A LINE PRINT ON THE
D649 20 01 C JR NZ,SHRPO3 ;COLUMN BEHIND THE STARTING
D64B 13 C INC DE ;POSITION
D64C CD 0710 C SHRPO3: CALL SCHARS ;GET SHARP CHARACTERS
D64F CD DEA1 C CALL DISPLAY ;PRINT SHARP
D652 3E 82 C LD A,02H ;NUMBER OF CHAR POSITIONS USED
D654 FD 84 05 C SHRPO4: ADD A,(IY+CURPOS) ;UPDATE CURRENT POSITION
D657 FD 77 05 C LD (IY+CURPOS),A
D65A C9 C RET

C;

C;

C; NCCHARS

C;

C; NATURAL SIGN CHARACTERS
PITCH AND ACCIDENTAL DISPLAY

; DETERMINES WHICH NATURAL SIGN CHARACTERS ARE TO BE USED
; DEPENDING UPON THE POSITION OF THE ACCIDENTAL ON THE STAFF.
; GIVEN: (A) - ADJUSTED PITCH
; (IY) - ADDRESS OF ACTIVE PARTITION TABLE
; RETURNS: (HL) - ADDRESS OF CHARACTER VECTOR

D65B  C5 C NCHARS: PUSH BC ;SAVE PITCH
D65C  47 C LD B,A
D65D  E6 0F C AND 0FH ;GET DISPLACEMENT ONLY
D65F  4F C LD C,A ; AND KEEP IT IN C
D668  CB 68 C BIT LNSPC,B ;TEST LINE/SPACE BIT
D662  28 33 C JR NZ,NCHR04 ;JUMP IF ON SPACE
D664  21 C4D4 C LD HL,NATLAN ;ASSUME ABOVE OR BELOW STAFF
D667  FD 7E 04 C LD A,(IY+STFORN) ;COMPARE PITCH POSITION
D66A  47 C LD B,A ; TO STAFF ORIENTATION
D66B  30 C DEC A
D66C  B9 C CP C
D66D  38 87 C JR C,NCHR01 ;JUMP IF NOT ABOVE STAFF
D66F  28 57 C JR NZ,NCHR08 ;DONE IF ABOVE STAFF
D671  21 C4DA C LD HL,NATLAL
D674  18 52 C JR NCHR08
D676  C6 06 C NCHR01: ADD A,06H ;CHECK FOR BELOW STAFF
D677  B9 C CP C
D679  38 40 C JR C,NCHR08 ;DONE IF BELOW
D67B  28 05 C JR NZ,NCHR02 ;ELSE JUMP IF IN MIDDLE STAFF
D67D  21 C4E0 C LD HL,NATLBL
D67F  18 46 C JR NCHR08
D682  21 C4E6 C NCHR02: LD HL,NATLAL ;MIDDLE STAFF
D685  30 C DEC A
D686  B9 C CP C
D687  28 85 C JR NZ,NCHR03
D689  21 C4EC C LD HL,NATLMB
D68C  18 3A C JR NCHR08
D68E  78 C NCHR03: LD A,B ;STAFF ORIENTATION
D68F  B9 C CP C
D690  28 36 C JR NZ,NCHR08
D692  21 C4F2 C LD HL,NATLHT
D695  18 31 C JR NCHR08
D697  21 C4FB C NCHR04: LD HL,NATSAN ;ON SPACE ABOVE/BELOW STAFF
D69A  FD 7E 04 C LD A,(IY+STFORN)
D69D  47 C LD B,A
D69E  30 C DEC A
D69F  B9 C CP C
D6A0  38 87 C JR C,NCHR05 ;JUMP IF NOT ABOVE STAFF
D6A2  28 24 C JR NZ,NCHR08
D6A4  21 C4FE C LD HL,NATSAB
D6A7  18 1F C JR NCHR08
MUSIC CONTROL PROGRAM  MACRO-80 3.44  09-Dec-81  PAGE 1-78

PITCH AND ACCIDENTAL DISPLAY

D6A9  C6 06  C  NCHR05: ADD A,06H ;POINT TO BOTTOM OF STAFF
D6AB  B9  C  CP C
D6AC  38 1A  C  JR C,NCHR08
D6AE  20 05  C  JR NZ,NCHR06
D6B0  21 C504  C  LD HL,NATSBT
D6B3  18 13  C  JR NCHR08
D6B5  30  C  NCHR06: DEC A  ;TEST FOR BOTTOM SPACE
D6B6  B9  C  CP C
D6B7  20 05  C  JR NZ,NCHR07 ;JUMP IF NOT ON BOTTOM SPACE
D6B9  21 C516  C  LD HL,NATSMB
D6BC  18 0A  C  JR NCHR08
D6BE  21 C50A  C  NCHR07: LD HL,NATSM8 ;ASSUME MIDDLE OF STAFF
D6C1  78  C  LD A,B ;STAFF ORIENTATION
D6C2  B9  C  CP C
D6C3  20 03  C  JR NZ,NCHR08
D6C5  21 C510  C  LD HL,NATMT
D6CB  C1  C  NCHR08: POP BC
D6C9  C9  C  RET

;FLAT SIGN CHARACTERS
;DETERMINES WHICH FLAT SIGN CHARACTERS ARE TO BE USED
;DEPENDING UPON THE POSITION OF THE NOTE ON THE STAFF.
;GIVEN: (A) - ADJUSTED PITCH
;        (Y) - ADDRESS OF ACTIVE PARTITION TABLE
;RETURNS: (HL) - ADDRESS OF CHARACTER VECTOR

D6CA  C5  C  FCHARS: PUSH BC  ;SAVE PITCH
D6CB  47  C  LD B,A  ;SET DISPLACEMENT ONLY
D6CC  E6 0F  C  AND 0FH
D6CE  4F  C  LD C,A  ;AND KEEP IT IN (C)
D6CF  CB 68  C  BIT LINSPCB ;TEST LINE/SPACE BIT
D6D1  20 23  C  JR NZ,FCHR02 ;JUMP IF ON SPACE
D6D3  21 C51C  C  LD HL,FLTLAN ;ASSUME ABOVE/BELLOM STAFF
D6D6  FD 7E 04  C  LD A,(Y+STFORD) ;COMPARE WITH STAFF
D6D9  47  C  LD B,A  ; ORIENTATION
D6DA  3D  C  DEC A
D6DB  B9  C  CP C
D6DC  38 38  C  JR NC,FCHR85 ;JUMP IF ABOVE STAFF
D6DE  C6 06  C  ADD A,06H  ;ELSE CHECK BELOW STAFF
D6EE  B9  C  CP C
D6F1  38 33  C  JR C,FCHR85  ;JUMP IF BELOW STAFF
D6F3  20 05  C  JR NZ,FCHR01 ;JUMP IF MIDDLE STAFF
D6F5  21 C528  C  LD HL,FLTLBT ;BELOW WITH TOP LINE
D6F8  18 2C  C  JR FCHR85
MUSIC CONTROL PROGRAM HACRO-88
PITCH AND ACCIDENTAL DISPLAY

D6E4 21 C524 C FCHR81: LD HL,FLTML ±MIDDLE STAFF
D6ED 78 C LD A,B
D6E6 B9 C CP C
D6F1 20 25 C JR NZ,FCHR85 ±MIDDLE STAFF, NO TOP LINE
D6F4 18 20 C JR FCHR85
D6F6 21 C52B C FCHR82: LD HL,FLTMT ±ON SPACE ABOVE/BELLOW STAFF
D6F9 F7 84 C LD A,(iY+STFORM)
D6FC 47 C LD B,A
D6FD B9 C CP C
D6FE 38 87 C JR C,FCHR83 ±JUMP IF MID/BELLOW STAFF
D708 20 14 C JR NZ,FCHR85 ±JUMP IF ABOVE STAFF
D702 21 C538 C LD HL,FTSAT ±ABOVE STAFF WITH BOTTOM LINE
D705 18 8F C JR FCHR85
D707 C6 05 C FCHR83: ADD A,03H
D709 B9 C CP C
D70A 38 8A C JR C,FCHR85 ±JUMP IF BELOW STAFF
D70C 20 05 C JR NZ,FCHR84 ±JUMP IF MIDDLE STAFF
D70E 21 C534 C LD HL,FTSBT ±BELOW STAFF WITH TOP LINE
D711 18 03 C JR FCHR85
D713 21 C538 C FCHR84: LD HL,FLTSL ±MIDDLE STAFF WITH LINES
D716 C1 C FCHR85: POP BC
D718 C5 C SCHARS: PUSH BC
D719 47 C LD B,A ±SAVE PITCH
D71A E6 8F C AND @FH ±GET DISPLACEMENT ONLY
D71C 4F C LD C,A ±KEEP IT IN (C)
D71D CB 80 C BIT LINSPC,B ±TEST LINE/SPACE BIT
D71F 20 33 C JR NZ,SCHR84 ±JUMP IF ON SPACE
D721 21 C53C C LD HL,SHPLAN ±ON LINE ABOVE/BELLOW STAFF
D724 FD 7E 84 C LD A,(iY+STFORM) ±GET STAFF ORIENTATION
D727 47 C LD B,A ±AND COMPARE IT TOO
D728 30 C DEC A ±POSITION OF NOTE
D72B B9 C CP C
D72C 38 87 C JR C,SCHR81 ±JUMP IF NOT ABOVE STAFF
D72E 20 46 C JR NZ,SCHR87 ±JUMP IF ABOVE STAFF
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-8

PITCH AND ACCIDENTAL DISPLAY

D72E  21 C544  C  LD  HL,SHPLAL  ;ABOVE STAFF WITH BOTTOM LINE
D731  18  41  C  JR  SCHR07
D733  C6  06  C  SCHR01:  ADD  A, 6H  ;SEE IF BELOW STAFF
D735  B9  C  CP  C
D736  38  3C  C  JR  C,  SCHR07  ;JUMP IF BELOW STAFF
D738  20  05  C  JR  NZ,  SCHR02  ;JUMP IF MIDDLE STAFF
D73A  21  C54C  C  LD  HL,SHPLBL  ;BELOW STAFF WITH TOP LINE
D73D  18  35  C  JR  SCHR07
D73F  21  C554  C  SCHR02:  LD  HL,SHPLML  ;MIDDLE STAFF WITH LINES
D742  30  C  DEC  A
D743  B9  C  CP  C
D744  20  05  C  JR  NZ,  SCHR03  ;JUMP IF MIDDLE STAFF
D746  21  C55C  C  LD  HL,SHPLMB  ;MIDDLE STAFF WITH NO BOTTOM LINE
D749  18  29  C  JR  SCHR07
D74B  78  C  SCHR03:  LD  A,  6  ;STAFF
D74C  B9  C  CP  C
D74D  20  25  C  JR  NZ,  SCHR07  ;JUMP IF MIDDLE STAFF
D74F  21  C564  C  LD  HL,SHPLMT  ;MIDDLE STAFF WITH NO TOP LINE
D752  18  20  C  JR  SCHR07
D754  21  C56C  C  SCHR04:  LD  HL,SHPSAN  ;ON SPACE ABOVE/BELOW STAFF
D757  FD  7E  04  C  LD  A, (IY+STFORMD
D75A  47  C  LD  B,  A
D75B  B9  C  CP  C
D75C  38  87  C  JR  C,  SCHR05  ;JUMP IF NOT ABOVE STAFF
D75E  20  14  C  JR  NZ,  SCHR07  ;JUMP IF ABOVE STAFF
D760  21  C574  C  LD  HL,SHPSAL  ;ABOVE STAFF WITH BOTTOM LINE
D763  18  8F  C  JR  SCHR07
D765  C6  05  C  SCHR05:  ADD  A,  85H  ;POINT TO BOTTOM OF STAFF
D767  B9  C  CP  C
D768  38  04  C  JR  C,  SCHR07  ;JUMP IF BELOW STAFF
D76A  20  05  C  JR  NZ,  SCHR06  ;JUMP IF NOT BELOW STAFF
D76C  21  C57C  C  LD  HL,SHPSBL  ;BELOW STAFF WITH TOP LINE
D76F  18  83  C  JR  SCHR07
D771  21  C584  C  SCHR06:  LD  HL,SHPSML  ;MIDDLE STAFF WITH ALL LINES
D774  C1  C  SCHR07:  POP  BC
D775  C9  C  RET

C
C ;
C ; DSCHAR
C
C ; DOUBLE SHARP SIGN CHARACTERS
C ; DETERMINES WHICH DOUBLE SHARP SIGN CHARACTERS ARE TO BE
C ; USED DEPENDING UPON THE POSITION OF THE NOTE ON THE STAFF.
C ;
C ; GIVEN:  (A) - ADJUSTED PITCH
C ;  (IY) - ADDRESS OF ACTIVE PARTITION TABLE
C ;  RETURNS:  (HL) - ADDRESS OF CHARACTER VECTOR
C ;
PITCH AND ACCIDENTAL DISPLAY

D776 C5 C DSCHAR: PUSH BC
D777 47 C LD B,A ;SAVE PITCH
D778 E6 0F C AND 0FH ;GET DISPLACEMENT ONLY
D779 4F C LD C,A ;SAVE IT IN (C)
D77B CB 68 C BIT LINSPC,B ;TEST LINE/SPACE BIT
D77D 20 14 C JR NZ,DSCHB1 ;JUMP IF ON SPACE
D77F 21 C5BC C LD HL,DSHLAN ;ON LINE ABOVE OR BELOW STAFF
D782 FD 7E 04 C LD A,(IY+STFORD) ;GET STAFF ORIENTATION
D785 3D C DEC A
D786 B9 C CP C
D787 30 29 C JR NC,DSCH04 ;JUMP IF ABOVE STAFF
D789 C6 05 C ADD A,08H ;CHECK BELOW STAFF
D78B B9 C CP C
D78C 38 24 C JR C,DSCH04 ;JUMP IF BELOW STAFF
D78E 21 C5BE C LD HL,DSHML ;ELSE IN MID STAFF
D791 18 1F C JR DSCH04
D793 21 C590 C DSCH01: LD HL,DSHSAN ;ON SPACE ABOVE/BELOW STAFF
D796 FD 7E 04 C LD A,(IY+STFORD)
D799 B9 C CP C
D79A 38 07 C JR C,DSCH02 ;JUMP IF NOT ABOVE STAFF
D79C 20 14 C JR NZ,DSCH04 ;JUMP IF ABOVE STAFF
D79E 21 C594 C LD HL,DSHSAB ;ABOVE STAFF WITH BOTTOM LINE
D7A1 18 0F C JR DSCH04
D7A3 C6 05 C DSCH02: ADD A,08H ;CHECK BELOW STAFF
D7A5 B9 C CP C
D7A6 38 0A C JR C,DSCH04 ;JUMP IF BELOW STAFF
D7A8 20 05 C JR NZ,DSCH03 ;JUMP IF MID STAFF
D7AD 21 C598 C LD HL,DSHSL ;BELOW STAFF WITH TOP LINE
D7A0 18 03 C JR DSCH04
D7A5 21 C59C C DSCH03: LD HL,DSHSBL ;MIDDLE STAFF WITH LINES
D7A2 C1 C DSCH04: POP BC
D7A3 C9 C RET

;SUBTTL CONVERSION Routines
MACLIB B:TRP06.MAC

;TRNPOL

;TRANSPOSE LETTER/ACCIDENTAL
;TAKES LETTER NAME/ACCIDENTAL AND TRANSPOSES IT BY CONVERTING
;IT TO ITS SCALE DEGREE IN THE ORIGINAL KEY AND CONVERTING
;THAT SCALE DEGREE TO ITS LETTER NAME IN THE NEW KEY.

;GIVEN: (B) - LETTER NAME IN ASCII
;(C) - ACCIDENTAL (-2 TO +2)
;(D) - OCTAVE
CONVERSION ROUTINES

; RETURNS: (A) - COMPLETION CODE
; = 00H - SUCCESSFUL
; = FFH - INVALID DATA
; (B) - TRANPOSED LETTER NAME IN ASCII
; (C) - TRANPOSED ACCIDENTAL (-2 TO +2)
; (D) - TRANPOSED OCTAVE
; (E) - CHANGED

D7B4 E5 C TRNPL01: PUSH HL
D7B5 3A 0000H C LD A,(XPOSOD) ; GET OCTAVE/DIRECTION
D7B8 CB 7F C BIT 7,A ; TEST FOR UP/DOWN
D7BA 3A 0000H C LD A,(XPOSOD) ; GET CUT OFF LETTER
D7BD 26 67 C JR NZ,TRPL01 ; JUMP IF TRANSPOSE DOWN
D7BF 3D C DEC A ; SEE IF GIVEN LETTER IS
D7C0 B8 C CP B ; ABOVE THE CUT OFF LETTER
D7C1 30 67 C JR NC,TRPL02 ; JUMP IF IT IS NOT
D7C3 14 C INC D ; ELSE ADJUST THE OCTAVE
D7C4 18 64 C JR TRPL02 ; FOR THE WRAP-AROUND EFFECT
D7C6 B8 C TRPL01: CP B ; SEE IF BELOW CUT OFF LETTER
D7C7 38 01 C JR C,TRPL02 ; JUMP IF IT IS NOT
D7C9 15 C DEC D ; ELSE ADJUST THE OCTAVE
D7CA 3A 0000H C TRPL02: LD A,(XPOSOD) ; ADJUST OCTAVE IF THE TRANSPOSE
D7CD CB 7F C BIT 7,A ; IS MORE THAN 1 OCTAVE
D7CF 28 63 C JR Z,TRPL03 ; JUMP IF TRANSPOSE IS UP
D7D1 3C C INC A ; DECREMENT ABSOLUTE VALUE
D7D2 18 01 C JR TRPL04
D7D4 3D C TRPL03: DEC A
D7D5 82 C TRPL04: ADD A,D ; ADD THE OCTAVE DISPLACEMENT
D7D6 67 C LD H,A ; SAVE THE ADJUSTED OCTAVE
D7D7 CD D0AA C CALL KEYSNP ; TEMP CHANGE TO OLD KEY
D7DA CD D022 C CALL CVTSTS ; GET SCALE DEGREE IN OLD KEY
D7DD 7A C LD A,D ; TEST FOR INVALID DATA
D7DE A7 C AND A
D7DF 3E FF C LD A,0FFH ; FIRST ASSUME INVALID
D7E1 28 8C C JR Z,TRPL05 ; JUMP IF INVALID
D7E3 CD D0AA C CALL KEYSNP ; CHANGE BACK TO NEW KEY
D7E6 42 C LD B,D ; MOVE SCALE DEGREE AND MODIFIER
D7E7 4B C LD C,E ; TO (BC) FOR NEXT CALL
D7E8 CD D081 C CALL CVTSTS ; CONVERT THE SCALE DEGREE BACK
D7EB 42 C LD B,D ; TO THE LETTER NAME IN NEW KEY
D7EC 4B C LD C,E ; MOVE DATA TO APPROPRIATE
D7ED 54 C LD D,H ; REGISTERS FOR RETURN
D7EE AF C XOR A ; SET SUCCESSFUL COMPLETION
D7EF E1 C TRPL05: POP HL
D7F8 C9 C RET
C ;
C ;
C ; TRNPOL

}
TRANSPOSE UNADJUSTED PITCH IN OLD KEY TO THE UNADJUSTED
PITCH IN THE NEW KEY USING THE PARAMETERS SET BY `SETXPO'.
THE UNADJUSTED PITCH IS CONVERTED TO ITS LETTER NAME/
ACCIDENTAL.  THIS LETTER NAME IS TRANSPOSED BY CONVERTING
IT TO ITS SCALE DEGREE IN THE OLD KEY AND CONVERTING THE
SCALE DEGREE BACK TO ITS LETTER NAME USING THE NEW KEY.
THE NEW LETTER NAME IS THEN CONVERTED BACK TO ITS UNADJUSTED
PITCH FORMAT.

GIVEN:  (B) - UNADJUSTED PITCH
(C) - OCTAVE
RETURNS:  (B) - TRANSPosed UNADJUSTED PITCH
(C) - TRANSPosed OCTAVE

; TRANSPONE: PUSH DE
LD D,C
;SEND UNADJUSTED PITCH IN (A)
CALL CVTUTL ;CONVERT TO LETTER NAME
CALL TNPOL ;TRANSPOSE LETTER NAME
CALL CVTLTU ;CONVERT TO UNADJUSTED PITCH
LD B,A ;SEND UNADJUSTED BACK IN (B)
LD C,D ;SEND ADJUSTED OCTAVE IN (C)
POP DE
RET

; CVTSTL
; CONVERT SCALE DEGREE TO LETTER NAME/ACCIDENTAL.
; NOTE THAT IT IS POSSIBLE TO SUPPLY AN ALTERED SCALE DEGREE
; THAT CAUSES THE RETURN OF A TRIPLE SHARP OR FLAT.  THIS
; ROUTINE DOES NOT CHECK FOR SUCH ERRORS.

; GIVEN:  (B) - SCALE DEGREE (1 TO 7)
(C) - MODIFIER (-2 TO +2 HALF STEPS)
RETURNS:  (D) - LETTER NAME IN ASCII
(E) - ACCIDENTAL (-2 TO +2)

; CVTSTL: PUSH BC
LD DE,0000H ;SET ERROR INDICATOR
LD A,B ;GET SCALE DEGREE
DEC A ;MAKE RELATIVE TO 0
CP 07H ;CHECK MAX VALUE
JR NC,CVSL02 ;JUMP IF INVALID
LD A,(KEYLET) ;GET KEY LETTER
AND A ;TEST IF VALID
JR Z,CVSL02 ;JUMP IF NOT
CONVERSION ROUTINES

D811 80 C ADD A,B ;ADD SCALE DEGREE TO KEY LETTER
D812 30 C DEC A ;ADJUST BY 1
D813 FE 40 C CP 40H ;TEST FOR WRAP AROUND
D814 39 82 C JR C,CVSL01 ;JUMP IF IN RANGE
D815 D6 07 C SUB 07H ;ELSE WRAP AROUND
D816 47 C CVSL01: LD B,A ;SAVE LETTER NAME
D817 DE10 C CALL ACNKEY ;GET ACCIDENTAL IN KEY
D818 5F C LD E,A ;LOAD ACCIDENTAL
D819 50 C LD D,B ;LOAD LETTER NAME
D820 C1 C CVSL02: POP BC
D821 C9 C RET

; CONVERT LETTER/ACCIDENTAL TO SCALE DEGREE. NOTE THAT IT IS
; POSSIBLE THAT A SCALE DEGREE MAY BE ALTERED BY MORE THAN A
; WHOLE STEP. THIS ROUTINE RETURNS THE NUMBER OF HALF STEPS
; THAT THE LETTER IS ALTERED REGARDLESS.
; GIVEM: (B) - LETTER NAME IN ASCII
; (C) - ACCIDENTAL (-2 TO +2)
; RETURNS: (D) - SCALE DEGREE (1 TO 7)
; = 00H - ERROR FOUND
; (E) - MODIFIER (HALF STEP COUNT)
; = 00H - UNALTERED
; = 01H - RAISED 1 HALFW STEP, ETC
; = FFH - LOWERED 1 HALFW STEP, ETC

D822 E5 C WILTS: PUSH HL
D823 11 0000 C LD DE,0000H ;ERROR INDICATOR VALUES
D824 CD DEBC C CALL ALNCHK ;CHECK FOR VALIDITY
D825 FE FF C CP 0FFH ;TEST ERROR FLAG
D826 28 1E C JR Z,CVLS02 ;JUMP IF INVALID
D827 3A 0000X C LD A,(KEYLET) ;GET KEY LETTER
D828 A7 C AND A ;CHECK FOR VALIDITY
D829 28 19 C JR Z,CVLS02 ;JUMP IF INVALID
D830 57 C LD D,A ;STORE KEY LETTER
D831 78 C LD A,B ;GET GIVEN LETTER
D832 92 C SUB D ;GIVEN LETTER MINUS KEY LETTER
D833 30 02 C JR NC,CVLS01 ;JUMP IF GIVEN LETTER LARGER
D834 C6 07 C ADD A,07H ;ELSE MAKE POSITIVE
D835 3C C CVLS01: INC A ;SCALE DEGREE
D836 57 C LD D,A ;SAVE SCALE DEGREE
D837 CD DE10 C CALL ACNKEY ;GET ACCIDENTAL IN KEY FOR GIVEN LETTER
D838 59 C LD E,C ;IF LETTER IS NOT ALTERED, THEN THE
D839 FE 00 C CP 00H ;ACCIDENTAL IS ALSO THE ALTERATION
CONVERSION ROUTINES

D 842 28 07  C  JR Z,CVLS82 ; JUMP IF LETTER IS NOT ALTERED
D 844 1C  C  INC E ; IF LETTER IS FLAT IN KEY, THEN THE
D 845 FF FF  C  CP 0FFH ; ALTERATION IS THE ACCIDENTAL + 1
D 847 28 02  C  JR Z,CVLS82 ; JUMP IF THE LETTER IS FLATTED
D 849 1D  C  DEC E ; ELSE IT IS SHARPED IN KEY, SO
D 84A 1D  C  DEC E ; ALTERATION IS ACCIDENTAL - 1
D 84B E1  C  CVLS82: POP HL
D 84C C9  C  RET

; CVLTU
;
; CONVERT LETTER NAME AND ACCIDENTAL TO UNADJUSTED PITCH.
;
; GIVEN: (B) - LETTER IN ASCII
; (C) - ACCIDENTAL (-2 TO +2)
; RETURNS: (A) - UNADJUSTED PITCH
; = FFH IF DATA IS INVALID
;
D 84D D5  C  CVLTU: PUSH DE
D 84E E3  C  PUSH HL
D 84F CD DEBC  C  CALL ALNCRI ; CHECK DATA FOR ERRORS
D 852 FF FF  C  CP 0FFH ; CHECK ERROR FLAG
D 854 28 13  C  JR Z,CVLU82 ; JUMP IF INVALID
D 856 78  C  LD A,B ; GET LETTER
D 857 D6 41  C  SUB 41H ; CONVERT TO VALUE 0 - 6
D 859 CB 27  C  SLA A ; MULTIPLY BY 2 FOR INDEX
D 85B 21 C31A  C  LD HL,LTRTBL ; LETTER TRANSLATION TABLE
D 85D 85  C  ADD A,L ; USE LETTER AS INDEX
D 85F 4F  C  LD L,A ; INTO TRANSLATION TABLE
D 861 30 01  C  JR NC,CVLU01 ; CHECK FOR OVERFLOW
D 862 24  C  INC H ; ADJUST FOR OVERFLOW
D 863 7E  C  CVLU01: LD A,(HL) ; GET UNADJUSTED PITCH
D 864 57  C  LD D,A ; SAVE IT
D 865 CD D77A  C  CALL CVT8T3 ; CONVERT GIVEN ACCIDENTAL
D 868 B2  C  OR D ; PUT IT WITH LETTER
D 869 E1  C  CVLU82: POP HL
D 86A D1  C  POP DE
D 86B C9  C  RET

; CVLTU
;
; CONVERT LETTER NAME/ACCIDENTAL TO ABSOLUTE PITCH
;
; CONVERT GIVEN ACCIDENTAL (PLUS/_MINUS FORMAT) AND ASCII LETTER
; NAME AND OCTAVE TO ABSOLUTE PITCH FORMAT: HIGH ORDER 4 BITS
; ARE THE OCTAVE IN THE RANGE -8 TO +7; THE LOW ORDER 4 BITS
; ARE THE PITCH WITHIN THAT OCTAVE IN THE RANGE 0 TO 11. A = 0,
C ; AM/B-FLAT = 1, B = 2, ETC.
C
C ; GIVEN: (B) - ASCII LETTER NAME
C ; (C) - ACCIDENTAL (-2 TO +2)
C ; (D) - OCTAVE (-8 TO +7)
C ; RETURNS: (A) - ABSOLUTE PITCH
C
C
D86C D5 C COTLTA: PUSH DE
D86D E5 C PUSH HL
D86E CD DEBC C CALL ALNCIK ; CHECK VALUES FOR VALIDITY
D871 FE FF C CP OFFH ; CHECK VALIDITY
D873 28 27 C JR Z,CVLA04 ; JUMP IF INVALID
D875 78 C LD A,B ; GET LETTER NAME
D876 D6 41 C SUB 41H ; CONVERT TO 0 - 6
D878 CB 27 C SLA A ; MULTIPLY BY 2 FOR INDEX
D87A 21 C31A C LD HL,LTRTBL ; POINT TO LETTER TRANSLATION TABLE
D87D 85 C ADD A,L ; USE LETTER NAME AS INDEX
D87E 6F C LD L,A
D87F 30 01 C JR NC,CVLA01 ; TEST FOR OVERFLOW
D881 24 C INC H ; ADJUST FOR OVERFLOW
D882 23 C CVLA01: INC HL ; POINT TO ABSOLUTE VALUE
D883 7E C LD A,(HL) ; GET ABSOLUTE VALUE FOR WHITE NOTE
D884 81 C ADD A,C ; ADD ACCIDENTAL
D885 CB 7F C BIT 7,A ; CHECK FOR A-FLAT AND DOUBLE FLAT
D887 28 03 C JR Z,CVLA02 ; JUMP IF NOT
D889 15 C DEC D ; ELSE ADJUST OCTAVE FOR WRAP AROUND
D88A C4 0C C ADD A,0CH ; CONVERT TO MOD 12
D88C FE 0C C CVLA02: CP 0CH ; CHECK FOR G DOUBLE SHARP OR ERROR
D88E 38 03 C JR C,CVLA03 ; JUMP IF IN RANGE
D88F 14 C INC D ; ADJUST OCTAVE FOR WRAP AROUND
D891 D6 0C C SUB 0CH ; ELSE CONVERT TO MOD 12
D893 CB 22 C CVLA03: SLA D ; MOVE OCTAVE TO HIGH ORDER
D895 CB 22 C SLA D ; FOUR BITS
D897 CB 22 C SLA D
D899 CB 22 C SLA D
D89B B2 C OR D ; PUT OCTAVE BY ABSOLUTE PITCH
D89C E1 C CVLA04: POP HL
D89D D1 C POP DE
D89E C9 C RET
C
C
C ; CUTLTL
C ; CONVERT UNADJUSTED PITCH TO LETTER NAME/ACCIDENTAL
C
C ; GIVEN: (A) - UNADJUSTED PITCH
C ; RETURNS: (A) - UNCHANGED
CONVERSION ROUTINES

C ; (B) - LETTER NAME IN ASCII
C ; (C) - ACCIDENTAL (-2 TO +2)
C

D89F F5 C CVTUTL: PUSH AF
D8A0 4F C LD C,A ; SAVE UNADJUSTED PITCH
D8A1 E6 07 C AND B7H ; STRIP ALL BUT DISPLACEMENT
D8A3 D6 86 C SUB B6H ; CONVERT TO 0 TO -3
D8A5 3A 02 C JR NC,CVUL01 ; JUMP IF RESULT NOT NEGATIVE
D8A7 ED 44 C NEG ; ELSE MAKE POSITIVE
D8A9 CB 27 C CVUL01: SLA A ; MULTIPLY BY 2
D8AB CB 61 C BIT LINSPC,C ; TEST LINE/SPACE BIT
D8AD 28 01 C JR Z,CVUL02 ; JUMP IF ON LINE
D8AF 3C C INC A ; ELSE POINT TO NEXT LETTER
D8B0 3C C CVUL02: INC A ; CONVERT TO 1 - 7
D8B1 F6 40 C OR B4H ; CONVERT TO ASCII LETTER
D8B3 47 C LD B,A ; SAVE ASCII LETTER
D8B4 79 C LD A,C ; RESTORE UNADJUSTED PITCH
D8B5 CD D9A8 C CALL CVT3T8 ; CONVERT ACCIDENTAL
D8B8 4F C LD C,A ; RETURN ACCIDENTAL IN (C)
D8BA C9 C POP AF

C ;
C ; CVTUTA
C ; CONVERT UNADJUSTED PITCH TO ABSOLUTE PITCH.
C ;
C ; GIVEN: (B) - UNADJUSTED PITCH
C ; (C) - OCTAVE
C ; RETURNS: (A) - ABSOLUTE PITCH
C ;

D8BB C5 C CVTUTA: PUSH BC
D8BC E5 C PUSH HL
D8BD 78 C LD A,B ; GET UNADJUSTED PITCH
D8BE E6 07 C AND B7H ; GET DISPLACEMENT ONLY
D8C0 D6 86 C SUB B6H ; CONVERT TO 0 TO -3
D8C2 3A 02 C JR NC,CVUA01 ; JUMP IF NOT NEGATIVE
D8C4 ED 44 C NEG ; ELSE MAKE IT POSITIVE 0 TO 3
D8C6 CB 27 C CVUA01: SLA A ; EVEN NUMBERS FROM 0 TO 6
D8C8 CB 68 C BIT LINSPC,B ; TEST LINE/SPACE BIT
D8CA 28 01 C JR Z,CVUA02 ; JUMP IF ON LINE
D8CC 3C C INC A ; ELSE GET ODD NUMBERS 1 TO 5
D8CD CB 27 C CVUA02: SLA A ; RESULT IS INDEX INTO LETTER
D8CF 21 C31A C LD HL,LTRTL ; TRANSLATION TABLE WITH 2
D8D2 85 C ADD A,B,L ; BYTES PER ENTRY, THE SECOND
D8D3 6F C LD L,A ; OF WHICH IS THE ABSOLUTE PITCH
D8D4 3A 01 C JR NC,CVUA03 ; JUMP IF NO OVERFLOW
D8D6 24 C INC H ; ELSE ADJUST FOR OVERFLOW
CONVERSION ROUTINES

D807 23 C CVUA03: INC HL
D808 78 C LD A,B 
D809 CD D9A8 C CALL CVT3T8 
D80C B6 C ADD A,(HL) 
D80D CB 7F C BIT 7,A 
D80F 28 05 C JR Z,CVUA04 
D81 C6 0C C ADD A,0CH 
D813 0D C DEC C 
D814 18 07 C JR CVUA05 
D816 FE 0C C CVUA04: CP 0CH 
D818 38 03 C JR C,CVUA05 
D819 B6 0C C SUB 0CH 
D81C 0C C INC C 
D81E CB 21 C CVUA05: SLA C 
D81F CB 21 C SLA C 
D821 CB 21 C SLA C 
D823 B1 C OR C 
D824 E1 C POP HL 
D82 C1 C F3 5C 
D828 C9 C RET 
D829 E5 C CVTATU: PUSH HL 
D82A 78 C LD A,B 
D82B 4F C LD C,A 
D82C CB 29 C SRA C 
D82E CB 29 C SRA C; value an 8-bit value 
D830 CB 29 C SRA C 
D832 CB 29 C SRA C 
D834 E6 0F C AND 0FH 
D836 21 E000 C LD HL,UNATBL; point to unadjusted pitch table 
D838 85 C ADD A,L 
D83A 6F C LD L,A 
D83B 38 01 C JR NC,CVUA01; check for page boundary cross 
D83D 24 C INC H 

CONVERT ABSOLUTE PITCH TO UNADJUSTED PITCH USING THE UNADJUSTED PITCH TABLE (UNATBL). IF THE ABSOLUTE PITCH IS AT EITHER END OF THE TABLE, THE OCTAVE MAY HAVE TO BE ADJUSTED DEPENDING UPON HOW THE PITCH IS SPelled. 

GIVEN: (B) - ABSOLUTE PITCH 

RETURNS: (A) - ABSOLUTE PITCH 
(C) - UNADJUSTED PITCH 
(C) - OCTAVE 

GET DISPLACEMENT ONLY 
POINT TO UNADJUSTED PITCH TABLE 
USE ABSOLUTE PITCH AS INDEX 
INTO TABLE 
CHECK FOR PAGE BOUNDARY CROSS 
ADJUST FOR CROSSING
CONVERSION ROUTINES

D90E 78 C CVAU01: LD A,B ;RESTORE ABSOLUTE PITCH
D90F 46 C LD B,(HL) ;GET UNADJUSTED PITCH
D910 6F C LD L,A ;SAVE ABSOLUTE PITCH
D911 E6 0F C AND 0FH ;GET DISPLACEMENT VALUE (0 TO B)
D913 FE 00 C CP 00H ;TEST FOR A/G DOUBLE-SHARP
D915 20 0A C JR NZ,CVAU02 ;GO TO OTHER TEST
D917 78 C LD A,B ;RESTORE UNADJUSTED PITCH
D918 E6 07 C AND 07H ;GET LETTER PORTION ONLY
D91A FE 03 C CP 03H ;TEST FOR NOTE A
D91C 20 8F C JR NZ,CVAU03 ;EXIT IF NOT LETTER A
D91E 0C C DEC C ;ELSE ADJUST THE OCTAVE FOR A
D920 FE 00 C CVAU03: CP @AH ;TEST FOR G/A-FLAT; G/A-DB FLAT
D922 38 0B C JR C,CVAU03 ;JUMP IF NOT LETTER G OR A
D924 78 C LD A,B ;GET UNADJUSTED PITCH
D926 E6 07 C AND 07H ;GET LETTER PORTION ONLY
D928 FE 06 C CP 06H ;TEST FOR LETTER A
D92A 20 81 C JR NZ,CVAU03 ;EXIT IF LETTER G OR F
D92C 0C C INC C ;ELSE ADJUST OCTAVE FOR A
D92E 70 C CVAU03: LD A,L ;RELOAD ABSOLUTE PITCH

; CONVERT ABSOLUTE PITCH TO AN INTEGER REPRESENTING THE NOTE ON THE STANDARD KEYBOARD. THIS INTEGER IS USED AS AN INDEX INTO THE INCREMENT TABLE FOR SOUND OUTPUT. THE OCTAVE OF THE ABSOLUTE PITCH IS USED AS AN INDEX INTO THE ABSOLUTE PITCH TRANSLATION TABLE. THE VALUE READ IS THE BASE VALUE FOR THAT OCTAVE.

; GIVEN: (B) - ABSOLUTE PITCH
; RETURNS: (A) - INTEGER (1 TO 840)
; = FF - OUT OF RANGE

D930 C5 C CVTATI: PUSH BC
D931 E5 C PUSH HL
D932 78 C LD A,B
D933 C6 30 C ADD A,30H ;USE OCTAVE AS INDEX INTO
D935 1F C RRA ;ABSPTT TABLE BY ADJUSTING
D936 1F C RRA ;OCTAVES 0 TO 3 TO THE
D937 1F C RRA ;VALUES 0 TO 6 IN THE
D938 1F C RRA ;LOW ORDER 4 BITS
D939 E6 0F C AND 0FH ;TEST FOR OUT OF RANGE
D93A FE 07 C CP 07H ;JUMP IF NOT OK
D93B 38 10 C JR NC,CVAU02
CONVERSION ROUTINES

```
D93F 21 C0C5 C    LD   HL,ABSPTT ;GET BASE ADDRESS OF TABLE
D942 85 C    ADD   A,L ;ADD INDEX VALUE
D943 6F C    LD   L,A
D944 30 01 C    JR   NC,CVA101 ;JUMP IF NO OVERFLOW
D946 24 C    INC   H ;ADJUST FOR OVERFLOW
D947 4E C    CVA101:  LD   C,(HL) ;GET BASE VALUE
D948 78 C    LD   A,B ;GET ABSOLUTE PITCH
D949 E6 0F C    AND   0FH ;GET LOW ORDER 4 BITS ONLY
D94A FE 8C C    CP   0CH ;TEST FOR MAXIMUM VALUE
D94B 38 04 C    JR   C,CVA103 ;JUMP IF VALUE IS OK
D94D 3E FF C    CVA103:  ADD  A,C ;ADD IT TO BASE VALUE
D94F 18 01 C    CVA104:  POP   BC
D950 C9 C    POP   HL
D951 C7 C
D952 CD D8F9 C    CALL   CVTATU ;CONVERT ABSOLUTE TO UNADJUSTED
D953 78 C    CVTATU:  LD   A,B ;GET INTEGER PITCH
D954 3D C    DEC   A ;MAKE RELATIVE TO 0 FOR TEST
D955 FE 54 C    CP   54H ;TEST FOR OUT OF RANGE
D956 38 04 C    JR   C,CVIU01 ;JUMP IF OK
D958 3E FF C    CVIU01:  LD   A,0FFH ;LOAD OUT OF RANGE FLAG
D95A 18 18 C    JR   CVIU04 ;EXIT WITH ERROR FLAG
D95C 0E FD C    CVIU04:  ADD  A,C ;ADD IT TO BASE VALUE
D95E CB 21 C    CVIU03:  SLA   C ;PUT OCTAVE IN HIGH
D95F CB 21 C    SLA   C ;ORDER 4 BITS
D960 CB 21 C    SLA   C ;PUT OCTAVE BESIDE PITCH
D961 47 C    LD   B,A ;SEND TO CONVERT ROUTINE
D962 CD D8F9 C    CALL   CVTATU ;CONVERT ABSOLUTE TO UNADJUSTED
```

CONVERT INTEGER PITCH TO UNADJUSTED AND ABSOLUTE PITCH.
THE LARGEST INTEGER PITCH ALLOWED IS 84 DECIMAL.

GIVEN:  (B) - INTEGER PITCH
RETURNS:  (A) - ABSOLUTE PITCH
= FFH - INVALID PITCH
(B) - UNADJUSTED PITCH
(C) - OCTAVE
CONVERSION ROUTINES

D979 C9 C CVIU04: RET

; CONVERT PLUS-MINUS (8-BIT) ACCIDENTAL TO 3-BIT CODE IN HIGH ORDER THREE BITS AS USED IN ADJUSTED/UNADJUSTED PITCH. THE LETTER NAME IS USED TO DETERMINE WHETHER THE PITCH IS DIATONIC OR NOT, SINCE THE 3-BIT ACCIDENTAL CODE DEPENDS UPON THIS FACT.

; GIVEN: (B) - LETTER IN ASCII
; (C) - ACCIDENTAL (-2 TO +2)

; RETURNS: (A) - ACCIDENTAL IN (UN)ADJUSTED PITCH FORMAT
; (ONLY HIGH ORDER 3 BITS)

D97A D5 C CVT8T3: PUSH DE

D978 CD DE10 C CALL ACONKY ; IS ACCIDENTAL IN KEY SIGNATURE

D97E B9 C CP C ; COMPARE ACCIDENTALS

D97F 20 10 C JR NZ,CV8301 ; JUMP IF NOT IN KEY SIGNATURE

D981 79 C LD A,C ; GET GIVEN ACCIDENTAL

D982 1E 80 C LD E,00H ; START ACCIDENTAL WITH NATURAL

D984 A7 C AND A ; IS GIVEN ACCIDENTAL A NATURAL

D985 28 1E C JR 2,CV8303 ; JUMP IF IT IS

D987 1E 20 C LD E,20H ; NOW ASSUME FLAT

D989 CB 7F C BIT 7,A ; NEGATIVE ACCIDENTAL IS FLAT

D98B 20 18 C JR NZ,CV8303 ; JUMP IF FLAT

D98D 1E 40 C LD E,40H ; ASSUME SHARP

D98F 18 14 C JR CV8303 ; EXIT

D991 79 C CV8301: LD A,C ; GET GIVEN ACCIDENTAL

D992 1E 60 C LD E,60H ; NATURAL NOT IN KEY SIGNATURE

D994 A7 C AND A ; TEST GIVEN FOR NATURAL

D995 28 0E C JR 2,CV8303 ; JUMP IF NATURAL

D997 1E 80 C LD E,80H ; ASSUME SINGLE FLAT

D999 CB 47 C BIT 0,A ; TEST FOR SINGLE/DUPLICATE ACCIDENTAL

D99B 20 82 C JR NZ,CV8302 ; JUMP IF SINGLE

D99D CB F3 C SET DOUBLE,E ; ELSE SET DOUBLE BIT

D99F CB 7F C CV8302: BIT 7,A ; TEST FOR FLAT OR SHARP

DA01 20 82 C JR NZ,CV8303 ; JUMP IF FLAT

DA03 CB EB C SET FLTPSP,E ; ELSE SET SHARP BIT

DA05 7B C CV8383: LD A,E ; LOAD ACCIDENTAL

DA06 D1 C POP DE

DA07 C9 C RET

; TAKES THE 3-BIT CODE FOR THE ACCIDENTAL AS USED IN ADJUSTED AND UNADJUSTED PITCH AND CONVERTS IT TO THE RELATIVE VALUE
C ; 8-BIT ACCIDENTAL IN THE RANGE -2 TO +2.
C
C ; GIVEN: (A) - ACCIDENTAL CODE IN HIGH ORDER 3 BITS
C ; RETURNS: (A) - 8-BIT RELATIVE ACCIDENTAL (-2 TO +2)
C
D9A8 C5 C ; CVT3T8: PUSH BC
D9A9 E6 E8 C AND @E8H ; STRIP ALL BUT ACCIDENTAL
D9A8 FE 60 C CP 68H ; TEST FOR DIATOMIC
D9AD 30 02 C JR NC,CV3881 ; JUMP IF NON-DIATOMIC
D9AF C6 60 C ADD A,68H ; ELSE MAKE NON-DIATOMIC
D9B1 47 C CV3881: LD B,A ; STORE ACCIDENTAL
D9B2 AF C XOR A ; START ACCIDENTAL AT NATURAL
D9B3 CB 78 C BIT 7,B ; TEST FOR NATURAL
D9B5 28 0C C JR Z,CV3803 ; JUMP IF IT IS
D9B7 3C C INC A ; ASSUME SINGLE SHARP/FLAT
D9B8 CB 70 C BIT DOUBLE,B ; TEST DOUBLE SHARP/FLAT BIT
D9BA 28 01 C JR Z,CV3802 ; JUMP IF SINGLE SHARP/FLAT
D9BC 3C C INC A ; MAKE DOUBLE ACCIDENTAL
D9BD CB 68 C CV3802: BIT FLTSHP,B ; TEST FLAT/SHARP BIT
D9BF 28 02 C JR NZ,CV3803 ; JUMP IF SHARP
D9C1 ED 44 C NEG ; MAKE FLAT ACCIDENTAL
D9C3 C1 C CV3803: POP BC
D9C4 C9 C ; RET
D ; CVTRHY
C ; CONVERT RHYTHM BYTE
C ; TAKES RHYTHM INDICATOR AND CONVERTS IT INTO THE TUNE TABLE
C ; FORMAT. IF THE INDICATOR IS EQUAL TO ZERO, A BLACK WHOLE
C ; NOTE IS USED.
C
C ; GIVEN: (A) - RHYTHM INDICATOR
C ; RETURNS: (A) - RHYTHM BYTE IN TUNE TABLE FORMAT
C
D9C5 E5 C ; CVTRHY: PUSH HL
D9C6 A7 C AND A ; TEST FOR ZERO
D9C7 28 04 C JR N2,CVRH01 ; JUMP IF NOT BLACK WHOLE
D9C9 3E 07 C LD A,07H ; ELSE LOAD BLACK WHOLE NOTE
D9CB 18 02 C JR CVRH02 ; EXIT
D9CD 3E 01 C CVRH01: LD A,01H ; MAKE EVERYTHING A WHITE WHOLE
D9CF E1 C CVRH02: POP HL
D9D0 C9 C ; RET
D ; CVTRTR
C ; CONVERT RHYTHM BYTE IN TUNE TABLE FORMAT TO RHYTHM IN
C
C ; PLAYABLE PITCH DURATION. FINAL VALUE IS TOTAL NUMBER
C ; OF 32ND NOTE VALUES.
C ;
C ; GIVEN: (A) - RHYTHM IN TUNE TABLE FORMAT
C ;
C ; RETURNS: (A) - RHYTHM IN 'SNGTUN' FORMAT
C ;
D9D1  C5  C  CURTR: PUSH BC
D9D2  E5  C  PUSH HL
D9D3  21  C0CD  C  LD HL,RHYTBL ;POINT TO START OF TABLE
D9D6  47  C  LD B,A ;SAVE RHYTHM
D9D7  E6  07  C  AND 87H ;GET NOTE VALUE ONLY
D9D9  85  C  ADD A,L ;USE VALUE AS INDEX INTO
D9DA  6F  C  LD L,A ; THE RHYTHM TRANSLATION
D9DB  30  01  C  JR NC,CVRR01 ;JUMP IF NO OVERFLOW
D9DD  24  C  INC H ;ELSE ADJUST FOR OVERFLOW
D9DE  7E  C  CVRR01: LD A,(HL) ;GET NUMBER OF 32ND NOTES
D9DF  4F  C  LD C,A ;STORE IT IN (C)
D9E2  28  0A  C  JR Z,CVRR02 ;JUMP IF NO DOT
D9E4  CB  39  C  SRL C ;ELSE ADD HALF THE VALUE
D9E6  81  C  ADD A,C ; TO THE ORIGINAL VALUE
D9E7  CB  60  C  BIT DBLDOT,B ;TEST FOR DOUBLE DOT
D9E9  28  03  C  JR Z,CVRR02 ;JUMP IF NO DOT
D9EB  CB  39  C  SRL C ;ELSE ADD HALF THE PREVIOUS
D9ED  81  C  ADD A,C ; VALUE TO THE ORIGINAL
D9EE  E1  C  CVRR02: POP HL
D9EF  C1  C  POP BC
D9F0  C9  C  RET
C ;
C ; SUBTTL SOUND OUTPUT
C  MAelib b:McP07.mac
C ;
C ; MONOPH
C ;
C ; MONOPHONIC
C ; TRANSLATE TUNE TABLE INTO FORMAT USED BY 'SNGTUN' AND
C ; SING IT WITH ALL 4 VOICES IN UNISON. NOTE THAT (IX)
C ; IS RESET TO THE BEGINNING OF THE TUNE AND THE TUNE MUST
C ; TERMINATE WITH A DOUBLE BAR.
C ;
C ; GIVEN: (B) - TEMPO
C ;
C ; RETURNS: (A) - COMPLETION CODE
C ;
C = 00 - SUCCESSFUL
C ;
C = FF - ERROR FOUND; TUNE NOT PLAYED
C ;
C (IX) - POINTER TO NOTE IN ERROR
SOUND OUTPUT

`MONOPH: PUSH BC`

`D9F1  C5  C    MONOPH: PUSH BC`

`D9F2  D5  C    PUS DE`

`D9F3  E5  C    PUSH HL`

`D9F4  78  C    LD A,B`

`D9F5  32 0000x C    LD (TEMPO),A`

`D9F6  21 0000x C    LD HL,MRKSPC`

`D9F7  DD 21 0000x C    LD IX,TUNBL`

`D9FF  11 0004 C    LD DE,TTELEN`

`DA02  DD 7E 00 C    MON01: LD A,(IX+UNAPIT) ;GET UNADJUSTED PITCH`

`DA05  FE FE C    CP TTSBAR ;TEST FOR SINGLE BAR LINE`

`DA07  28 22 C    JR Z,MON04 ;JUMP TO SKIP BAR LINE`

`DA09  FE FF C    CP TTDBAR ;TEST FOR DOUBLE BAR`

`DA0B  28 22 C    JR Z,MON05 ;STORE IF END OF TUNE`

`DA0D  47 C    LD B,A`

`DA0E  DD 4E 01 C    LD C,(IX+OCTAVE) ;GET OCTAVE`

`DA11  CD 08BB C    CALL CVTUTA ;CONVERT TO ABSOLUTE PITCH`

`DA14  47 C    LD B,A ;PASS ABSOLUTE PITCH IN (B)`

`DA15  CD D930 C    CALL CVTATI ;CONVERT TO PLAYABLE INTEGER`

`DA18  FE 54 C    CP 54H ;TEST FOR OUT OF RANGE`

`DA1A  38 01 C    JR C,MON02 ;JUMP IF IN RANGE`

`DA1C  AF C    XOR A ;ELSE STORE A REST`

`DA1D  06 04 C    MON02: LD B,04H ;NUMBER OF VOICES`

`DA1F  77 C    MON03: LD (HL),A ;PUT PITCH IN ALL 4 VOICES`

`DA28  23 C    INC HL`

`DA2A  10 FC C    DJNZ MON03`

`DA2B  DD 7E 02 C    LD A,(IX+RHYBYT) ;GET RHYTHM BYTE`

`DA2C  CD D901 C    CALL CVTRTR ;CONVERT TO PLAYABLE RHYTHM`

`DA29  77 C    LD (HL),A ;STORE IT`

`DA2A  23 C    INC HL ;POINT TO NEXT NOTE GROUP`

`DA2B  DD 19 C    MON04: ADD IX,DE ;POINT TO NEXT NOTE`

`DA2D  18 D3 C    JR MON01`

`DA2F  06 05 C    MON05: LD B,05H ;STORE END OF TUNE MARKER`

`DA31  AF C    XOR A ; OF 5 ZEROS`

`DA32  77 C    MQ06: LD (HL),A`

`DA33  23 C    INC HL`

`DA34  10 FC C    DJNZ MQ06`

`DA36  11 0000x C    LD DE,MRKSPC ;POINT TO START OF TUNE`

`DA39  3A 0000x C    LD A,(TEMPO) ;GET TEMPO`

`DA3C  47 C    LD B,A ;PASS IT IN (B)`

`DA3D  CD DADC C    CALL SNTUN ;SING THE TUNE`

`DA4B  DD 21 0000x C    LD IX,TUNBL ;POINT TO TOP OF TUNE TABLE`

`DA44  E1 C    POP HL`

`DA45  D1 C    POP DE`

`DA46  C1 C    POP BC`

`DA47  C9 C    RET`

`C ;`

`C ; PlanKey`
C ;
C ; PLAY KEY
C ; PLAYS THE SCALE, UP AND DOWN, OF THE CURRENT KEY AND THE
C ; CHORDS I, IV, V7, I, TO AURALLY ESTABLISH THE KEY FOR THE
C ; USER.
C ;
C ; GIVEN: NONE
C ; RETURNS: NONE
C ;
DA48  C5 C  PLAKEY: PUSH BC
DA49  D5 C  PUSH DE
DA4A  E5 C  PUSH HL
DA4B  21 0000X C  LD HL,KEYLET ;POINT TO CURRENT KEY
DA4C  3A 0000X C  LD A,(XPOSHS) ;CHECK FOR ACTIVE TRANSPOSITION
DA51  A7 C  AND A ;ZERO ASSUMED TO MEAN NONE
DA52  20 03 C  JR Z,PLKY01 ;JUMP IF NO TRANSPOSITION
DA54  21 0000X C  LD HL,OLDKYL ;ELSE POINT TO OLD KEY
DA57  46 C  PLKY01: LD B,(HL) ;GET KEY LETTER
DA58  23 C  INC HL ;POINT TO ACCIDENTAL
DA59  4E C  LD C,(HL) ;GET ACCIDENTAL OF KEY LETTER
DA5A  16 FF C  LD D,OFFH ;PUT BASS NOTE IN -1 OCTAVE
DA5C  CD D930 C  CALL CVTLTA ;CONVERT LETTER TO ABSOLUTE PITCH
DA5F  FE FF C  CP OFFH ;TEST FOR ERROR
DA61  28 3A C  JR Z,PLKY07 ;EXIT IF ERROR
DA63  47 C  LD B,A ;PASS ABSOLUTE PITCH IN (B)
DA64  CD D930 C  CALL CVTATI ;CONVERT ABSOLUTE TO INTEGER
DA67  47 C  LD B,A ;KEEP KEY/ROOT NOTE
DA68  3A 0000X C  LD A,(OFFSET) ;GET OUTPUT INTEGER OFFSET
DA6B  4F C  LD C,A ;SAVE IT
DA6C  88 C  ADD A,B ;ADD IT TO ROOT NOTE
DA6D  FE 14 C  PLAKEY03: CP 14H ;ADJUST PITCH BY AN OCTAVE
DA6F  38 04 C  JR NC,PLKY04 ;Either UP or DOWN UNTIL
DA71  C6 0C C  ADD A,0CH ;IT IS NOT TOO HIGH OR
DA73  18 F8 C  JR PLKY03 ;LOW
DA75  FE 28 C  PLAKEY04: CP 28H ;KEEP IT BETWEEN GREAT E AND
DA77  38 04 C  JR C,PLKY05 ;SMALL E-FLAT
DA79  D6 0C C  SUB 0CH
DA7B  18 F8 C  JR PLKY04
DA7D  91 C  PLAKEY05: SUB C ;TAKE OFF OFFSET BEFORE CALL
DA7E  47 C  LD B,A
DA7F  3A 0000X C  LD A,(KEYMOD) ;GET MODE OF KEY
DA82  8E 78 C  LD C,78H ;TEMPO OF SCALE
DA84  11 CB03 C  LD DE,MJSCAL ;FIRST ASSUME MAJOR SCALE
DA87  21 CB98 C  LD HL,MJCHRD ;AND MAJOR CHORDS
DA8A  A7 C  AND A ;NOW TEST FOR MINOR KEY
DA8B  28 06 C  JR Z,PLKY06 ;JUMP IF MAJOR KEY
DA8D  11 CB4F C  LD DE,MNSCAL ;ELSE LOAD ADDRESS OF MINOR
DA90  21 CB08 C  LD HL,MNSCHRD ;SCALE AND CHORDS
SOUND OUTPUT

PLKY06: PUSH HL ;SAVE CHORDS FOR LAST
CALL PLAREL ;PLAY THE SCALE
POP DE ;RESTORE CHORD ADDRESS
LD C,78H ;SET TEMPO FOR CHORDS
CALL PLAREL ;PLAY THE CHORDS
PLKY07: POP HL
POP DE
POP BC
RET

PLAY RELATIVE
PLAYS A TUNE THAT IS IN A LOWEST-NOTE RELATIVE FORMAT. EACH NOTE IS A VALUE TO BE ADDED TO THE GIVEN ROOT NOTE. THE TUNE IS TRANSLATED INTO THE ABSOLUTE FORMAT AND THEN GIVEN TO 'SINGTUN' TO PLAY. THE NUMBER OF NOTE GROUPS MUST BE SUPPLIED IN THE FIRST BYTE OF THE TUNE.

GIVEN:
(B) - ROOT NOTE
(C) - TEMPO
(DE) - ADDRESS OF BEGINNING OF TUNE; THE FIRST BYTE MUST BE THE NUMBER OF NOTE GROUPS TO BE PLAYED

RETURNS: NONE

PLAREL: PUSH BC
PUSH DE
PUSH HL
LD HL,WKSPC ;USE WORKSPACE TO STORE TUNE
LD A,C ;GET TEMPO
LD (TEMPO),A ;GET ROOT NOTE
LD A,(DE) ;FIRST BYTE IS NUMBER OF NOTES
INC DE ;POINT TO START OF TUNE
LD (TEMP01),A ;SAVE NUMBER OF NOTES
PLRL01: LD B,04H ;NUMBER OF VOICES
LD A,(DE) ;GET NOTE FOR NEXT VOICE
ADD A,C ;ADD IT TO ROOT NOTE
LD (HL),A ;STORE IT IN TUNE SPACE
INC HL ;NEXT VOICE IN ABSOLUTE FILE
INC DE ;NEXT VOICE IN RELATIVE FILE
PLRL02: LD A,(DE) ;GET RHYTHM BYTE
 INC DE ;POINT TO NEXT NOTE GROUP
LD (HL),A ;NO TRANSLATION OF RHYTHM
INC HL ;POINT TO NEXT NOTE GROUP
LD A,(TEMP01) ;GET NUMBER OF NOTE GROUPS LEFT
MUSIC CONTROL PROGRAM

SOUND OUTPUT

DAC1 3D C DEC A ; DECREMENT THE COUNT
DAC2 32 0000H C LD (TEMP01),A ; STORE NEW COUNT
DAC5 20 EA C JR NZ,PLRL01 ; CONTINUE UNTIL COUNT IS ZERO
DAC7 06 05 C LD B,05H ; STORE ALL RESTS AT END OF TUNE
DAC9 AF C XOR A
DAC9 77 C PLRL03: LD (HL),A
DAC0 23 C INC HL
DAC2 10 FC C DJNZ PLRL03
DAC3 11 0000H C LD DE,MRKSPC ; POINT TO START OF ABSOLUTE TUNE
DAC5 3A 0000H C LD A,(TEMPO) ; GET TEMPO
DAC4 47 C LD B,A ; PASS IT IN (B)
DAC5 CD DADC C CALL SNGTUN ; SING THE TUNE
DAC8 E1 C POP HL
DAC9 D1 C POP DE
DA0 C1 C POP BC
DA0 C9 C POP HL

SNGTUN

SING TUNE


GIVEN: 
(B) - TEMPO
(DE) - ADDRESS OF BEGINNING OF TUNE

RETURNS: NONE

SNGTUN: PUSH BC
DADD D5 C PUSH DE
DADE E5 C PUSH HL
DADE DD E5 C PUSH IX
DA1 D5 C PUSH DE ; PUT TUNE ADDRESS IN (IX)
DA2 DD E1 C PUSH IX
DAE 88 C EX AF,AF' ; SAVE ALTERNATE ACCUMULATOR
DAE5 32 0000H C LD (TEMPO1),A
DAE8 78 C LD A,B ; GET TEMPO
DAE9 32 0000H C LD (TEMPO),A ; STORE IT
DAEC CD DB73 C CALL CALC32 ; CALCULATE VALUE OF 32ND NOTE
DAEF 3A 0000H C LD A,(XPOSHS) ; GET TRANPOSITION VALUE
DAF2 4F C LD C,A ; KEEP IT IN (C) THROUGHOUT
MUSIC CONTROL PROGRAM  MACRO-80 3.44  09-Dec-81  PAGE  1-98

SOUND OUTPUT

DAF3  3A 0000X C    LD   A,(10FSET)    ;GET INTEGER OFFSET
DAF6  61 C         ADD   A,C         ;ADD IT TO TRANPOSITION
DAF7  4F C         LD   C,A         ;KEEP TOTAL IN (C)
DAF8  21 0000 C    SNTN01: LD   HL,0000H ;ZERO OUT EACH POINTER
DAF8  22 0000X C    LD   (PTR1),HL
DAFE  22 0000X C    LD   (PTR2),HL
D981  22 0000X C    LD   (PTR3),HL
D984  22 0000X C    LD   (PTR4),HL
D987  26 C1 C       LD   H,HIGH(INCTBL) ;INCREMENT TABLE ADDRESS
D989  DD 7E 00 C    LD   A,(IX+00H)    ;GET NOTE OF VOICE ONE
D98C  A7  C         AND   A            ;ZERO VALUE IS SILENCE
D98D  28 01 C       JR    Z,SNTN82     ;JUMP IF REST
D98F  81 C         ADD   A,C         ;ADD TRANPOSITION VALUE
D990  CB 27 C      SNTN02: SLA   A        ;DOUBLE IT FOR INDEX ADDRESS
D992  30 01 C       JR    NC,SNTN03    ;JUMP IF IN RANGE
D994  AF  C         XOR   A            ;ELSE SING NOTHING
D995  6F C         SNTN03: LD   L,A     ;COMPLETE ADDRESS
D996  56 C         LD   D,(HL)       ;LOAD INCREMENT
D997  23 C         INC   HL
D998  5E C         LD   E,(HL)
D999  ED 53 0000X C    LD   (INCI),DE ;STORE INCREMENT FOR VOICE ONE
D9A0  DD 7E 01 C    LD   A,(IX+01H)    ;DO THE SAME THING FOR THE
D9A2  A7  C         AND   A            ;SECOND VOICE
D9A4  28 01 C       JR    Z,SNTN84     ;THIRD VOICE
D9A6  81 C         ADD   A,C         ;FOURTH VOICE
D9A8  CB 27 C      SNTN04: SLA   A        ;DO THE SAME THING FOR THE
D9B0  30 01 C       JR    NC,SNTN05    ;SECOND VOICE
D9B2  AF  C         XOR   A            ;ELSE SING NOTHING
D9B3  6F C         SNTN05: LD   L,A     ;COMPLETE ADDRESS
D9B4  56 C         LD   D,(HL)
D9B6  23 C         INC   HL
D9B7  5E C         LD   E,(HL)
D9B8  ED 53 0000X C    LD   (INCI2),DE
D9B9  DD 7E 02 C    LD   A,(IX+02H)    ;DO THE SAME THING FOR THE
D9C1  A7  C         AND   A            ;SECOND VOICE
D9C3  28 01 C       JR    Z,SNTN86     ;THIRD VOICE
D9C5  81 C         ADD   A,C         ;FOURTH VOICE
D9C7  CB 27 C      SNTN06: SLA   A        ;DO THE SAME THING FOR THE
D9C9  30 01 C       JR    NC,SNTN07    ;SECOND VOICE
D9CA  AF  C         XOR   A            ;ELSE SING NOTHING
D9CB  6F C         SNTN07: LD   L,A     ;COMPLETE ADDRESS
D9CD  56 C         LD   D,(HL)
D9CE  23 C         INC   HL
D9CF  5E C         LD   E,(HL)
D9D0  ED 53 0000X C    LD   (INCI3),DE
D9D1  DD 7E 03 C    LD   A,(IX+03H)    ;DO THE SAME THING FOR THE
D9D3  A7  C         AND   A            ;SECOND VOICE
D9D5  28 01 C       JR    Z,SNTN88     ;THIRD VOICE

|GET INTEGER OFFSET|
|ADD IT TO TRANPOSITION|
|KEEP TOTAL IN (C)|
|ZERO OUT EACH POINTER|

|GET NOTE OF VOICE ONE|
|ZERO VALUE IS SILENCE|
|JUMP IF REST|
|ADD TRANPOSITION VALUE|
|DOUBLE IT FOR INDEX ADDRESS|
|JUMP IF IN RANGE|
|ELSE SING NOTHING|
|COMPLETE ADDRESS|
|LOAD INCREMENT|

|INCREASE TABLE ADDRESS|
|STORE INCREMENT FOR VOICE ONE|
|DO THE SAME THING FOR THE|
|SECOND VOICE|

|INCREASE ADDRESS|
|LOAD INCREMENT|
|STORE INCREMENT FOR VOICE |
|DO THE SAME THING FOR THE|
|SECOND VOICE|
|THIRD VOICE|
|FOURTH VOICE|
SOUND OUTPUT

MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-99

DB4B 81 C ADD A,C
DB4C CB 27 C SNTN08: SLA A
DB4E 30 01 C JR NC,SNTN09
DB50 AF C XOR A
DB51 6F C SNTN09: LD L,A
DB52 56 C LD D,(HL)
DB53 23 C INC HL
DB54 5E C LD E,(HL)
DB55 ED 53 0000X C LD (INCA),DE
DB59 DD 7E 04 C LD a,(IX+04H) ;GET RHYTHM BYTE
DB5C 11 0005 C LD DE,0005H ;LENGTH OF NOTE GROUP
DB5F DD 19 C ADD IX,DE ;POINT TO NEXT NOTE GROUP
DB61 A7 C AND A ;TEST FOR END OF TUNE
DB62 28 05 C JR Z,SNTN10 ;JUMP IF END OF TUNE
DB64 CD DB04 C CALL SING ;ELSE SING THE NOTES
DB67 18 8F C JR SNTN01 ;GO GET NEXT NOTE GROUP
DB69 3A 0000X C SNTN10: LD A,(TEMP01) ;RESTORE ALTERNATE ACCUMULATOR
DB8C 08 C POP IX
DB8F E1 C POP HL
DB70 D1 C POP DE
DB71 C1 C POP BC
DB72 C9 C RET

; CALC32
; CALCULATE 32ND NOTE VALUE
; CALCULATES THE DURATION OF THE 32ND NOTE IN TERMS OF THE
; TEMPO SUPPLIED AND THE SAMPLING RATE OF THE OUTPUT ROUTINE.
; ALL RHYTHMIC VALUES ARE DERIVED FROM THIS VALUE.
; GIVEN: NONE
; RETURNS: NONE

DB73 11 9858 C CALC32: LD DE,9858H ;(SAMPLING RATE X 60) / 8
DB76 3A 0000X C LD A,(TEMP0) ;GET TEMPO
DB79 4F C LD CA
DB7A 06 00 C LD B,00H
DB7C CD DEFD C CALL DIVIDE ;SAMPLING RATE CONSTANT / TEMPO
DB7F ED 53 0000X C GA3201: LD (SAMP32),DE ;STORE 32ND NOTE VALUE
DB83 C9 C RET

; SING
; SING THE FOUR NOTES WHOSE INCREMENT TABLE VALUES ARE STORED
; IN (INCI), (INC2), (INC3) AND (INC4). THE RHYTHMIC VALUE
C ; IS GIVEN IN THE ACCUMULATOR AS A MULTIPLE OF 32ND NOTE
C VALUES. THE NUMBER OF CLOCK PULSES ARE SHOWN IN THE
C COMMENT FIELD FOR THE SIGNIFICANT INSTRUCTIONS. THE
C NUMBER OF CLOCK PULSES ADDS TO 81 FOR ONE VOICE PLUS
C THE OVERHEAD OF ANOTHER 81 CLOCK PULSES. THE ENTIRE
C CYCLE TAKES (31 x 4) + 81, GIVING A TOTAL OF 485 CLOCK
C PULSES PER SAMPLE. THE SAMPLING RATE IS THE PROCESSOR
C CLOCK FREQUENCY (2186330 Hz FOR THE SORCERER) DIVIDED
C BY THE TIMING LOOP OF 485 CLOCK PULSES: 5280 SAMPLES
C PER SECOND.
C
C GIVEN: (A) - RHYTHM VALUE
C RETURNS: NONE
C
DB84 C5 C SING: PUSH BC
DB85 05 C PUSH DE
DB86 E5 C PUSH HL
DB87 88 C SING01: EX AF,AF' ; 4 SAVE RHYTHM VALUE
DB88 ED 4B 0000H C LD BC,(SAMPLE) ; 26 GET 32ND NOTE VALUE
DB8C 2A 0000H C SING02: LD HL,(PTR1) ; 16 POINTER INTO WAVEFORM
DB8F ED 5B 0000H C LD DE,(INC1) ; 20 INCREMENT INTO WAVEFORM TABLE
DB93 19 C ADD HL,DE ; 11 INTEGER IN (D); FRACTION IN (E)
DB94 22 0000H C LD (PTR1),HL ; 16 HIGH ORDER BYTE IS THE LOW ORDER
DB97 6C C LD L,H ; 4 INDEX INTO THE WAVEFORM
DB98 26 C2 C LD HL,(HIGH(WAVFORM)) ; 7 HIGH ORDER BYTE CONSTANT
DB9A 7E C LD A,(HL) ; 7 GET WAVEFORM SAMPLE
DB9B 2A 0000H C LD HL,(PTR2) ; DO THE SAME THING FOR THE
DB9E ED 5B 0000H C LD DE,(INC2) ; SECOND VOICE
DBA2 19 C ADD HL,DE
DBA3 22 0000H C LD (PTR2),HL
DBA6 6C C LD L,H
DBA7 26 C2 C LD HL,(HIGH(WAVFORM))
DBA9 86 C ADD A,(HL)
DBAB 2A 0000H C LD HL,(PTR3) ; DO THE SAME THING FOR THE
DBAD ED 5B 0000H C LD DE,(INC3) ; THIRD VOICE
DBB1 19 C ADD HL,DE
DBB2 22 0000H C LD (PTR3),HL
DBB5 6C C LD L,H
DBB6 26 C2 C LD HL,(HIGH(WAVFORM))
DBB8 86 C ADD A,(HL)
DBB9 2A 0000H C LD HL,(PTR4) ; DO THE SAME THING FOR THE
DBBC ED 5B 0000H C LD DE,(INC4) ; FOURTH VOICE
DBC0 19 C ADD HL,DE
DBC1 22 0000H C LD (PTR4),HL
DBC4 6C C LD L,H
DBC5 26 C2 C LD HL,(HIGH(WAVFORM))
DBC7 86 C ADD A,(HL)
DBC8 03 FF C OUT (DPORT),A ; 11 OUTPUT THE SAMPLE
**Music Control Program**

**Sound Output**

```
DBGA  0B  C   DEC  BC     ; 6 count down the 32nd note
DBCD  78  C   LD   A,B    ; 4 test for zero
DBCC  81  C   OR   C      ; 4
DBCD  28  07  C  JR  Z,SING03  ; 7/12 jump if end of 32nd note
DBCF  F5  C   PUSH  AF     ; 11 else kill the same amount of
DBD0  F1  C   POP   AF     ; 10 time that the other path
DBD1  2A  0000X C  LD   HL,(PTR)  ; 16 takes
DBD4  18  B6  C  JR   SING02  ; 12 now continue
DBD6  08  C  SING03: EX  AF,AF'  ; 4 get rhythm value
DBD7  3D  C   DEC  A      ; 4 count 32nd note values
DBD8  20  AD  C  JR  NZ,SING01  ; 12 continue if not finished
DBDA  E1  C   POP   HL     
DBDB  D1  C   POP   DE     
DBDC  C1  C   POP   BC     
DBDD  C9  C   RET         

; 
; SETOFS
; 
; set integer offset for sound output only. This field
; is added to the pitches that are played so that tunes
; that are displayed in one key may be played in another
; without transposing it. This facility is needed to
; allow users to sing a tune in a key that is different
; from the one that is displayed, thereby allowing the
; placement of the tune in the users vocal range.
; 
; GIVEN: (B) - integer offset
; RETURNS: NONE
; 
DBDE  78  C  SETOFS: LD  A,B
DBDF  32  0000X C  LD  (IOFS+T),A
DBE2  C9  C   RET         

; SUBTITLE SOUND INPUT
MACLIB B:MCP08.HAC

; 
; NEXTPIT
; 
; NEXT PITCH
; listens for the current pitch until it changes and then
; evaluates when the next pitch can be returned. The new
; pitch is returned when the minimum count of consecutive,
; identical pitches with any valid deviation are found.
; a check for keyboard interrupt is also made.
```
SOUND INPUT

Given: (B) - Current integer pitch
Returns: (A) - Next integer pitch

C = KBINTR - Keyboard interrupt; character in (C)
C = NOISE - Non-pitched or silent
(B) = Previous pitch
(C) = ASCII keyboard character
= 08H - No character pressed

DBE3 D5 C NXTPT: PUSH DE
DBE4 E5 C
DBE5 58 C LD D,B ; Save current pitch
DBE6 CD DEd3 C NXPT01: CALL KBFOLD ; Check for keyboard interrupt
DBE9 28 0F C JR NZ,NXPT04 ; Jump if found
DBEB CD DC0F C CALL GETNOT ; Listen for next average sample
DBEE 79 C LD A,C ; Get integer pitch average
DBEF 8A C NXTPT02: CP D ; Is it same as current pitch?
DBFO 28 F4 C JR Z,NXPT01 ; Jump until current pitch is over
DBF2 2E 05 C LD L,MINSAM-1 ; Get minimum number of samples
DBF4 67 C LD H,A ; Save new pitch
DBF5 CD DEd3 C NXPT03: CALL KBFOLD ; Check for keyboard interrupt
DBF8 28 05 C JR Z,NXPT05 ; Jump if not found
DBFA 4F C NXTPT04: LD C,A ; Return character in (C)
DBFB 3E FD C LD A,KBINTR ; Else load interrupt flag
DBFD 18 0C C JR NXTPT07 ; And exit
DBFF CD DC0F C NXTPT05: CALL GETNOT ; Listen for next sample
DC02 79 C LD A,C ; Get average pitch
DC03 8C C CP H ; Else compare with previous sample
DC04 28 E9 C JR NZ,NXPT02 ; Jump if unequal
DC06 20 C DEC L ; Else count consecutive equal ones
DC07 28 EC C JR NZ,NXPT03 ; Jump if not enough yet
DC09 0E 08 C NXTPT06: LD C,08H ; Set no keyboard flag
DC0B 42 C NXTPT07: LD B,D ; Restore previous pitch
DC0C E1 C POP HL
DC0D D1 C POP DE
DC0E C9 C RET
C ;
C ; GETNOT
C ;
C ;

given: none
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-103

SOUND INPUT

C ; RETURNS: (B) - AVERAGE DEVIATION FACTOR
C ; (C) - INTEGER PITCH AVERAGE
C ;
DC0F D5 C GETNOT: PUSH DE
DC10 ED 4B 0000X C LD BC, (SAMPL3) ; MOVE LAST 2 SAMPLES TO
DC14 ED 43 0000X C LD (SAMPL1), BC ; PREVIOUS 2 SAMPLE POSITIONS
DC18 CD DC4D C CALL GETPIT ; GET NEXT PITCH SAMPLE
DC1B FF FF C CP NOISE ; CHECK FOR NOISE
DC1D 28 04 C JR Z, GTN01 ; JUMP IF NOISE
DC1F 47 C LD B, A ; PASS IT TO CONVERT ROUTINE
DC20 CD D930 C CALL CVTATI ; CONVERT IT TO INTEGER PITCH
DC23 32 0000X C GTN01: LD (SAMPL3), A ; STORE IT IN THIRD POSITION
DC26 CD DC4D C CALL GETPIT ; GET NEXT PITCH SAMPLE
DC29 FE FF C CP NOISE ; CHECK FOR NOISE
DC2B 28 04 C JR Z, GTN02 ; JUMP IF NOISE
DC2D 47 C LD B, A ; PASS ABSOLUTE PITCH IN (B)
DC2E CD D930 C CALL CVTATI ; CONVERT TO INTEGER PITCH
DC31 32 0000X C GTN02: LD (SAMPL4), A ; STORE IT IN FOURTH POSITION
DC34 CD DC70 C CALL AVGSMPl ; GET AVERAGE OF ALL 4 SAMPLES
DC37 FE 10 C CP MINPIT ; TEST RANGE FOR MINIMUM VALUE
DC39 38 0C C JR C, GTN03 ; JUMP IF TOO LOW
DC3B FE 41 C CP MAXPIT+1 ; ELSE COMPARE WITH LARGEST VALUE
DC3D 38 08 C JR NC, GTN03 ; JUMP IF TOO HIGH
DC3F CD DC94 C CALL CALDEV ; ELSE CALCULATE DEVIATION
DC42 78 C LD A, B ; GET DEVIATION FACTOR
DC43 FE 06 C CP MAXDEV+1 ; TEST FOR TOO MUCH DEVIATION
DC45 38 04 C JR C, GTN04 ; JUMP IF VALUE IS OK
DC47 8E FF C GTN03: LD C, NOISE ; RETURN PITCH AS NOISE
DC49 86 FF C LD B, 0FFH ; SET DEVIATION OUT OF RANGE
DC4B D1 C GTN04: POP DE
DC4C C9 C RET

C ;
C ; GETPIT
C ;
C ; GET PITCH
C ; GETS THE NEXT FREQUENCY SAMPLE AVAILABLE AND CONVERTS IT
C ; TO THE ABSOLUTE PITCH, AND UNADJUSTED PITCH AND OCTAVE.
C ; THE GOOD PITCH AND GATE SIGNALS PROVIDED BY HARDWARE IN
C ; BITS 5 AND 4, RESPECTIVELY, OF (D) ARE IGNORED.
C ;
C ; GIVEN: NONE
C ; RETURNS: (A) - ABSOLUTE PITCH
C ; (B) = NOISE - WHEN FREQUENCY = 0
C ; (C) - UNADJUSTED PITCH
C ; (D) - OCTAVE
C ;
DC4D D5 C GETPIT: PUSH DE
SOUND INPUT

MUSIC CONTROL PROGRAM MACRO-88 3.44 09-Dec-81 PAGE 1-184

DC4E CD DC5A C CALL GETFRQ ; GET NEXT FREQUENCY SAMPLE
DC51 3E 0F C LD A,8FH ; STRIP OFF GOOD PITCH AND
DC53 A2 C AND D ; GATE SIGNALS
DC54 57 C LD D,A
DC55 CD DCAD C CALL FRQPIT ; CONVERT TO ABSOLUTE PITCH
DC58 D1 C GTPT02: POP DE
DC59 C9 C RET

DC5A C5 C ; GETFRQ
DC5B 8E FF C LD C,DPORT ; POINT TO OUTPUT PORT
DC5D 3E 01 C LD A,01H ; SELECT HIGH ORDER BYTE
DC5F ED 79 C OUT (C),A
DC61 DB FE C GTFR01: IN A,(STPORT) ; CHECK STATUS
DC63 CB 7F C BIT 7,A ; DATA AVAILABLE
DC65 2B FA C JR 2,GTFR01 ; WAIT TILL DATA IS READY
DC67 ED 58 C IN D,(C) ; NOW GET HIGH ORDER BYTE
DC69 AF C XOR A ; ZERO OUT ACCUMULATOR
DC6A ED 79 C OUT (C),A ; SELECT LOW ORDER BYTE
DC6C ED 58 C IN E,(C) ; GET LOW ORDER BYTE
DC6E C1 C POP BC
DC6F C9 C RET

DC70 C5 C ; AVGSMIP
DC71 D5 C ; AVERAGE SAMPLES
DC72 C5 C ; CALCULATES THE ROUNDED AVERAGE OF THE FOUR VALUES IN
DC73 C ; THE LOCATIONS STARTING AT SAMPL1.
DC74 C ; GIVEN:  NONE
DC75 C ; RETURNS:  (A)  -  AVERAGE
DC76

DC77 C5 C AVGSMIP: PUSH BC
DC78 DE
SOUND INPUT

DC72 E5 C PUSH HL
DC73 21 0000 C LD HL,SAMPL1 ;POINT TO FIRST SAMPLE
DC76 11 0000 C LD DE,0000H ;START SUM AT ZERO
DC79 06 04 C LD B,NUMSMP ;LOAD NUMBER OF SAMPLES
DC7B 7E C AVSM01: LD A,(HL) ;GET SAMPLE
DC7C 83 C ADD A,E ;ADD TO RUNNING TOTAL
DC7D 30 01 C JR NC,AVSM02 ;JUMP IF NO OVERFLOW
DC7F 14 C INC D ;ELSE ADJUST FOR OVERFLOW
DC80 5F C AVSM02: LD E,A
DC81 23 C INC HL ;POINT TO NEXT SAMPLE
DC82 10 F7 C DJNZ AVSM01 ;JUMP UNTIL ALL SAMPLES TOTALED
DC84 CB 3A C SRL D ;DIVIDE BY THE NUMBER OF
DC86 CB 1B C RR E ; SAMPLES (4) BY SHIFTING
DC88 CB 3A C SRL D ; RIGHT TWICE
DC8A CB 1B C RR E
DC8C 30 01 C JR NC,AVSM03 ;JUMP IF NO ROUND UP
DC8E 1C C INC E ;ELSE ROUND UP
DC90 7B C AVSM03: LD A,E ;GET RESULT
DC93 C9 C CALDEV
DC94 D5 C CALDEV: PUSH DE
DC95 E5 C PUSH HL
DC96 1E 00 C LD E,00H ;START SUM AT ZERO
DC99 21 0000 C LD HL,SAMPL1 ;POINT TO FIRST SAMPLE
DC9B 4F C LD C,A ;GET AVERAGE
DC9C 06 04 C LD B,NUMSMP ;COUNT NUMBER OF SAMPLES
DC9E 7E C CLDV01: LD A,(HL) ;GET SAMPLE
DC9F 23 C INC HL ;POINT TO NEXT SAMPLE
DCA0 91 C SUB C ;SUBTRACT AVERAGE
DCA1 30 02 C JR NC,CLDV02 ;JUMP IF RESULT IS POSITIVE
DCA3 ED 44 C NEG ;ELSE GET ABSOLUTE VALUE

CALCULATE DEVIATION


GIVEN: (A) - AVERAGE OF THE 4 SAMPLES
RETURNS: (B) - AVERAGE DEVIATION FACTOR
(C) - AVERAGE GIVEN IN (A)
SOUND INPUT

HUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-106

SOUND INPUT

DCA5 83 C  CLDV02: ADD A,E ;ADD TO RUNNING TOTAL
DCA6 5F C  LD E,A
DCA7 18 F5 C  DJNZ CLDV01 ; JUMP TILL ALL SAMPLES DONE
DCA9 43 C  LD B,E ;LOAD RESULT
DCAA E1 C  POP HL
D CAB D1 C  POP DE
DCAE C9 C  RET

;FREQPIT
;
;FREQUENCY TO PITCH
;
;CONVERTS ANY FREQUENCY TO ITS CLOSEST PITCH/OCTAVE. ONLY ONE
;OCTAVE OF FREQUENCIES IS STORED IN THE FREQUENCY TABLE SO
;THE GIVEN FREQUENCY MUST BE EITHER HALVED OR DOUBLED UNTIL
;IT FITS IN THE RANGE OF FREQUENCIES IN THE TABLE. NOTE THAT
;THIS MAY PRODUCE MISINTERPRETATIONS OF HIGH PITCHES SINCE
;THESE TEND TO BE STRETCHED HIGHER TO SATISFY THE EAR.
;
;THE ABSOLUTE PITCH IS IN THE FOLLOWING FORMAT:
;
;BIT
;VALUE  MEANING
;7-4  -8 TO +7  OCTAVE
;3-0  0 TO 11  ABSOLUTE PITCH: A=0, A#/B-FLAT=1
;B=2, C=3, C#/#D-FLAT=4, ETC.
;
;GIVEN:  (DE) - FREQUENCY
;RETURNS:  (A) - ABSOLUTE PITCH
;   = NOISE - IF FREQUENCY = 0
;   (B) - UNADJUSTED PITCH
;   (C) - OCTAVE DESIGNATION
;
Dcad 05 C  FRQPIT: PUSH DE
Dcae E5 C  PUSH HL
DCAF 7A C  LD A,D ;TEST FOR ZERO
Dcb8 B3 C  OR E
Dcb1 26 04 C  JR NZ,FRPT01 ; JUMP IF NON-ZERO
Dcb3 3E FF C  LD A,NOISE ;ELSE LOAD NOISE
Dcb5 18 54 C  JR FRPT08 ; AND EXIT
Dcb7 AF C  FRPT01: XOR A ;START OCTAVE SHIFT COUNTS AT 0
Dcb8 ED 4B C318 C  LD BC,(HIFREQ) ;GET HIGHEST FREQUENCY IN TABLE
DcbC CD DF1F C  CALL DCOMP ;COMPARE FREQUENCIES
DcbF 30 10 C  JR NC,FRPT03 ; JUMP IF GIVEN FREQ IS HIGHER
Dcd1 ED 4B C300 C  LD BC,(LOFFREQ) ;GET LOWEST FREQUENCY IN TABLE
Dcd5 CD DF1F C  FRPT02: CALL DCOMP ;COMPARE WITH LOWEST
Dcc8 30 11 C  JR NC,FRPT04 ; JUMP WHEN GIVEN FREQ IS HIGHER
Dcaa CB 23 C  SLA E ;ELSE MULTIPLY GIVEN FREQ BY 2
Dccc CB 12 C  RL D
Dcce 3D C  DEC A ;COUNT OCTAVE SHIFTS UP
Dccf 18 F4 C  JR FRPT02 ; CONTINUE TILL IT FITS IN TABLE
SOUND INPUT

DC01 CB 3A C FRPT03: SRL D ;DIVIDE GIVEN FREQ BY 2
DC03 CB 1B C RR E ;COUNT OCTAVE SHIFTS DOWN
DC05 3C C INC A ;COMPARE AGAIN
DC06 CD DF1F C CALL DCOMP ;KEEP DIVIDING TILL IN TABLE
DC09 30 F6 C JR NC,FRPT03 ;POINT TO 2ND FREQ HIGH BYTE
D0B 21 C303 C FRPT04: LD HL,LOWFREQ+03H ;LOAD OCTAVE SHIFTS
D0E 47 C LD B,A ;SAVE OCTAVE SHIFTS
DCF 7A C FRPT05: LD A0,D ;LOAD HIGH BYTE OF GIVEN FREQ
DCE8 BE C FRPT06: CP (HL) ;COMPARE WITH TABLE HIGH BYTE
DCE1 28 06 C JR Z,FRPT07 ;JUMP IF SAME
DCE3 38 10 C JR C,FRPT08 ;FOUND WHEN LESS THAN
DCE5 23 C INC HL ;ELSE POINT TO NEXT FREQ
DCE6 23 C INC HL
DCE7 18 F7 C JR FRPT06 ;CONTINUE LINEAR SEARCH
DCE9 7B C FRPT07: LD A0,E ;COMPARE LOW ORDER BYTES
DCEA 2B C DEC HL ;POINT TO LOW ORDER BYTE
DCEB BE C CP (HL) ;COMPARE
DCEC 38 07 C JR C0,FRPT08 ;IF SMALLER, THEN IT'S PREVIOUS
DCEE 28 09 C JR Z,FRPT09 ;IF EQUAL, THEN IT'S CURRENT
DCF0 23 C INC HL ;ELSE TRY NEXT HIGHER FREQUENCY
DCF1 23 C INC HL
DCF2 23 C INC HL ;3RD TIME POINTS TO HIGH BYTE
DCF3 18 EA C JR FRPT05
DCF5 28 C FRPT08: DEC HL ;POINT TO PREVIOUS FREQUENCY
DCF6 28 C DEC HL
DCF7 CB 95 C RES 01,L ;TRUNCATE TO NEAREST EVEN NUMBER
DCE9 70 C FRPT09: LD A0,L ;PITCH IS FUNCTION OF ADDRESS
DCF0 D6 00 C SUB (HL), (LOWFREQ) ;GET DISPLACEMENT INTO TABLE
DCF0 CB 3F C SRL A ;ABSOLUTE PITCH IS ADDRESS / 2
DCEA CB 29 C SLA B ;PUT OCTAVE IN HIGH ORDER 4 BITS
D080 CB 28 C SLA B
D082 CB 28 C SLA B
D084 CB 28 C SLA B
D086 B0 C OR B ;PLACE OCTAVE BY ABSOLUTE PITCH
D087 47 C LD B,A ;PUT IN (B) FOR CALL
D088 CD D8F9 C CALL CVTATU ;CONVERT TO UNADJUSTED PITCH
D08D E1 C FRPT10: POP HL
D0C D1 C POP DE
D0D C9 C RET
SOUND INPUT

C
GIVEN: NONE
C
RETURNS: NONE
C

DD0E C5
C
INITPT: PUSH BC
DD0F E5
C
PUSH HL
DD10 21 0000%
C
LD HL,SAMPL1 ; POINT TO FIRST SAMPLE
DD13 3E FF
C
LD A,NOISE ; INITIALIZE TO NOISE
DD15 06 04
C
LD B,NUMSMP ; COUNT NUMBER OF SAMPLES
DD17 77
C
INPT01: LD (HL),A ; STORE NOISE IN TABLE
DD19 23
C
INC HL ; POINT TO NEXT LOCATION
DD1A 10 FC
C
DJNZ INPT01 ; JUMP TILL ALL DONE
DD1C E1
C
POP HL
DD1D C1
C
POP BC
DD1E C9
C
RET

SUBTL MISCELLANEOUS ROUTINES
MACLIB B:MCP89.MAC

PRTEXT

C
PRINT TEXT
C
DISPLAY TEXT IN ACTIVE PARTITION. ADDRESS OF TEXT STRING IS
GIVEN AND AN END-OF-STRING CHARACTER IS ASSUMED TO BE
PRESENT AT THE END OF THE STRING. THE STARTING POSITION IS
GIVEN AS AN X/Y COORDINATE. IF BOTH COORDINATES ARE ZERO, THE
CURRENT X/Y COORDINATES ARE USED. ANY DISPLAYABLE CHARACTER
AND 7 CONTROL CODES MAY BE INCLUDED IN THE TEXT STRING. THE
CONTROL CODES RECOGNIZED ARE: CARRIAGE RETURN (0DH), LINE
FEED (0AH), CURSOR UP (17H), CURSOR RIGHT (13H), CURSOR LEFT
(0EH), HOME (11H) AND TAB (09H). WHEN A TAB IS FOUND, THE
NEXT BYTE IS ASSUMED TO BE THE NUMBER OF SPACES TO TAB.
NO TEXT IS ALLOWED TO BE PRINTED OUTSIDE OF THE PARTITION
IN WHICH IT IS BEING PRINTED.
C

C
GIVEN: (B) - STARTING POSITION X COORDINATE
C
(C) - STARTING POSITION Y COORDINATE
C
(DE) - ADDRESS OF TEXT STRING
C
(IY) - ADDRESS OF PARTITION TABLE ENTRY
C
RETURNS: (DE) - POINTER TO 1 BYTE PAST END OF STRING FLAG
C

DD1E C5
C
PRTEXT: PUSH BC
DD1F E5
C
PUSH HL
DD20 78 C
C
LD A,B ; TEST GIVEN X/Y COORDINATES
DD21 B1
C
OR C
DD22 28 86
C
JR 2,PRTEXT01 ; JUMP IF BOTH ARE ZERO
MUSIC CONTROL PROGRAM MACRO-88 3.44 09-Dec-81 PAGE 1-109

MISCELLANEOUS ROUTINES

D024 FD 70 05 C LD (IY+COORD),B \(\text{; ELSE USE AS NEW STARTING}\)
D027 FD 71 04 C LD (IY+COORD),C \(\text{; POSITION}\)
D02A FD 66 01 C PTXT01: LD H,(IY+PARADH) \(\text{; GET BASE ADDRESS OF PARTITION}\)
D02D FD 4E 00 C LD L,(IY+PARALD) \(\text{; GET CURSOR POSITION}\)
D030 FD 4E 05 C LD B,(IY+COORD) \(\text{; GET CURSOR POSITION}\)
D033 FD 4E 04 C LD C,(IY+COORD) \(\text{; GET CURSOR POSITION}\)
D036 CD DE81 C CALL XYCORD \(\text{; GET STARTING ADDRESS}\)
D039 1A C PTXT02: LD A,(DE) \(\text{; CHARACTER}\)
D03A 13 C INC DE \(\text{; POINT TO NEXT CHARACTER}\)
D03B FE 00 C CP EOSTRG \(\text{; CHECK FOR END OF STRING}\)
D03D 28 07 C JR Z,PTXT11 \(\text{; JUMP IF END FOUND}\)
D03F FE 20 C CP 20H \(\text{; CONTROL OR DISPLAYABLE CHARACTER?}\)
D041 30 03 C JR C,PTXT03 \(\text{; JUMP IF CONTROL CHARACTER}\)
D043 77 C LD (HL),A \(\text{; DISPLAY CHARACTER}\)
D044 18 2D C JR PTXT07 \(\text{; UPDATE CURSOR POSITION}\)
D046 FE 00 C PTXT03: CP CARRET \(\text{; CARRIAGE RETURN}\)
D048 20 11 C JR NZ,PTXT05 \(\text{; JUMP IF NOT}\)
D04A FD 3E 05 01 C LD (IY+COORD),81H \(\text{; POINT TO START OF LINE}\)
D04E FD 7E 04 C PTXT04: LD A0,(IY+COORD) \(\text{; GET CURRENT LINE NUMBER}\)
D051 FD BE 03 C CP (IY+PARLEN) \(\text{; COMPARE WITH BOTTOM NUMBER}\)
D054 30 84 C JR NC,PTXTB1 \(\text{; JUMP IF ALREADY ON BOTTOM LINE}\)
D056 FD 34 04 C INC (IY+COORD) \(\text{; ELSE POINT TO NEXT LINE}\)
D059 1A CF C JR PTXT01 \(\text{; BEGIN UPDATE}\)
D05B FE 0A C PTXTB5: CP LINFED \(\text{; TEST FOR LINE FEED}\)
D05D 2B CF C JR Z,PTXT04 \(\text{; JUMP IF NOT}\)
D05F FE 17 C CP CURSUP \(\text{; MOVE CURSOR UP?}\)
D061 20 0C C JR NZ,PTXT06 \(\text{; JUMP IF NOT}\)
D063 FD 7E 04 C LD A0,(IY+COORD) \(\text{; GET CURRENT LINE NUMBER}\)
D066 FE 02 C CP 02H \(\text{; COMPARE WITH SECOND LINE}\)
D068 38 CF C JR C,PTXT02 \(\text{; JUMP IF ALREADY ON TOP LINE}\)
D06A FD 35 04 C DEC (IY+COORD) \(\text{; POINT TO NEXT LINE ABOVE}\)
D06D 18 BB C JR PTXT01 \(\text{; BEGIN UPDATE}\)
D06F FE 13 C PTXT06: CP CURRT \(\text{; COMPARE WITH CURSOR RIGHT}\)
D071 20 0E C JR NZ,PTXT08 \(\text{; JUMP IF NOT}\)
D073 FD 7E 05 C PTXT07: LD A0,(IY+COORD) \(\text{; GET CURRENT POSITION}\)
D076 FD BE 02 C CP (IY+PARLEN) \(\text{; COMPARE WITH LINE LENGTH}\)
D079 30 BE C JR NC,PTXTB2 \(\text{; JUMP IF ALREADY AT END OF LINE}\)
D07B FD 34 05 C INC (IY+COORD) \(\text{; ELSE INCREMENT TO NEXT SPACE}\)
D07E 23 C INC HL \(\text{; POINT TO NEXT SCREEN POSITION}\)
D07F 1B B8 C JR PTXT02 \(\text{; BEGIN UPDATE}\)
D081 FE 01 C PTXT08: CP CULT \(\text{; COMPARE WITH CURSOR LEFT}\)
D083 20 0D C JR NZ,PTXT09 \(\text{; JUMP IF NOT}\)
D085 FD 7E 05 C LD A0,(IY+COORD) \(\text{; GET CURRENT POSITION}\)
D088 FE 02 C CP 02H \(\text{; COMPARE WITH MINIMUM POSITION}\)
D08A 38 AD C JR C,PTXT02 \(\text{; JUMP IF ALREADY AT START OF LINE}\)
D08C FD 35 05 C DEC (IY+COORD) \(\text{; ELSE POINT TO PREVIOUS POSITION}\)
D08F 2B C DEC HL \(\text{; POINT TO PREVIOUS SCREEN POSITION}\)
D090 1A A7 C JR PTXT02 \(\text{; BEGIN UPDATE}\)
MUSIC CONTROL PROGRAM  MACRO-80 3.44  09-Dec-81  PAGE 1-118

MISCELLANEOUS ROUTINES

DD92  FE 11  C  PTXT09:  CP  HOME  ;COMPARE WITH HOME
DD94  20  8A  C  JR  NZ,PTXT10  ;ELSE IGNORE CHARACTER
DD96  FD 36 05 01  C  LD  (IY+COORD),01H  ;POINT TO HOME POSITION
DD98  FD 36 04 01  C  LD  (IY+COORD),01H
DD9E  18 8A  C  JR  PTXT01
DDA0  FE 89  C  PTXT10:  CP  TAB  ;CHECK FOR TAB
DDA2  20 95  C  JR  NZ,PTXT82  ;IGNORE ANY OTHER CONTROL CODE
DDA4  1A  C  LD  A,(DE)  ;GET NUMBER OF POSITIONS TO TAB
DDA5  13  C  INC  DE  ;POINT TO NEXT CHARACTER
DDA6  47  C  LD  B,A  ;SAVE NUMBER
DDA7  FD 86 05  C  ADD  A,(IY+COORD)  ;ADD TO CURRENT POSITION
DDA8  FD BE 02  C  CP  (IY+PARLEN)  ;CHECK FOR END OF LINE
DDA9  30 8A  C  JR  NC,PTXT82  ;IGNORE IT IF END OF LINE
DDAF  FD 77 05  C  LD  (IY+COORD),A  ;ELSE UPDATE CURRENT POSITION
DDB0  7D  C  LD  A,L  ;UPDATE SCREEN POSITION
DDB1  80  C  ADD  A,B
DDB2  4F  C  LD  L,A
DDB3  30 82  C  JR  NC,PTXT82  ;GET NEXT CHARACTER
DDB4  24  C  INC  H  ;ADJUST ON OVERFLOW
DDB5  C3 DD39  C  JP  PTXT02  ;NOW GET NEXT CHARACTER
DDB6  E1  C  PTXT11:  POP  HL
DDB7  C1  C  POP  BC
DDB8  C9  C  RET

C
C

; HSFRNT
C
C
HALF STEPS FRONT END
C
ALLOWS THE HALF STEP ROUTINE TO BE USED AS A SYSTEM
C CALL. SINCE ALL SYSTEM CALLS CANNOT USE (HL) TO
C PASS DATA, THE DATA WILL HAVE TO BE POINTED TO BY
C (DE) AND PICKED UP BEFORE CALLING 'HFLSTP'.
C
; GIVEN:  (DE) - POINTER TO DATA IN THE FORMAT:
C LETTER 1, ACCIDENTAL 1, OCTAVE 1,
C LETTER 2, ACCIDENTAL 2, OCTAVE 2
C
; RETURNS:  (A) - COMPLETION CODE
C = 00 - SUCCESSFUL
C = FF - INVALID DATA
C
; (B) - NUMBER OF HALF STEPS BETWEEN THE 2 NOTES
C (C) - NUMBER OF HALF STEPS WITHIN 1 OCTAVE
C
C
DDBE  D5  C  HSFRNT:  PUSH  DE
DDBF  E5  C  PUSH  HL
DC08  EB  C  EX  DE,HL  ;PUT ADDRESS IN (HL)
DC09  46  C  LD  B,(HL)  ;GET LETTER 1
DC0A  23  C  INC  HL
DC0B  4E  C  LD  C,(HL)  ;GET ACCIDENTAL 1
MISCELLANEOUS ROUTINES

DDC4 23 C INC HL
DDC5 56 C LD D,(HL) ;GET OCTAVE 1
DDC6 23 C INC HL
DDC7 5E C LD E,(HL) ;GET LETTER 2
DDC8 23 C INC HL
DDC9 05 C PUSH DE
DDCA 56 C LD D,(HL) ;GET ACCIDENTAL 2
DDCB 23 C INC HL
DDCC 5E C LD E,(HL) ;GET OCTAVE 2
DDCD EB C EX DE,HL ;PUT ACCIDENTAL/OCTAVE IN (HL)
DDCE D1 C POP DE ;RESTORE OCTAVE 1/LETTER 2
DDCF CD D0D5 C CALL HLFSTP ;GET NUMBER OF HALF STEPS
DDD2 E1 C POP HL
DDD3 D1 C POP DE
DDD4 C9 C RET

DDD5 05 L HLFSTP
DDD6 01 E HLFSTP: PUSH DE
DDD7 CD D06C C CALL CVTLTA ;CONVERT NOTE 1 TO ABSOLUTE
DDDA FE FF C CP 0FFH ;TEST FOR INVALID DATA
DDDC 28 2F C JR Z,HFST03 ;EXIT IF IN ERROR
DDDE 47 C LD B,A ;PASS ABSOLUTE PITCH IN (B)
DDDF CD D0930 C CALL CVTATI ;CONVERT TO ABSOLUTE INTEGER
DDE2 FE FF C CP 0FFH ;TEST FOR INVALID DATA
DDE4 28 27 C JR Z,HFST03 ;EXIT IF IN ERROR
MUSIC CONTROL PROGRAM  MACRO-88 3.44  09-Dec-81  PAGE  1-112  

MISCELLANEOUS ROUTINES

D0E6  43  C   LD  B,E  ;LETTER OF NOTE 2
D0E7  4C  C   LD  C,H  ;ACCIDENTAL OF NOTE 2
D0E8  55  C   LD  D,L  ;OCTAVE OF NOTE 2
D0E9  5F  C   LD  E,A  ;SAVE NOTE 1 INTEGER
D0EA  CD D86C  C   CALL CVTLTA  ;CONVERT NOTE 2 TO ABSOLUTE
D0EB  FE FF  C   CP  0FFH  ;TEST FOR INVALID DATA
D0EC  28 1C  C   JR  Z,H,FST03  ;JUMP IF DATA IS VALID
D0ED  47  C   LD  B,A  ;PASS ABSOLUTE PITCH IN (B)
D0EE  CD D930  C   CALL CVTATI  ;CONVERT TO ABSOLUTE INTEGER
D0EF  FE FF  C   CP  0FFH  ;TEST FOR ERROR
D0F0  28 14  C   JR  Z,H,FST03  ;EXIT IF IN ERROR
D0F1  93  C   SUB  E  ;NOTE 2 MINUS NOTE 1
D0F2  ED 44  C   NEG  ;CHANGE SIGN OF NOTE 1 MINUS NOTE 2
D0F3  FE 0C  C   LD  B,A  ;STORE RESULT
D0F4  CB 7F  C   BIT  7,A  ;TEST SIGN BIT
D0F5  28 02  C   JR  Z,H,FST01  ;JUMP IF RESULT IS POSITIVE
D0F6  ED 44  C   NEG  ;ELSE MAKE IT POSITIVE
D0F7  FE 0C  C   LD  B,A  ;SUCCESSFUL CODE
D0F8  38 04  C   JR  Z,H,FST02  ;JUMP IF IT IS
D0F9  D6 8C  C   SUB  0CH  ;SUBTRACT 1 OCTAVE UNTIL
D0FA  18 F8  C   JR  HFST01  ;LESS THAN 1 OCTAVE
D0FB  4F  C   LD  C,A  ;RETURN RESULT IN (C)
D0FC  AF  C   XOR  A  ;RETURN RESULT IN (C)
D0FD  E1  C   HFST03  ;POP HL  ;SET SUCCESSFUL CODE
D0FE  D1  C   POP  DE
D0FF  C9  C   RET

ACONKY

ACONKY:  PUSH  BC

ACONKY:  PUSH  DE

ACONKY:  PUSH  HL

ACONKY:  LD  E,B  ;SAVE LETTER NAME

ACONKY:  LD  HL,KEYLET  ;POINT TO CURRENT KEY LETTER

ACONKY:  LD  B,HL  ;GET CURRENT KEY LETTER

ACONKY:  INC  HL  ;POINT TO CURRENT KEY ACCIDENTAL

ACONKY:  LD  C,HL  ;GET CURRENT KEY ACCIDENTAL

ACONKY:  CALL CVTLTA  ;CONVERT NOTE 2 TO ABSOLUTE

ACONKY:  CP  0FFH  ;TEST FOR INVALID DATA

ACONKY:  JR  Z,H,FST03  ;JUMP IF DATA IS VALID

ACONKY:  CALL CVTATI  ;CONVERT TO ABSOLUTE INTEGER

ACONKY:  CP  0FFH  ;TEST FOR ERROR

ACONKY:  JR  Z,H,FST03  ;EXIT IF IN ERROR

ACONKY:  SUB  E  ;NOTE 2 MINUS NOTE 1

ACONKY:  NEG  ;CHANGE SIGN OF NOTE 1 MINUS NOTE 2

ACONKY:  LD  B,A  ;STORE RESULT

ACONKY:  BIT  7,A  ;TEST SIGN BIT

ACONKY:  JR  Z,H,FST01  ;JUMP IF RESULT IS POSITIVE

ACONKY:  NEG  ;ELSE MAKE IT POSITIVE

ACONKY:  LD  B,A  ;SUCCESSFUL CODE

ACONKY:  JR  Z,H,FST02  ;JUMP IF IT IS

ACONKY:  SUB  0CH  ;SUBTRACT 1 OCTAVE UNTIL

ACONKY:  JR  HFST01  ;LESS THAN 1 OCTAVE

ACONKY:  LD  C,A  ;RETURN RESULT IN (C)

ACONKY:  XOR  A  ;RETURN RESULT IN (C)

ACONKY:  POP  BC

ACONKY:  POP  DE

ACONKY:  POP  HL

ACONKY:  LD  E,B  ;SAVE LETTER NAME

ACONKY:  LD  HL,KEYLET  ;POINT TO CURRENT KEY LETTER

ACONKY:  LD  B,HL  ;GET CURRENT KEY LETTER

ACONKY:  INC  HL  ;POINT TO CURRENT KEY ACCIDENTAL

ACONKY:  LD  C,HL  ;GET CURRENT KEY ACCIDENTAL
MISCELLANEOUS ROUTINES

DE1A 23  C  INC  HL  ;POINT TO MAJOR/MINOR FLAG
DE1B 56  C  LD  D,(HL)  ;GET MAJOR/MINOR FLAG
DE1C CD CFA0  C  CALL  GETKEY  ;GET NUMBER/TYPRE OF ACCIDENTAL
DE1F 4F  C  LD  C,A  ;SAVE ACCIDENTALS IN KEY
DE20 A7  C  AND  A  ;TEST FOR NO ACCIDENTALS
DE21 28 18  C  JR  Z,ACKN03  ;JUMP IF NONE
DE23 FE FF  C  CP  OFFH  ;CHECK FOR INVALID KEY
DE25 28 14  C  JR  Z,ACKN03  ;JUMP IF INVALID KEY
DE27 E6 87  C  AND  07H  ;GET NUMBER OF ACCIDENTALS IN KEY
DE29 47  C  LD  B,A  ;COUNT THEM IN (B)
DE2A 21 C412  C  LD  HL,KSFLAT  ;POINT TO LETTERS IN FLAT KEYS
DE2D CB 69  C  BIT  FLTSHA,C  ;TEST FLAT/SHARP OF CURRENT KEY
DE2F 28 83  C  JR  Z,ACKN01  ;JUMP IF FLAT KEY
DE31 21 C419  C  LD  HL,KSSHAP  ;ELSE POINT TO LETTERS IN SHARPS
DE34 7B  C  ACKN01: LD  A,E  ;GET GIVEN LETTER NAME
DE35 BE  C  ACKN02: CP  (HL)  ;SEE IF IT IS IN THE KEY SIGNATURE
DE36 23  C  INC  HL  ;POINT TO NEXT LETTER
DE37 28 06  C  JR  Z,ACKN04  ;JUMP IF FOUND
DE39 18 FA  C  DJNZ  ACKN02  ;ELSE CONTINUE SEARCH TILL END OF KEY
DE3B 3E 08  C  ACKN03: LD  A,08H  ;FALL HERE IF LETTER NOT IN KEY SIG
DE3D 18 08  C  JR  ACKN05  ;FLAG IS SET IN EITHER CASE
DE3F 3E FF  C  ACKN04: LD  A,0FFH  ;ASSUME FLAT
DE41 CB 69  C  BIT  FLTSHA,C  ;TEST FLAT/SHARP BIT OF CURRENT KEY
DE43 28 02  C  JR  Z,ACKN05  ;JUMP IF FLAT
DE45 3E 81  C  LD  A,01H  ;LOAD SHARP
DE47 E1  C  ACKN05: POP  HL
DE48 D1  C  POP  DE
DE49 C1  C  POP  BC
DE4A C9  C  RET

; ; ; ; ; ; ;

DE4B FD 70 05  C  SETCP: LD  (IY+CURPOS),B
DE4E FD 71 04  C  LD  (IY+STFORM),C
DE51 C9  C  RET

; ; ; ; GETCP

; ; ; ; GET CURRENT POSITION
; Given: None
; (IY) - Address of partition table entry
; Returns: (B) - Current position (X coordinate)
; (C) - Staff orientation (Y coordinate)

DE52 FD 46 05
DE55 FD 4E 04
DE50 C9

GETCP: LD B, (IY+CURPOS)
LD C, (IY+STFORN)
RET

; Inhibit
; Inhibit store
; Sets a flag indicating that notes are not to be stored
; in the tune table when they are displayed.

DE59 3E FF
DE5B 32 0000
DE5E C9

INHIBT: LD A, 0FFH ; Non-zero means inhibit
LD (INH STO), A ; Store it
RET

; Enable store
; Resets the flag indicating that notes are to be stored
; in the tune table as they are displayed.

DE5F AF
DE60 32 0000
DE63 C9

ENASTO: XOR A
LD (INH STO), A ; Zero means enable store
RET

MACLIB B:MCP10.MAC

; Staff address calculation
; Finds the absolute address of the current position on the
; given line for the active staff in the active partition.
; No check is made to see if the resultant address is in
C ; THE ACTIVE PARTITION.
C ;
C ; GIVEN:  (A) - LINE NUMBER FROM TOP OF PARTITION RELATIVE TO 0
C ;  (IY) - ADDRESS OF PARTITION TABLE ENTRY
C ;  RETURNS: (HL) - ADDRESS OF CURRENT POSITION ON GIVEN LINE
C ;
DE64 C5 C SAGLC: PUSH BC
DE65 D5 C PUSH DE
DE66 4F C LD C,A ; PREPARE TO MULTIPLY
DE67 06 00 C LD B,00 ; NUMBER OF LINES
DE69 11 00 40 C LD DE,LINLEN ; BY LINE LENGTH
DE6C CD DEE3 C CALL MLTPLY ; MULTIPLY (BC) BY (DE)
DE6F FD 66 01 C LD H,(IY+PARADH) ; BASE ADDRESS OF PARTITION
DE72 FD 6E 00 C LD L,(IY+PARADL)
DE75 19 C ADD HL,DE ; ADD NUMBER OF LINES
DE76 FD 7E 05 C LD A,(IY+CURPOS) ; ADD HORIZONTAL
DE79 85 C ADD A,L ; DISPLACEMENT
DE7A 4F C LD L,A
DE7B 30 01 C JR NC,SACL01 ; TEST FOR OVERFLOW
DE7D 24 C INC H
DE7E D1 C SACL01: POP DE
DE7F C1 C POP BC
DE80 C9 C RET
C ;
C ; XYCORD
C ;
C ; X-Y COORDINATE
C ; CALCULATES THE ADDRESS OF THE X-Y COORDINATE POSITION
C ; RELATIVE TO THE BASE ADDRESS SUPPLIED. IF THE BASE
C ; ADDRESS SUPPLIED IS THE BASE ADDRESS OF A PARTITION
C ; THEN THE X-Y COORDINATE FUNCTIONS RELATIVE TO THAT
C ; PARTITION. NOTE THAT THERE IS NO CHECK TO SEE IF THE
C ; X-Y COORDINATE LIES OUTSIDE OF THE PARTITION. IF THE
C ; X COORDINATE IS GREATER THAN THE LINE LENGTH OF THE
C ; SCREEN, IT IS SET TO ZERO. IF THE Y COORDINATE IS
C ; LARGER THAN THE NUMBER OF LINES ON THE SCREEN, IT TOO
C ; IS SET TO ZERO.
C ;
C ; GIVEN:  (B) - X COORDINATE RELATIVE TO 1
C ;  (C) - Y COORDINATE RELATIVE TO 1
C ;  (HL) - BASE ADDRESS
C ;  RETURNS: (BC) - CHANGED
C ;  (HL) - ADDRESS OF X-Y COORDINATE RELATIVE TO GIVEN
C ;  BASE ADDRESS
C ;
DE81 D5 C XYCORD: PUSH DE
DE82 78 C LD A,B ; GET X COORDINATE
MUSIC CONTROL PROGRAM MACRO-80 3.44 09-Dec-81 PAGE 1-116

MISCELLANEOUS ROUTINES

DE83 3D C DEC A ;MAKE RELATIVE TO 0
DE84 FE 40 C CP LINLEN ;LARGER THAN SCREEN LINE LENGTH?
DE86 38 01 C JR C,XYCR01 ;JUMP IF OK
DE88 AF C XOR A ;ELSE ASSUME COLUMN 1
DE89 85 C XYCR01: ADD A,L ;ADJUST ADDRESS
DE8A 38 01 C JR NC,XYCR82 ;CHECK FOR OVERFLOW
DE8C 24 C INC H
DE8D 6F C XYCR02; LD L,A
DE8E 0D C DEC C ;MAKE Y RELATIVE TO 0
DE8F 79 C LD A,C ;GET Y COORDINATE
DE90 FE 1E C CP NUMLIN ;LARGER THAN NUM SCREEN LINES?
DE92 38 02 C JR C,XYCR03 ;JUMP IF OK
DE94 6E 00 C LD C,0OH ;ELSE ASSUME LINE 1
DE96 6E 00 C XYCR03: LD B,00H ;SET UP FOR MULTIPLY
DE98 11 0040 C LD DE,LINLEN
DE9B CD DEE3 C CALL MLTPLY ;NUM LINES TIMES LINE LENGTH
DE9E 19 C ADD HL,DE ;ADD TO BASE ADDRESS
DEA0 D1 C POP DE
DEA0 C9 C RET

; DISPLAY
; DISPLAY CHARACTER VECTOR
; DISPLAYS ANY NUMBER OF CHARACTERS IN ANY FORMAT BY USING
; A VECTOR IN THE FOLLOWING FORMAT:
; BYTE 1 - NUMBER OF CHARACTERS TO DISPLAY
; BYTE 2 - CHARACTER 1
; BYTE 3 - DISPLACEMENT TO ADD TO CURRENT ADDRESS FOR
; NEXT CHARACTER POSITION
; BYTE 4 - CHARACTER 2
; BYTE 5 - DISPLACEMENT FOR NEXT CHARACTER
; ETC.
; DISPLACEMENTS MAY BE ANY VALUE BETWEEN -128 AND +127.
; GIVEN: (DE) - ADDRESS ON SCREEN TO BEGIN DISPLAY
; (HL) - ADDRESS OF CHARACTER VECTOR
; RETURNS: NONE

DEA1 C5 C DISPLA: PUSH BC
DEA2 D5 C PUSH DE
DEA3 E5 C PUSH HL
DEA4 46 C LD B,(HL) ;GET NUMBER OF CHARACTERS
DEA5 10 0C C JR DISP03
DEA7 23 C DISP01: INC HL ;POINT TO DISPLACEMENT
DEA8 7E C LD A,(HL) ;GET DISPLACEMENT
DEA9 83 C ADD A,E ;ADD TO PREVIOUS POSITION
DEAA 5F C LD E,A
MUSIC CONTROL PROGRAM MACRO-86 3.44 09-Dec-81 PAGE 1-117
MISCELLANEOUS ROUTINES

DEAB 30 01 C JR MC,DISP02
DEAD 14 C INC D ; INCREMENT ON OVERFLOW
DEAE CB 7E C DISP02: BIT 7,(HL) ; CHECK FOR MINUS DISPLACEMENT
DEB0 28 01 C JR Z,DISP03 ; JUMP IF POSITIVE
DEB2 15 C DEC D ; DECREMENT IF NEGATIVE
DEB3 23 C DISP03: INC HL ; POINT TO NEXT CHARACTER
DEB4 7E C LD A,(HL) ; GET CHARACTER
DEB5 12 C LD (DE),A ; PRINT IT
DEB6 10 EF C DJNZ DISP01 ; CONTINUE FOR ALL CHARACTERS
DEB8 E1 C POP HL
DEB9 D1 C POP DE
DEBA C1 C POP BC
DEBB C9 C RET

; ALNCHK
; ACCIDENTAL/LETTER NAME CHECK
; VERIFIES THE VALIDITY OF GIVEN ASCII LETTER NAMES (A TO G)
; AND ACCIDENTALS IN THE PLUS/MINUS FORMAT (-2 TO +2).
; GIVEN: (B) - ASCII LETTER NAME
; (C) - ACCIDENTAL
; RETURNS: (A) - ERROR FLAG
; = 00H IF NO ERRORS FOUND
; = FFH IF ERROR FOUND

DEBC 79 C ALNCHK: LD A,C ; GET ACCIDENTAL
DEBD FE 83 C CP 03H ; COMPARE WITH UPPER BOUND
DEBF 38 04 C JR C,ALCK01 ; JUMP IF VALID
DEC1 FE FE C CP 0FH ; ELSE COMPARE WITH LOWER BOUND
DEC3 38 08 C JR C,ALCK02 ; JUMP IF INVALID
DECC 78 C ALCK01: LD A,B ; GET LETTER NAME
DEC6 D6 41 C SUB 41H ; CONVERT TO 0 TO 6 VALUE
DECB 38 06 C JR C,ALCK02 ; JUMP IF BELOW RANGE
DECA FE 87 C CP 07H ; ELSE COMPARE WITH UPPER BOUND
DEC9 38 02 C JR NC,ALCK02 ; JUMP IF INVALID
DECE AF C XOR A ; LOAD VALID FLAG
DECF C9 C RET
DED0 3E FF C ALCK02: LD A,0FFH ; LOAD INVALID FLAG
DED2 C9 C RET

; KBFOLD
; GET CHARACTER FROM KEYBOARD WITH FOLDOVER FROM LOWER
; CASE TO UPPER CASE. RETURNS WITH ZERO FLAG SET IF
; NO CHARACTER IS PRESENT, RESET IF CHARACTER IS FOUND.
MUSIC CONTROL PROGRAM
MACRO-80 3.44  09-Dec-81  PAGE 1-118

MISCELLANEOUS ROUTINES

; GIVEN:  NONE
; RETURNS: (A) - KEYBOARD CHARACTER
;         = 00H - NO KEY PRESSED

DED3  CD E018  C  KBFOLD: CALL KEYBRD ;USE MONITOR KEYBOARD DRIVER
DED6  C8  C  RET Z ;RETURN IF NO CHARACTER
DED7  FE 61  C  CP 'a' ;TEST FOR LOWER CASE
DED9  08  C  RET C ;EXIT IF NOT
DEDA  FE 7B  C  CP 'z'+1
DED8  02  C  JR NC,KBFLO1
DEDE  D6 20  C  SUB 20H ;CONVERT TO UPPER CASE
DEE0  FE 00  C  KBFLO1: CP 00H ;RESET ZERO FLAG
DEE2  C9  C

; MLTPLY
; MULTIPLY (BC) BY (DE)
; GIVEN: (BC) - MULTIPLIER
;       (DE) - MULTIPLICAND
; RETURNS: (A) - UNCHANGED
; (BC) - CHANGED
; (DE) - PRODUCT

DEE3  F5  C  MLTPLY: PUSH AF
DEE4  E5  C  PUSH HL
DEE5  21 0000  C  LD HL,0000
DEE8  3E 10  C  LD A,10H
DEEA  CB 43  C  MULT01: BIT 0,E
DEEC  2B 01  C  JR Z,MULT02
DEEE  09  C  ADD HL,BC
DEEF  CB 2C  C  MULT02: SRA H
DEF1  CB 1D  C  RR L
DEF3  CB 1A  C  RR D
DEF5  CB 1B  C  RR E
DEF7  3D  C  DEC A
DEF8  2F 00  C  JR NZ,MULT01
DEFA  E1  C  POP HL
DEFB  F1  C  POP AF
DEFCC  C9  C  RET

; DIVIDE
; DIVIDE (DE) BY (BC) AND PUT THE QUOTIENT IN (DE) AND
; THE REMAINDER IN (BC).
MISCELLANEOUS ROUTINES

C
; GIVEN: (BC) - DIVISOR
; (DE) - DIVIDEND
; RETURNS: (BC) - REMAINDER
; (DE) - QUOTIENT

DEFD F5    C  DIVIDE: PUSH AF
DEFE E5    C  PUSH HL
DEFF 21 0000 C  LD HL,0000H
DF02 3E 10   C  LD A,10H
DF04 CB 23   C  DVID01: SLA E
DF06 CB 12   C  RL D
DF08 CB 15   C  RL L
DF0A CB 14   C  RL H
DF0C A7     C  AND A
DF0D ED 42   C  SBC HL,BH
DF0F CB 7C   C  BIT 7,H
DF11 20 03   C  JR NZ,DVID02
DF13 13     C  INC DE
DF14 18 01   C  JR DVID03
DF16 09     C  DVID02: ADD HL,BH
DF17 30     C  DVID03: DEC A
DF18 20 0A   C  JR NZ,DVID01
DF1A 44     C  LD B,H
DF1B 4D     C  LD C,L
DF1C E1     C  POP HL
DF1D F1     C  POP AF
DF1E C9     C  RET

; DOUBLE COMPARE (16-BIT)
; COMPARE (DE) TO (BC)
; GIVEN: (DE),(BC) - VALUES TO BE COMPARED
; RETURNS: (A) - UNCHANGED
; C - CARRY FLAG SET/RESET ACCORDING TO COMPARISON
; Z - ZERO FLAG SET/RESET ACCORDING TO COMPARISON

DF1F 32 0000 X C  DCOMP: LD (TEMP01),A ;SAVE A
DF22 7A C  LD A,D ;COMPARE HIGH ORDER BYTES FIRST
DF23 B8 C  CP B
DF24 20 02 C  JR NZ,DCOMP01
DF26 7B C  LD A,E ;IF SAME COMPARE LOW ORDER BYTE
DF27 B9 C  CP C
DF28 3A 0000 X C  DCOMP01: LD A,(TEMP01) ;RESTORE A BUT NOT FLAGS
DF28 C9 C  RET
.DEPHASE

; ORG CORG+13FAH ;ASSUMES CORG IS XC80H

13FA' C3 A000
13FD' C3 CC00

; JP BASE-2000H ;NARM START
; JP COLD ;COLD START

; SUBTTL SYMBOL TABLE
; END
Symbols:

C0C5 ABSPTT
D077 ACNTNL
D384 ACOT03
DE34 ACK01
DE3F ACK04
D447 ACRO02
CC54 ACTP01
CC5E ACTP05
D9AB ADJUST
00C1 ADSLOL
00C4 ADSSBN
00C2 ADSSTN
00A9 AFLBML
C352 AFLMAJ
00A7 AISBNL
00A8 AISBNL
00AB AFSBLN
00AD AFSBNL
D04C AFSTNL
D33E ALNCHK
004A ANLBNL
C359 ANLTNL
00A5 ANSBNL
00A3 ANSTNL
00B2 ASLMLL
00B4 ASLMRL
00AF ASLTON
00BE ASSBRL
00B8 ASTTLL
00B9 ASTTRN
DC78 AVSM01
D308 AVSM02
CE65 BARLIN
0099 BARLIN
00B9 BARLIN
C44C BASCLF
C476 BASF0C
C36A BFLMAJ
CEA1 BLN03
00BE BSCL11
C5AE BHHLIN
00E1 BSCL12
C50B BHSMNL
D873 CALC32
C0C8 CBASE
DC9E CLD01
D646 CLEF01
C486 CLEF10
0015 CLFL01
C0BE CLR001
C09E CLR002
C0BE CLR003

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD26</td>
<td>CLRPA5</td>
</tr>
<tr>
<td>CD28</td>
<td>CLRPA6</td>
</tr>
<tr>
<td>CD3A</td>
<td>CLRPA7</td>
</tr>
<tr>
<td>CCE</td>
<td>CLRPAR</td>
</tr>
<tr>
<td>C38E</td>
<td>CMAJOR</td>
</tr>
<tr>
<td>C804</td>
<td>CPIDEF</td>
</tr>
<tr>
<td>C808</td>
<td>COLD</td>
</tr>
<tr>
<td>C809</td>
<td>CORG</td>
</tr>
<tr>
<td>C80A</td>
<td>CURPOS</td>
</tr>
<tr>
<td>C80B</td>
<td>CURRT</td>
</tr>
<tr>
<td>C80C</td>
<td>CUPA81</td>
</tr>
<tr>
<td>C80D</td>
<td>CUPA82</td>
</tr>
<tr>
<td>C80E</td>
<td>CUPA83</td>
</tr>
<tr>
<td>C80F</td>
<td>CUPA84</td>
</tr>
<tr>
<td>C810</td>
<td>CUPA85</td>
</tr>
<tr>
<td>C811</td>
<td>CUPA86</td>
</tr>
<tr>
<td>C812</td>
<td>CUPA87</td>
</tr>
<tr>
<td>C813</td>
<td>CUPA88</td>
</tr>
<tr>
<td>C814</td>
<td>CUPA89</td>
</tr>
<tr>
<td>C815</td>
<td>CUPA90</td>
</tr>
<tr>
<td>C816</td>
<td>CUPA91</td>
</tr>
<tr>
<td>C817</td>
<td>CUPA92</td>
</tr>
<tr>
<td>C818</td>
<td>CUPA93</td>
</tr>
<tr>
<td>C819</td>
<td>CUPA94</td>
</tr>
<tr>
<td>C820</td>
<td>CUPA95</td>
</tr>
</tbody>
</table>

**Symbol Table**
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Address</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>C468</td>
<td>FLTCT</td>
<td>C52C</td>
<td>FLTSAN</td>
<td>C530</td>
</tr>
<tr>
<td>C534</td>
<td>FLTSBT</td>
<td>0085</td>
<td>FLTSAN</td>
<td>C538</td>
</tr>
<tr>
<td>C506</td>
<td>MAJOR</td>
<td>DCB7</td>
<td>FRPT01</td>
<td>DCC5</td>
</tr>
<tr>
<td>DCD1</td>
<td>FRPT03</td>
<td>DCDB</td>
<td>FRPT04</td>
<td>DCF5</td>
</tr>
<tr>
<td>DCE8</td>
<td>FRPT06</td>
<td>DCE9</td>
<td>FRPT07</td>
<td>DCF5</td>
</tr>
<tr>
<td>DCF9</td>
<td>FRPT09</td>
<td>DD08</td>
<td>FRPT10</td>
<td>DCAD</td>
</tr>
<tr>
<td>C380</td>
<td>FRTBL</td>
<td>C3E2</td>
<td>FSHMAJ</td>
<td>001A</td>
</tr>
<tr>
<td>DE52</td>
<td>GETCP</td>
<td>DC5A</td>
<td>GETFRQ</td>
<td>CFA8</td>
</tr>
<tr>
<td>DC08</td>
<td>GETNCT</td>
<td>DC4D</td>
<td>GETPIT</td>
<td>C3EE</td>
</tr>
<tr>
<td>C3FA</td>
<td>GMAJOR</td>
<td>C5BC</td>
<td>GRAPHCS</td>
<td>DC61</td>
</tr>
<tr>
<td>CFAE</td>
<td>HTY01</td>
<td>DC23</td>
<td>GTINT01</td>
<td>DC31</td>
</tr>
<tr>
<td>C547</td>
<td>GTINT03</td>
<td>DC4B</td>
<td>GTINT04</td>
<td>DC58</td>
</tr>
<tr>
<td>C4C8</td>
<td>HALFN0</td>
<td>0010</td>
<td>HDRLEN</td>
<td>DE03</td>
</tr>
<tr>
<td>CDE8</td>
<td>HFST02</td>
<td>DE0D</td>
<td>HFST03</td>
<td>C318</td>
</tr>
<tr>
<td>D22F</td>
<td>HI0</td>
<td>D239</td>
<td>HILO01</td>
<td>D24B</td>
</tr>
<tr>
<td>D25C</td>
<td>HILO03</td>
<td>D272</td>
<td>HILO04</td>
<td>D276</td>
</tr>
<tr>
<td>D27A</td>
<td>HILO06</td>
<td>DD05</td>
<td>HLFSTP</td>
<td>0098</td>
</tr>
<tr>
<td>0011</td>
<td>HOLE</td>
<td>DBBE</td>
<td>HSFINT</td>
<td>0FPIK</td>
</tr>
<tr>
<td>0FA8X</td>
<td>INC2</td>
<td>0FAFX</td>
<td>INC3</td>
<td>0FBEX</td>
</tr>
<tr>
<td>C100</td>
<td>INCTBL</td>
<td>0024</td>
<td>INCYH</td>
<td>002C</td>
</tr>
<tr>
<td>DE59</td>
<td>INHIBT</td>
<td>1261X</td>
<td>INHST0</td>
<td>D06E</td>
</tr>
<tr>
<td>D017</td>
<td>INPT01</td>
<td>0FE0X</td>
<td>IOFSET</td>
<td>CB08</td>
</tr>
<tr>
<td>DEEB</td>
<td>KBFL01</td>
<td>DE0D</td>
<td>KBFL01</td>
<td>00FD</td>
</tr>
<tr>
<td>0088X</td>
<td>KEYACD</td>
<td>E918</td>
<td>KEYBRO</td>
<td>C44A</td>
</tr>
<tr>
<td>1215X</td>
<td>KEYLET</td>
<td>0B80X</td>
<td>KEYMOD</td>
<td>CFB1</td>
</tr>
<tr>
<td>C451</td>
<td>KEYSHP</td>
<td>CEE8</td>
<td>KEYSIG</td>
<td>D0AA</td>
</tr>
<tr>
<td>C428</td>
<td>KEYTBL</td>
<td>C328</td>
<td>KEYUPD</td>
<td>C412</td>
</tr>
<tr>
<td>C419</td>
<td>KSSHP</td>
<td>CFBC</td>
<td>KYNX01</td>
<td>CFC8</td>
</tr>
<tr>
<td>CFCC</td>
<td>KYNX03</td>
<td>CFCD</td>
<td>KYNX04</td>
<td>CFDF</td>
</tr>
<tr>
<td>CFE1</td>
<td>KYNX06</td>
<td>CFE9</td>
<td>KYNX07</td>
<td>CF17</td>
</tr>
<tr>
<td>CF25</td>
<td>KYSN02</td>
<td>CFDB</td>
<td>KYSN03</td>
<td>CF3F</td>
</tr>
<tr>
<td>CF4F</td>
<td>KYSN05</td>
<td>CF52</td>
<td>KYSG06</td>
<td>CF53</td>
</tr>
<tr>
<td>D58A</td>
<td>LDGR01</td>
<td>D512</td>
<td>LDGR02</td>
<td>D522</td>
</tr>
<tr>
<td>D527</td>
<td>LDGR04</td>
<td>D4F8</td>
<td>LEDGER</td>
<td>0180</td>
</tr>
<tr>
<td>004A</td>
<td>LINFED</td>
<td>0040</td>
<td>LINLEN</td>
<td>0004</td>
</tr>
<tr>
<td>C308</td>
<td>LWFRO</td>
<td>C31A</td>
<td>LTRTBL</td>
<td>0002</td>
</tr>
<tr>
<td>0005</td>
<td>MAXDEV</td>
<td>0035</td>
<td>MAXFNC</td>
<td>0010</td>
</tr>
<tr>
<td>0049</td>
<td>MAXPIT</td>
<td>C000</td>
<td>MCP</td>
<td>0009</td>
</tr>
<tr>
<td>0010</td>
<td>MINTPT</td>
<td>0066</td>
<td>MINGAM</td>
<td>C079</td>
</tr>
<tr>
<td>C003</td>
<td>MJSCAL</td>
<td>DE32</td>
<td>MLTPLY</td>
<td>C080</td>
</tr>
<tr>
<td>C04F</td>
<td>MNISCAL</td>
<td>DA02</td>
<td>MON01</td>
<td>DA1D</td>
</tr>
<tr>
<td>DA1F</td>
<td>MON03</td>
<td>DA2B</td>
<td>MON04</td>
<td>DA2F</td>
</tr>
<tr>
<td>DA32</td>
<td>MON06</td>
<td>D9F1</td>
<td>MON0PH</td>
<td>D39B</td>
</tr>
<tr>
<td>DEEA</td>
<td>MULT01</td>
<td>DEEF</td>
<td>MULT02</td>
<td>C40A</td>
</tr>
<tr>
<td>C4D4</td>
<td>NATLAN</td>
<td>C4E0</td>
<td>NATLBM</td>
<td>C4EC</td>
</tr>
<tr>
<td>C4E3</td>
<td>NATLM</td>
<td>C4F2</td>
<td>NATLMT</td>
<td>C4FE</td>
</tr>
<tr>
<td>C4F8</td>
<td>NATSN</td>
<td>C504</td>
<td>NATSBN</td>
<td>C516</td>
</tr>
<tr>
<td>C50A</td>
<td>NATSML</td>
<td>C510</td>
<td>NATSMT</td>
<td>D611</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D65B</td>
<td>NCHARS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D65E</td>
<td>NCHAR83</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D65F</td>
<td>NCHAR86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE48</td>
<td>NDEST01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CESA</td>
<td>NDEST04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0C6</td>
<td>NEUTN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00CE</td>
<td>NHBSN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C6</td>
<td>NHNL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C8</td>
<td>NHSTL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0066</td>
<td>NOKEY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0064</td>
<td>NUMSMP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D181</td>
<td>NMTN03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D128</td>
<td>NMTN06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D06F</td>
<td>NXPTR02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D06F</td>
<td>NXPTR05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0CE</td>
<td>NXPTR11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D36B</td>
<td>OCAD03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D081</td>
<td>OCTAVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0016</td>
<td>OORANG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0138</td>
<td>PARBIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD3E</td>
<td>PARMSK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008X</td>
<td>PART83</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008X</td>
<td>PART86</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008X</td>
<td>PART89</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008X</td>
<td>PART12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008X</td>
<td>PART15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CF64</td>
<td>PATC01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CF57</td>
<td>PATCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D35E</td>
<td>PITA01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D378</td>
<td>PITA04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D391</td>
<td>PITA07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D391</td>
<td>PITA08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D319</td>
<td>PITCHI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D32A</td>
<td>PITCHU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D317</td>
<td>PITL02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA48</td>
<td>PLAKEY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D45C</td>
<td>PLKY02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA7D</td>
<td>PLKY05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAB1</td>
<td>PLRL01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1EA</td>
<td>PRBR01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D22B</td>
<td>PRBR07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD54</td>
<td>PRMK03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1A8</td>
<td>PRSC03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1D7</td>
<td>PRST03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1E1</td>
<td>PRTEXT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C478</td>
<td>PTCM4D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D208</td>
<td>PTCM4T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Address</td>
<td>Description</td>
<td>Address</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>0FD2X</td>
<td>PTR1</td>
<td>0FA4X</td>
<td>PTR2</td>
<td>0FB3X</td>
</tr>
<tr>
<td>0FC2X</td>
<td>PTR4</td>
<td>D298</td>
<td>PTST01</td>
<td>D2AD</td>
</tr>
<tr>
<td>D2E4</td>
<td>PTUT01</td>
<td>D02A</td>
<td>PTXT01</td>
<td>D039</td>
</tr>
<tr>
<td>D046</td>
<td>PTXT03</td>
<td>D04E</td>
<td>PTXT04</td>
<td>D05B</td>
</tr>
<tr>
<td>D06F</td>
<td>PTXT06</td>
<td>D073</td>
<td>PTXT07</td>
<td>D081</td>
</tr>
<tr>
<td>D092</td>
<td>PTXT09</td>
<td>D0A0</td>
<td>PTXT10</td>
<td>D0BB</td>
</tr>
<tr>
<td>C4GA</td>
<td>QUARTR</td>
<td>CC60</td>
<td>RETURN</td>
<td>C062</td>
</tr>
<tr>
<td>C8CD</td>
<td>RHYTBL</td>
<td>CFF5</td>
<td>RSKY01</td>
<td>CFFE</td>
</tr>
<tr>
<td>CFE0</td>
<td>RSTKEY</td>
<td>DE64</td>
<td>SACALC</td>
<td>DE7E</td>
</tr>
<tr>
<td>088EX</td>
<td>SAFBAR</td>
<td>0F89X</td>
<td>SAMP32</td>
<td>1111X</td>
</tr>
<tr>
<td>088X</td>
<td>SAMP02</td>
<td>1024X</td>
<td>SAMPL3</td>
<td>1032X</td>
</tr>
<tr>
<td>0862X</td>
<td>SAVEIX</td>
<td>0866X</td>
<td>SAVI1Y</td>
<td>D718</td>
</tr>
<tr>
<td>D733</td>
<td>SCHRO1</td>
<td>D73F</td>
<td>SCHRO2</td>
<td>D74B</td>
</tr>
<tr>
<td>D754</td>
<td>SCHRO4</td>
<td>D75F</td>
<td>SCHRO5</td>
<td>D771</td>
</tr>
<tr>
<td>D774</td>
<td>SCHRO7</td>
<td>D77F</td>
<td>SCHRN1</td>
<td>F080</td>
</tr>
<tr>
<td>DE4B</td>
<td>SETCP</td>
<td>C6A5</td>
<td>SETKEY</td>
<td>D5DE</td>
</tr>
<tr>
<td>D611</td>
<td>SETP0</td>
<td>D630</td>
<td>SHARP</td>
<td>C544</td>
</tr>
<tr>
<td>C53C</td>
<td>SHPLAN</td>
<td>C54C</td>
<td>SHPLBL</td>
<td>C55E</td>
</tr>
<tr>
<td>C554</td>
<td>SHPLUL</td>
<td>C564</td>
<td>SHPLMT</td>
<td>C458</td>
</tr>
<tr>
<td>C574</td>
<td>SHPSAL</td>
<td>C56C</td>
<td>SHPSAN</td>
<td>C57E</td>
</tr>
<tr>
<td>C584</td>
<td>SHPSNL</td>
<td>D63D</td>
<td>SHPS01</td>
<td>D647</td>
</tr>
<tr>
<td>D64C</td>
<td>SHPS03</td>
<td>D654</td>
<td>SHPS04</td>
<td>D884</td>
</tr>
<tr>
<td>D887</td>
<td>SING81</td>
<td>D88C</td>
<td>SING82</td>
<td>D806</td>
</tr>
<tr>
<td>C4CE</td>
<td>SIXTEN</td>
<td>0803</td>
<td>SNGDOT</td>
<td>D4DC</td>
</tr>
<tr>
<td>DAF8</td>
<td>SNTN01</td>
<td>D810</td>
<td>SNTN02</td>
<td>D815</td>
</tr>
<tr>
<td>D824</td>
<td>SNTN04</td>
<td>D829</td>
<td>SNTN05</td>
<td>D838</td>
</tr>
<tr>
<td>D83D</td>
<td>SNTN07</td>
<td>D84C</td>
<td>SNTN06</td>
<td>D851</td>
</tr>
<tr>
<td>D849</td>
<td>SNTN10</td>
<td>D828</td>
<td>SPACE</td>
<td>D3F9</td>
</tr>
<tr>
<td>D484</td>
<td>STAD02</td>
<td>D412</td>
<td>STAD03</td>
<td>D416</td>
</tr>
<tr>
<td>D4A1</td>
<td>STAD05</td>
<td>CDAC</td>
<td>STAF01</td>
<td>CDBA</td>
</tr>
<tr>
<td>D880</td>
<td>STAFF</td>
<td>CC80</td>
<td>START</td>
<td>CD7C</td>
</tr>
<tr>
<td>D356</td>
<td>STFADJ</td>
<td>C535</td>
<td>STFCLF</td>
<td>C404</td>
</tr>
<tr>
<td>CEC9</td>
<td>STKY01</td>
<td>CEE4</td>
<td>STKY02</td>
<td>D540</td>
</tr>
<tr>
<td>D52A</td>
<td>STORE</td>
<td>08EF</td>
<td>STPORT</td>
<td>D82F</td>
</tr>
<tr>
<td>D88C</td>
<td>STXP02</td>
<td>D845</td>
<td>STXP03</td>
<td>D859</td>
</tr>
<tr>
<td>D85D</td>
<td>STXP05</td>
<td>D86B</td>
<td>STXP06</td>
<td>D870</td>
</tr>
<tr>
<td>D873</td>
<td>STXP08</td>
<td>D882</td>
<td>STXP09</td>
<td>D891</td>
</tr>
<tr>
<td>D899</td>
<td>STXP11</td>
<td>D89C</td>
<td>STXP12</td>
<td>D89E</td>
</tr>
<tr>
<td>D89F</td>
<td>STXP14</td>
<td>D8A6</td>
<td>STXP15</td>
<td>C337</td>
</tr>
<tr>
<td>0889</td>
<td>TAB</td>
<td>1329X</td>
<td>TEMP01</td>
<td>0651X</td>
</tr>
<tr>
<td>0F77X</td>
<td>TEMPO</td>
<td>C450</td>
<td>THRITY2</td>
<td>D0BD</td>
</tr>
<tr>
<td>C4BE</td>
<td>TRCLCF</td>
<td>C468</td>
<td>TRBF0C</td>
<td>C47C</td>
</tr>
<tr>
<td>C45B</td>
<td>TRBSGC</td>
<td>0880</td>
<td>TRCL12</td>
<td>0801</td>
</tr>
<tr>
<td>0882</td>
<td>TRCL23</td>
<td>0883</td>
<td>TRCL32</td>
<td>0884</td>
</tr>
<tr>
<td>0885</td>
<td>TRCL42</td>
<td>0886</td>
<td>TRCL43</td>
<td>0887</td>
</tr>
<tr>
<td>0888</td>
<td>TRCL52</td>
<td>0889</td>
<td>TRCL53</td>
<td>088A</td>
</tr>
<tr>
<td>0888</td>
<td>TRCL62</td>
<td>088C</td>
<td>TRCL63</td>
<td>088D</td>
</tr>
<tr>
<td>D784</td>
<td>TRNPOL</td>
<td>D7F1</td>
<td>TRNP01</td>
<td>D7C6</td>
</tr>
<tr>
<td>Symbol</td>
<td>Meaning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D7CA</td>
<td>TRPL02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D7DF</td>
<td>TRPL03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00FF</td>
<td>TTDBAR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>TTEN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>TUNEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00A0</td>
<td>TUNI2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>UNAPIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0077</td>
<td>UNALT1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0097</td>
<td>UNALT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4C2</td>
<td>WHCHR1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4CA</td>
<td>WHCHR2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4CB</td>
<td>WHCHR3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D4AA</td>
<td>WHCHR4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0087</td>
<td>WHCHR5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00A0</td>
<td>WHCHR6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0097</td>
<td>WHCHR7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0105</td>
<td>WHCHR8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D12D</td>
<td>WHCHR9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D13F</td>
<td>WHCHR10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D152</td>
<td>WHCHR11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D165</td>
<td>WHCHR12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0EF0X</td>
<td>WHCHR13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08C0X</td>
<td>WHCHR14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE81</td>
<td>WHCHR15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE99</td>
<td>WHCHR16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>WHCHR17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No Fatal error(s)
APPENDIX B

LISTING OF SIGHTSINGER'S MIRROR
TITLE SIGHTSINGER'S MIRROR
SUBTTL RAM DATA SECTION

; RAM DATA STORAGE AREA
;
.
0000
DSEG
ORG 0000H
.
.PHASE 9000H

9000 TOPRAM EQU $ ; EFFECTIVE TOP OF RAM

; GLOBAL TOPRAM
GLOBAL TUNGRP
GLOBAL DATE
GLOBAL USRNAME
GLOBAL RECENT
GLOBAL TUNRING
GLOBAL TUNLOM
GLOBAL TUNHI
GLOBAL ATTEMPT
GLOBAL NUMSNG
GLOBAL ERRCNT
GLOBAL SCORE
GLOBAL NUMPLA
GLOBAL TUNCNT
GLOBAL USRRANG
GLOBAL USRLOW
GLOBAL USRHI
GLOBAL USRRAT
GLOBAL IDFSET
GLOBAL KEYN3
GLOBAL KEYN3F
GLOBAL PARNUM
GLOBAL FHMLEF
GLOBAL SAVCMD
GLOBAL SEED1
GLOBAL SEED2
GLOBAL DIGITS
GLOBAL PREVIP
GLOBAL TEMPCT
SIGHTSINGER'S MIRROR  MACRO-80 3.44  09-Dec-81  PAGE 1-1
RAM DATA SECTION

GLOBAL TEMPSO
GLOBAL TEMPO1
GLOBAL TEMPO2
GLOBAL TMPMSG
GLOBAL WRKSPC
GLOBAL LOGTBL

_BASE 8

.TAB 16

0800 TUNGRP EQU 0100H  "LOAD ADDRESS OF TUNE GROUPS"

9000 DATE:  DEFS  3  "CURRENT DATE"

9003 0000 USRNAME: DEFW 0000H  "ADDRESS OF USER NAME IN LOG TABLE"

9005 0000 RECEND: DEFW 0000H  "ADDRESS OF NEXT POSITION IN BUFFER"

9007 00 TUNRNG: DEFB 00H  "RANGE OF CURRENT TUNE"

9008 00 TUNLOH: DEFB 00H  "LOWEST PITCH IN CURRENT TUNE"

9009 00 TUNHI: DEFB 00H  "HIGHEST PITCH IN CURRENT TUNE"

900A 00 ATTEMPT: DEFB 00H  "NUMBER OF PERFORMANCE ATTEMPT OF TUNE"

900B 00 NUMSNG: DEFB 00H  "NUMBER OF NOTES SUNG IN TUNE"

900C 00 ERRCNT: DEFB 00H  "NUMBER OF PITCHES SUNG IN ERROR"

900D 00 SCORE: DEFB 00H  "PERCENT CORRECT"

900E 00 NUMPLA: DEFB 00H  "NUMBER OF TIMES TUNE WAS PLAYED"

900F 00 TUCNT: DEFB 00H  "COUNT OF TUNES SELECTED"

; THE FOLLOWING THREE FIELDS MUST BE CONTIGUOUS AND
; IN THE ORDER GIVEN.

9010 00 USRANG: DEFB 00H  "USER'S VOCAL RANGE"

9011 00 USRLOH: DEFB 00H  "USER'S LOWEST PITCH"

9012 00 USRHI: DEFB 00H  "USER'S HIGHEST PITCH"

9013 00 USRRAT: DEFB 00H  "USER'S CURRENT DIFFICULTY RATING"

9014 00 IOFSET: DEFB 00H  "INTEGER OFFSET VALUE TO BE ADDED TO
SUNG PITCHES TO ADJUST THEM FOR THE
RANGE OF THE USER"

9015 00 KEYNT3: DEFB 00H  "INTEGER VALUE OF KEY NOTE IN OCTAVE 3"

9016 00 KEYNOF: DEFB 00H  "INTEGER VALUE OF KEY NOTE WITH IOFSET
ADDED TO IT"

9017 00 PARTNUM: DEFB 00H  "CURRENT PARTITION NUMBER IN FREE MODE"

9018 00 FMCLEF: DEFB 00H  "CLEF IN USE IN FREE MODE"
RAM DATA SECTION

; SAVE MEMORY TO TAPE COMMAND LINE

9019 20  ; SAVCHD: DEFB 20H  ; SPACE
901A 05  ; DEFtram NAME
901F 20  ; SPACE
9020 04  ; FROM ADDRESS
9024 20  ; SPACE
9025 04  ; TO ADDRESS
9029 20  ; SPACE
902A 00  ; TAPE UNIT NUMBER
902B 0D  ; CARRIAGE RETURN

902C 1936  ; SEED1: DEFW 1936H  ; RANDOM NUMBER STARTING SEEDS
902D 1936  ; SEED2: DEFW 1936H

9030 00  ; DIGITS: DEFS 5  ; STORAGE OF BINARY-TO-ASCII RESULT

9035 00  ; PREVIP: DEFB 00H  ; PREVIOUS INTEGER PITCH

9036 00  ; TEMPCP: DEFB 00H  ; TEMPORARY CURRENT POSITION SAVE
9037 00  ; TEMPSO: DEFB 00H  ; TEMPORARY STAFF ORIENTATION SAVE

9038 00  ; TEMPO1: DEFS 2  ; GENERAL TEMPORARY STORAGE
9039 00  ; TEMPO2: DEFS 2

903C 00  ; TMPMSG: DEFS 00H  ; SPACE FOR MESSAGES WITH VOLATILE PARTS

909C 100H  ; WKSPC: DEFS 100H

; DEPHASE

ORG $8100H-LOH($)

; PHASE WKSPC+0200H-LOH(WKSPC)

9200 100H  ; RECBUF: DEFS 100H

9308 300H  ; LOGTBL: DEFS 300H

; DEPHASE

END
### Macros:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>908AI</td>
<td>ATEMPT</td>
<td>90031I</td>
<td>DATE</td>
</tr>
<tr>
<td>908Ci</td>
<td>ERRCNT</td>
<td>90815I</td>
<td>DATE</td>
</tr>
<tr>
<td>90816I</td>
<td>KEYNOF</td>
<td>90815I</td>
<td>KEYNT3</td>
</tr>
<tr>
<td>908Ei</td>
<td>NUMPLA</td>
<td>9088Bi</td>
<td>NUMSNQ</td>
</tr>
<tr>
<td>90835I</td>
<td>PREVIP</td>
<td>9200I</td>
<td>RECBUF</td>
</tr>
<tr>
<td>90819I</td>
<td>SAVCMD</td>
<td>9080Di</td>
<td>SCORE</td>
</tr>
<tr>
<td>9082EI</td>
<td>SEED2</td>
<td>90838I</td>
<td>TEMPO1</td>
</tr>
<tr>
<td>90836I</td>
<td>TEMPCP</td>
<td>90837I</td>
<td>TEMPSO</td>
</tr>
<tr>
<td>9088I</td>
<td>TOPRAM</td>
<td>9080FI</td>
<td>Tuncnt</td>
</tr>
<tr>
<td>9089I</td>
<td>TUNHI</td>
<td>90888I</td>
<td>TUNLOW</td>
</tr>
<tr>
<td>90818i</td>
<td>USRANG</td>
<td>90812I</td>
<td>USRHI</td>
</tr>
<tr>
<td>90831I</td>
<td>USRNAD</td>
<td>90813I</td>
<td>USBRAT</td>
</tr>
</tbody>
</table>

### Symbols:

- 988A  IATEMPT
- 980C  IERRCNT
- 9816I  KEYNOF
- 988E  NUMPLA
- 9835I  PREVIP
- 9819I  SAVCMD
- 982E  SEED2
- 9836I  TEMPCP
- 9888I  TOPRAM
- 9889I  TUNHI
- 98818I USRANG
- 98831I USRNAD

No Fatal error(s)
TITLE SIGHTSINGER'S MIRROR
SUBTL EXTERNAL NAMES SECTION

; SIGHTSINGER'S MIRROR
; A SYSTEM FOR THE REAL-TIME TRANSCRIPTION OF MONOPHONIC
; PITCHES TO MUSIC NOTATION AS AN AID IN SIGHTSINGING.
; VERSION 1.0
; OCTOBER 1983
; RANDY KOLB

.200

0000' DSEG
ORG 0000H

0000* DORG EQU $
A000 BASE EQU 0A000H

.PHASE BASE

A000 DBASE EQU $

MACLIB B:SSMEXT.MAC

; EXTERNAL NAMES TABLE

EXTERNAL TOPRAN
EXTERNAL TUNGRP
EXTERNAL DATE
EXTERNAL USRNAD
EXTERNAL RECEND
EXTERNAL TUNRNG
EXTERNAL TUNLOH
EXTERNAL TUNHI
EXTERNAL ATEMPT
EXTERNAL NUMSNG
EXTERNAL ERRONT
EXTERNAL SCORE
EXTERNAL NUMPLA
EXTERNAL TUNONT
EXTERNAL USRANG
EXTERNAL NAMES SECTION

C  EXTERNAL  USRLOW
C  EXTERNAL  USRHI
C  EXTERNAL  USRRAT
C  EXTERNAL  IOFSET
C  EXTERNAL  KEYNT3
C  EXTERNAL  KEYNOF
C  EXTERNAL  PARNUM
C  EXTERNAL  FMCLEF
C  EXTERNAL  SAUCMD
C  EXTERNAL  SEED1
C  EXTERNAL  SEED2
C  EXTERNAL  DIGITS
C  EXTERNAL  PREVIP
C  EXTERNAL  TEMPCP
C  EXTERNAL  TEMPS0
C  EXTERNAL  TEMPO1
C  EXTERNAL  TEMPO2
C  EXTERNAL  TMPMSG
C  EXTERNAL  WRKSPC
C  EXTERNAL  RECBUF
C  EXTERNAL  LOGTBL

;

SUBTTL EQUATE SECTION
@EJECT
MACLIB B:ASCII.MAC

<table>
<thead>
<tr>
<th>hex</th>
<th>character</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>NUL</td>
</tr>
<tr>
<td>0001</td>
<td>SOH</td>
</tr>
<tr>
<td>0002</td>
<td>STX</td>
</tr>
<tr>
<td>0003</td>
<td>ETX</td>
</tr>
<tr>
<td>0004</td>
<td>EOT</td>
</tr>
<tr>
<td>0005</td>
<td>ENQ</td>
</tr>
<tr>
<td>0006</td>
<td>ACK</td>
</tr>
<tr>
<td>0007</td>
<td>BEL</td>
</tr>
<tr>
<td>0008</td>
<td>BS</td>
</tr>
<tr>
<td>0009</td>
<td>HT</td>
</tr>
<tr>
<td>000A</td>
<td>LF</td>
</tr>
<tr>
<td>000B</td>
<td>VT</td>
</tr>
<tr>
<td>000C</td>
<td>FF</td>
</tr>
<tr>
<td>000D</td>
<td>CR</td>
</tr>
<tr>
<td>000E</td>
<td>SO</td>
</tr>
<tr>
<td>000F</td>
<td>SI</td>
</tr>
<tr>
<td>0010</td>
<td>DLE</td>
</tr>
<tr>
<td>0011</td>
<td>DC1</td>
</tr>
<tr>
<td>0012</td>
<td>DC2</td>
</tr>
<tr>
<td>0013</td>
<td>DC3</td>
</tr>
<tr>
<td>0014</td>
<td>DC4</td>
</tr>
<tr>
<td>0015</td>
<td>NAK</td>
</tr>
<tr>
<td>0016</td>
<td>SYN</td>
</tr>
<tr>
<td>0017</td>
<td>ETB</td>
</tr>
<tr>
<td>0018</td>
<td>CAN</td>
</tr>
<tr>
<td>0019</td>
<td>EM</td>
</tr>
<tr>
<td>001A</td>
<td>SUB</td>
</tr>
<tr>
<td>001B</td>
<td>ESC</td>
</tr>
<tr>
<td>001C</td>
<td>FS</td>
</tr>
<tr>
<td>001D</td>
<td>GS</td>
</tr>
<tr>
<td>001E</td>
<td>RS</td>
</tr>
<tr>
<td>001F</td>
<td>US</td>
</tr>
<tr>
<td>0020</td>
<td>SPACE</td>
</tr>
<tr>
<td>0021</td>
<td></td>
</tr>
</tbody>
</table>
EQUATE SECTION

C ; EQUATE VALUES
C ;

C800
C MCP EQU BASE+2000H ; MCP ENTRY POINT ADDRESS
C ;
C ; SORCERER MONITOR ADDRESSES
C ;
E003
C MONITR EQU 0E003H ; WARM START; RETURN TO SORCERER MONITOR
E1A2
C FNMONA EQU 0E1A2H ; FIND MONITOR WORK AREA ADDRESS IN (IY)
E1A2
C SYNC EQU 0E1A2H ; SYNCHRONIZE WITH VIDEO OUTPUT
E018
C KEYBD EQU 0E018H ; KEYBOARD INPUT
E018
C VIDEO EQU 0E018H ; VIDEO OUTPUT
E14D
C MONINP EQU 0E14DH ; INPUT LINE FROM KEYBOARD TO BUFFER
E099
C GETCHR EQU 0E009H ; GET CHARACTER FROM CURRENT INPUT DEVICE
E631
C SETINP EQU 0E631H ; SET ADDRESS OF INPUT DEVICE DRIVER
E615
C SETOUT EQU 0E615H ; SET ADDRESS OF OUTPUT DEVICE DRIVER
E18A
C PRTRN EQU 0E18AH ; PRINT TEXT STRING ON SCREEN
EYCC
C MoveCUR EQU 0EYCC : MOVE CURSOR TO CURRENT POSITION
E638
C TAPSAV EQU 0E638H ; SAVE MEMORY TO TAPE
E7B8
C TAPLODE EQU 0E7B8H ; LOAD INTO MEMORY FROM TAPE
E205
C CRLF EQU 0E205H ; PRINT CARRIAGE RETURN/LINE FEED
F008
C ENDRAM EQU 0F008H ; STORAGE FOR TOP OF RAM ADDRESS
C ;
C ; DATA
C ;
F008
C SCREEN EQU 0F008H ; STARTING ADDRESS OF SORCERER SCREEN
0040
C LINLEN EQU 40H ; LENGTH OF SCREEN LINE
0012
C HDLEN EQU 12H ; LENGTH OF TUNE HEADER
0020
C TGHLEN EQU 20H ; LENGTH OF TUNE GROUP HEADER
0008
C PDBLEN EQU 80H ; PARTITION DEFINITION BLOCK LENGTH
0005
C NOTLEN EQU 05H ; LENGTH OF NOTE ENTRY
00A8
C TYPSCLE EQU 0A8H ; TUNE TYPE OF SCALE DEGREE/MODIFIER
0080
C TYPLET EQU 080H ; TUNE TYPE OF LETTER NAME/ACCIDENTAL
0001
C WILNOT EQU 01H ; TUNE TABLE VALUE FOR WHOLE NOTE
0000
C BLWNH EQU 00H ; BLACK WHOLE NOTE RHYTHMIC VALUE
007F
C FLDLFLG EQU 7FH ; FOLD LOWER CASE TO UPPER CASE FLAG
0000
C NOFOLD EQU 00H ; DO NOT FOLD LOWER TO UPPER CASE FLAG
0000
C NHPL EQU 02H ; NOTES PER MEASURE IN FREE MODE
C ;
C ; MCP RETURN CODES AND VALUE ASSIGNMENTS
C ;
0000
C EOSTRG EQU 00H ; END OF TEXT STRING
0016
C OORANG EQU 16H ; OUT OF RANGE
0017
C EOM EQU 17H ; END OF MEASURE
SIGHTSINGER'S MIRROR
EQUATE SECTION

MACRO-80 3.44 09-Dec-81

PAGE 1-4

0018 C EOS EQU 18H ;END OF STAFF
0019 C EOP EQU 19H ;END OF PARTITION NUMBER LIST
001A C EOTUNE EQU 1AH ;END OF TUNE
00FD C KBITR EQU 0FDH ;KEYBOARD INTERRUPT INDICATOR
00FE C SNGBAR EQU 0FEH ;SINGLE BAR INDICATOR
00FF C DBLBAR EQU 0FFH ;DOUBLE BAR (END OF TUNE) INDICATOR
00FF C NOISE EQU 0FFH ;SOUND INPUT NOISE VALUE

; TAPE RECORD TYPES

0018 C TYPUSR EQU 00H ;USER RECORD TYPE INDICATOR
0088 C TYPGRP EQU 08H ;TUNE GROUP RECORD TYPE INDICATOR
0100 C TYPNUN EQU 10H ;TUNE RECORD TYPE INDICATOR
0200 C TYPPER EQU 20H ;PERFORMANCE RECORD TYPE INDICATOR

; SPECIAL CHARACTERS

00FE C SMFLT EQU 0FEH ;VALUE FOR SMALL FLAT CHARACTER

; SINGLE-BIT ASSIGNMENTS

0007 C USEDBT EQU 7 ;HIGH ORDER BIT SET INDICATES TUNE ALREADY USED
0006 C ERRBIT EQU 6 ;INDICATES ERROR FOUND IN TUNE WHEN HIGH

; TUNE GROUP HEADER INDEXES

0000 C GRPNAME EQU 0 ;NAME OF TUNE GROUP
0005 C GRPTYP EQU 5 ;TYPE OF TUNE GROUP
0006 C SELSCH EQU 6 ;TUNE SELECTION SCHEME INDICATOR
0007 C RATONE EQU 7 ;STARTING RATING VALUE
0008 C XPOSFL EQU 8 ;TRANSPOSITION INDICATOR
0009 C ADDLIM EQU 9 ;LIMIT ON ACCIDENTALS IN KEY SIGNATURE
00A0 C NUMTUN EQU 0AH ;NUMBER OF TUNES IN GROUP
000B C GRPEND EQU 0BH ;END OF GROUP ADDRESS

; TUNES HEADER INDEXES

0000 C LINSCH EQU 00H ;LINEAR SELECTION SCHEME INDICATOR VALUE
0008 C RATSCH EQU 08H ;RATING SELECTION SCHEME INDICATOR VALUE
0010 C RNDSCH EQU 10H ;RANDOM SELECTION SCHEME INDICATOR VALUE

; TUNE IDENTIFIER

0008 C TUNLEN EQU 0 ;LENGTH OF TUNE AND HEADER
0002 C TUNEID EQU 2 ;TUNE IDENTIFIER
0003 C TunTyp EQU 3 ;TUNE TYPE
SIGHTSINGER'S MIRROR

MACRO-88 3.44 09-Dec-81

EQUATE SECTION

0004  C  TUNCLF EQU  4  ;CLEF
0005  C  KEYLET EQU  5  ;KEY LETTER IN ASCII
0006  C  KEYACD EQU  6  ;KEY ACCIDENTAL
0007  C  KEYMOD EQU  7  ;KEY MODE
0008  C  RATING EQU  8  ;DIFFICULTY RATING
0009  C  BITFLG EQU  9  ;BIT FLAGS
000A  C  RNGTUN EQU 0AH ;RANGE OF TUNE
000B  C  NUMNOT EQU 0BH ;NUMBER OF NOTES IN TUNE

000C  C  SCLET EQU 00H ;SCALE DEGREE OR LETTER NAME
000D  C  MODACD EQU 01H ;MODIFIER OR ACCIDENTAL
000E  C  OCTAVE EQU 02H ;OCTAVE
000F  C  RHYBYT EQU 03H ;RHYTHM
0010  C  CLEF EQU 06H ;CURRENT POSITION INCREMENT

0011  C  FILEN EQU 01H ;FILE NAME
0012  C  SAUFRM EQU 07H ;FROM ADDRESS
0013  C  SAUTO EQU 08H ;TO ADDRESS
0014  C  UNITNO EQU 11H ;TAPE UNIT NUMBER

MACLIB B:MCPCALL.MAC

0015  C  DEFPAR EQU 00H
0016  C  ACTPAR EQU 01H
0017  C  CLRPAR EQU 02H
0018  C  CURPAR EQU 03H
0019  C  STFCLF EQU 04H
001A  C  STAFF EQU 05H
001B  C  CLEF EQU 06H
001C  C  CLOSTF EQU 07H
001D  C  BARLIN EQU 08H
001E  C  SETKEY EQU 09H
001F  C  KEYSIG EQU 0AH
0020  C  GETKEY EQU 0BH
0021  C  RSTKEY EQU 0CH
0022  C  TIMSIG EQU 0DH
0023  C  SETXPO EQU 0EH
SIGHTSINGER'S MIRROR

EQUATE SECTION

`000F C NEWTUN EQU 0FH
0010 C XLATUN EQU 10H
0011 C PRSCR EQU 11H
0012 C PRSTF EQU 12H
0013 C PRTEST EQU 13H
0014 C PTCHST EQU 14H
0015 C PTCHLT EQU 15H
0016 C PTCHUT EQU 16H
0017 C PITCHI EQU 17H
0018 C ERANOT EQU 18H
0019 C ERACOL EQU 19H
001A C ACDNTL EQU 1AH
001B C TRNPOL EQU 1BH
001C C CVTSTL EQU 1CH
001D C CVTLTS EQU 1DH
001E C CVTLTA EQU 1EH
001F C CVTUTL EQU 1FH
0020 C CVTTAT EQU 20H
0021 C CVTTAT EQU 21H
0022 C MONOPH EQU 22H
0023 C PLAKEY EQU 23H
0024 C PLAREL EQU 24H
0025 C SINGTUN EQU 25H
0026 C SETOFS EQU 26H
0027 C NXTPIT EQU 27H
0028 C GETNOT EQU 28H
0029 C GETPIT EQU 29H
002A C GETFRQ EQU 2AH
002B C FROPIT EQU 2BH
002C C INITPT EQU 2CH
002D C HSFRTN EQU 2DH
002E C HILO EQU 2EH
002F C ACDNRY EQU 2FH
0030 C SETCP EQU 30H
0031 C GETCP EQU 31H
0032 C INHIBT EQU 32H
0033 C ENASTO EQU 33H
0034 C PRTEXT EQU 34H`

;`

SUBTTL ROM DATA SECTION

@EJECT
MACLIB B:SSMRROM.MAC

; ROM DATA STORAGE AREA

; PARTITION DEFINITIONS

PART01: DEFB 01,01,01H,40H,1EH,00,00,00
PART02: DEFB 02,01,01H,40H,0DH,04,00,00
PART03: DEFB @3,0,0EH,40H,8DH,04,00,00
PART04: DEFB 04,0,1BH,40H,04H,08,00,00
PART05: DEFB 05,0,0EH,40H,11H,00,00,00

; PARTITION LIST FOR DISPLAY OF TUNES
PARLST: DEFB 82H,0FFH

; SPECIAL GRAPHIC CHARACTER DEFINITION
SPCHAR: DEFB 48H,48H,40H,58H ; DEFINITION OF SMALL FLAT
       DEFB 68H,48H,58H,60H ; FOR KEY SIGNATURES

; RELATIVE INTEGER PITCH OFFSETS FOR MAJOR AND MINOR TRIADS
MJTRAD: DEFB 00H,04H,07H,04H,00H ; INTEGER OFFSETS FOR A MAJOR TRIAD UP AND DOWN
MNTRAD: DEFB 00H,03H,07H,03H,00H ; INTEGER OFFSETS FOR A MINOR TRIAD UP AND DOWN

; TONIC NOTE IN 'PLAREL' FORMAT
A83F  01  C  ONENOT: DEFB 01H ; NUMBER OF NOTES
A840  00 00 00 00  C  DEFB 00,00,00,00,04
A844  04  C  
C  ;
A845  05  C  RMJ135: DEFB 05H ; NUMBER OF NOTES
A846  00 00 00 00  C  DEFB 00,00,00,00,04 ; 1
A84A  04  C  
A84B  04 04 04 04  C  DEFB 04,04,04,04,04 ; 3
A84F  04  C  
A850  07 07 07 07  C  DEFB 07,07,07,07,04 ; 5
A854  04  C  
A855  04 04 04 04  C  DEFB 04,04,04,04,04 ; 3
A857  04  C  
A85A  00 00 00 00  C  DEFB 00,00,00,00,04 ; 1
A85E  04  C  
A85F  05  C  RMN135: DEFB 05H ; NUMBER OF NOTES
A860  00 00 00 00  C  DEFB 00,00,00,00,04 ; 1
A864  04  C  
A865  03 03 03 03  C  DEFB 03,03,03,03,04 ; 3
A869  04  C  
A86A  07 07 07 07  C  DEFB 07,07,07,07,04 ; 5
A86E  04  C  
A86F  03 03 03 03  C  DEFB 03,03,03,03,04 ; 3
A873  04  C  
A874  00 00 00 00  C  DEFB 00,00,00,00,04 ; 1
A878  04  C  
A879  00 00  C  MSGNUL: DEFB 00H,00H
A87B  4D 43 50 20  C  MSGMCP: DEFM MCP not loaded'
A87F  6E 6F 74 20  C  
A883  6C 6F 61 64  C  
A887  65 64  C  
A88B  00  C  DEFB 00H
A899  08 0A  C  MSGPET: DEFB 00H,00AH
A89C  45 6E 74 65  C  DEFM 'Enter today's date in the form mm/dd/yy.'
A89F  72 20 74 6F  C  

MESSAGES

C  ;
C  ;
C  ;
C  ;
C  ;
C  ;
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A094</td>
<td>64 61 79 27 C</td>
<td>DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A095</td>
<td>73 28 64 61 C</td>
<td>'Use leading zeros if necessary.'</td>
</tr>
<tr>
<td>A096</td>
<td>6E 28 74 68 C</td>
<td></td>
</tr>
<tr>
<td>A097</td>
<td>65 20 66 6F C</td>
<td></td>
</tr>
<tr>
<td>A098</td>
<td>72 6D 20 60 C</td>
<td></td>
</tr>
<tr>
<td>A099</td>
<td>6D 2F 64 64 C</td>
<td></td>
</tr>
<tr>
<td>A0A0</td>
<td>2F 79 79 2E C</td>
<td></td>
</tr>
<tr>
<td>A0A4</td>
<td>80 8A C</td>
<td></td>
</tr>
<tr>
<td>A0A5</td>
<td>55 73 65 20 C</td>
<td>DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A0A6</td>
<td>6C 65 61 64 C</td>
<td>'For example: 09/38/83'</td>
</tr>
<tr>
<td>A0A7</td>
<td>69 6E 67 20 C</td>
<td></td>
</tr>
<tr>
<td>A0C2</td>
<td>7A 6D 72 6F C</td>
<td></td>
</tr>
<tr>
<td>A0C5</td>
<td>73 28 69 66 C</td>
<td></td>
</tr>
<tr>
<td>A0CA</td>
<td>28 6E 65 63 C</td>
<td></td>
</tr>
<tr>
<td>A0CE</td>
<td>65 73 73 61 C</td>
<td></td>
</tr>
<tr>
<td>A0D2</td>
<td>72 79 2E C</td>
<td></td>
</tr>
<tr>
<td>A0D5</td>
<td>80 8A C</td>
<td></td>
</tr>
<tr>
<td>A0D6</td>
<td>46 6F 72 20 C</td>
<td>DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A0D7</td>
<td>65 70 61 60 C</td>
<td>'Date: '</td>
</tr>
<tr>
<td>A0DB</td>
<td>7C 6C 65 3A C</td>
<td></td>
</tr>
<tr>
<td>A0DF</td>
<td>28 28 38 39 C</td>
<td></td>
</tr>
<tr>
<td>A0E3</td>
<td>2F 33 30 2F C</td>
<td></td>
</tr>
<tr>
<td>A0E4</td>
<td>38 33 C</td>
<td></td>
</tr>
<tr>
<td>A0ED</td>
<td>80 8A C</td>
<td>DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A0EF</td>
<td>44 61 74 65 C</td>
<td>'Do you want to load the user file? (Y/N)'</td>
</tr>
<tr>
<td>A0F3</td>
<td>3A 28 28 C</td>
<td></td>
</tr>
<tr>
<td>A0F6</td>
<td>80 C</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A0F7</td>
<td>80 8A C</td>
<td>MSGUF: DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A0F9</td>
<td>44 6F 28 79 C</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A0FD</td>
<td>6F 75 28 77 C</td>
<td></td>
</tr>
<tr>
<td>A101</td>
<td>61 6E 74 20 C</td>
<td></td>
</tr>
<tr>
<td>A105</td>
<td>74 6F 20 6C C</td>
<td></td>
</tr>
<tr>
<td>A109</td>
<td>6F 61 64 20 C</td>
<td></td>
</tr>
<tr>
<td>A10D</td>
<td>74 6E 65 20 C</td>
<td></td>
</tr>
<tr>
<td>A111</td>
<td>75 73 65 72 C</td>
<td></td>
</tr>
<tr>
<td>A115</td>
<td>28 66 69 6C C</td>
<td></td>
</tr>
<tr>
<td>A119</td>
<td>65 3F 28 20 C</td>
<td></td>
</tr>
<tr>
<td>A11D</td>
<td>28 59 2F 4E C</td>
<td></td>
</tr>
<tr>
<td>A121</td>
<td>29 28 C</td>
<td></td>
</tr>
<tr>
<td>A123</td>
<td>80 C</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A124</td>
<td>80 8A C</td>
<td>MSGUF: DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A126</td>
<td>50 6C 61 63 C</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A12A</td>
<td>65 28 74 68 C</td>
<td>'Place the user file tape in the tape player.'</td>
</tr>
<tr>
<td>A12E</td>
<td>65 28 75 73 C</td>
<td></td>
</tr>
<tr>
<td>A132</td>
<td>65 72 28 66 C</td>
<td></td>
</tr>
<tr>
<td>A136</td>
<td>69 6C 65 28 C</td>
<td></td>
</tr>
<tr>
<td>A13A</td>
<td>74 61 70 65 C</td>
<td></td>
</tr>
</tbody>
</table>
A13E 28 69 6E 28 C
A142 74 68 65 28 C
A146 74 61 70 65 C
A14A 28 70 6C 61 C
A14E 79 65 72 2E C
A152 0D C
A153 58 72 65 73 C
A157 73 28 61 6E C
A15B 79 28 68 65 C
A15F 79 28 77 68 C
A163 65 6E 28 72 C
A167 65 61 64 79 C
A16B 2E C
A16C 0D 8A C
A16E 28 28 45 53 C
A172 43 28 28 2D C
A176 28 28 41 62 C
A17A 6F 72 74 C
A17D 88 C
A17E 0D 8A C
A180 58 72 65 73 C
A184 73 28 74 68 C
A188 65 28 58 4C C
A18C 41 59 28 62 C
A190 75 74 74 6F C
A194 6E 28 6F 6E C
A198 28 74 68 65 C
A19C 28 74 61 78 C
A1A0 65 28 70 6C C
A1A4 61 79 65 72 C
A1A8 2E C
A1A9 88 C
A1AA 0D 8A C
A1AC 4E 6F 28 74 C
A1B0 75 6E 65 28 C
A1B4 67 72 6F 75 C
A1B8 70 28 68 61 C
A1BC 73 26 62 65 C
A1C0 65 6E 28 6C C
A1C4 6F 61 64 65 C
A1C8 64 2E C
A1CA 0D 8A C
A1CC 44 6F 28 79 C
A1D0 6F 75 28 77 C
A1D4 61 6E 74 28 C
A1D8 74 6F 28 6C C
A1DC 6F 61 64 28 C
A1E0 61 28 74 75 C

Press any key when ready.

ESC - Abort

Press the PLAY button on the tape player.

No tune group has been loaded.

Do you want to load a tune group? (Y/N)
ROM DATA SECTION

```
A1E4  6E 65 20 67 C
A1E8  72 6F 75 70 C
A1EC  3F 20 20 20 C
A1F0  59 2F 4E 29 C
A1F4  20 C
A1F5  00 C
A1F6  0D 0A C
A1F8  54 75 6E 65 C
A1FC  26 67 72 6F C
A200  75 70 28 28 C
A204  28 20 20 20 C
A208  28 69 73 20 C
A20C  6E 6F 77 28 C
A210  6C 6F 61 64 C
A214  65 64 2E C
A217  00 0A C
A219  44 6F 20 79 C
A21D  6F 75 28 77 C
A221  61 6E 74 20 C
A225  74 6F 20 75 C
A229  73 65 20 74 C
A22D  68 69 73 20 C
A231  74 75 6E 65 C
A235  28 67 72 6F C
A239  75 70 3F 28 C
A23D  28 20 28 20 20 C
A23F  4E 20 20 20 C
A241  0D C
A245  0D 0A C
A247  50 6C 61 63 C
A24B  65 20 74 68 C
A24F  65 20 20 20 20 C
A253  73 69 72 65 C
A257  64 20 74 75 C
A25B  6E 65 20 67 C
A25F  72 6F 75 70 C
A263  28 63 61 73 C
A267  73 65 74 74 C
A26B  65 20 74 61 C
A26F  70 65 C
A271  0D C
A272  69 6E 20 74 C
A276  69 65 20 74 C
A27A  61 78 65 28 C
A27E  7E 6C 61 79 C
A282  65 72 2E 28 C
A286  28 50 72 65 C
A28A  73 73 20 61 C
```
A28E 6E 79 28 6B C
A292 65 79 28 77 C
A296 68 65 6E 28 C
A29A 72 65 61 64 C
A29E 79 2E 28 C
A2A1 0D 8A C
A2A3 20 20 45 53 C
A2A7 43 28 28 20 C
A2AB 28 28 41 62 C
A2AF 6F 72 74 C
A2B0 08 C
A2B2 0D 8A C
A2B5 55 73 65 72 C
A2B9 6E 61 6D 65 C
A2BD 3A 20 28 C
A2C0 88 C
A2C1 58 61 73 73 C
A2C5 77 6F 72 64 C
A2C9 3A 28 28 C
A2CC 88 C
A2CD 0D 8A C
A2CF 59 6F 75 28 C
A2D3 61 72 65 28 C
A2D7 6E 6F 74 28 C
A2DB 61 75 74 68 C
A2DF 6F 72 69 7A C
A2E3 65 64 28 74 C
A2E7 6F 28 75 73 C
A2EB 65 28 74 68 C
A2EF 69 73 28 73 C
A2F3 79 73 74 65 C
A2F7 6D 2E C
A2FA 4C 4F 57 45 C
A2FE 53 54 28 C
A301 4B 49 47 48 C
A305 45 53 54 C
A30B 46 69 6E 64 C
A30C 20 74 68 85 C
A310 20 26 28 20 C
A314 20 28 28 20 C
A31B 20 70 69 74 C
A31C 63 68 20 74 C
A320 68 61 74 28 C
A324 79 6F 75 28 C
A328 63 61 6E 20 C
A32C 63 6F 60 66 C
A330 6F 72 74 61 C

A28E 6E 79 28 6B C
A292 65 79 28 77 C
A296 68 65 6E 28 C
A29A 72 65 61 64 C
A29E 79 2E 28 C
A2A1 0D 8A C
A2A3 20 20 45 53 C
A2A7 43 28 28 20 C
A2AB 28 28 41 62 C
A2AF 6F 72 74 C
A2B0 08 C
A2B2 0D 8A C
A2B5 55 73 65 72 C
A2B9 6E 61 6D 65 C
A2BD 3A 28 28 C
A2C0 88 C
A2C1 58 61 73 73 C
A2C5 77 6F 72 64 C
A2C9 3A 28 28 C
A2CC 88 C
A2CD 0D 8A C
A2CF 59 6F 75 28 C
A2D3 61 72 65 28 C
A2D7 6E 6F 74 28 C
A2DB 61 75 74 68 C
A2DF 6F 72 69 7A C
A2E3 65 64 28 74 C
A2E7 6F 28 75 73 C
A2EB 65 28 74 68 C
A2EF 69 73 28 73 C
A2F3 79 73 74 65 C
A2F7 6D 2E C
A2FA 4C 4F 57 45 C
A2FE 53 54 28 C
A301 4B 49 47 48 C
A305 45 53 54 C
A30B 46 69 6E 64 C
A30C 20 74 68 85 C
A310 20 26 28 20 C
A314 20 28 28 20 C
A31B 20 70 69 74 C
A31C 63 68 20 74 C
A320 68 61 74 28 C
A324 79 6F 75 28 C
A328 63 61 6E 20 C
A32C 63 6F 60 66 C
A330 6F 72 74 61 C

DEFB 0DH,0AH
DEFM 'ESC - Abort'
DEFM 'Username: ' 'Password: '
DEFM 'You are not authorized to use this system.'
DEFM 'LOWEST'
DEFM 'HIGHEST'
DEFM 'Find the pitch that you can comfortably sing.'
ROM DATA SECTION

A334  62 6C 79 20  C
A338  73 69 6E 67  C
A33C  2E  C
A33D  8D 8A  CDEFB 0DH,0AH
A33F  58 72 65 73  CDEFM 'Press any key when ready.'
A343  73 26 61 6E  C
A347  79 20 66 65  C
A34B  79 20 77 68  C
A34F  66 6E 26 72  C
A353  65 61 64 79  C
A357  2E  C
A358  80  CDEFB 00H
A359  7D 69 6E 67  CMSGSTP: DEFM 'Sing that pitch.'
A35D  26 68 61 61  C
A361  74 26 76 69  C
A365  74 63 68 2E  C
A369  80  CDEFB 00H
A36A  7D 74 6F 70  CMSGSSI: DEFM 'Stop singing.'
A36E  26 69 6E 67  C
A372  67 69 6E 67  C
A376  2E  C
A377  80  CDEFB 00H
A37B  49 73 26 74  CMSGITR: DEFM 'Is this the right pitch? (Y/N)'
A37C  68 69 73 26  C
A380  74 69 6E 67  C
A384  72 69 67 66  C
A388  74 26 76 69  C
A38C  74 63 68 2E  C
A390  26 26 26 59  C
A394  2F 4E 29  C
A397  80  CDEFB 00H
A398  54 68 61 74  CMSGNWR: DEFM 'That's not a very wide range,'
A39C  27 73 26 6E  C
A3A0  6F 74 26 61  C
A3A4  26 76 65 72  C
A3A8  79 26 77 69  C
A3AC  64 65 26 72  C
A3B0  61 6E 26 65  C
A3B4  2C  C
A3B5  80 8A  CDEFB 0DH,0AH
A3B7  62 75 74 20  CDEFM 'but I will try to accommodate you.'
<table>
<thead>
<tr>
<th>Address</th>
<th>Hex Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A307</td>
<td>75 2E</td>
<td></td>
</tr>
<tr>
<td>A309</td>
<td>00</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A30A</td>
<td>5F 6F 75 20</td>
<td>MSGYC F: DEF M 'You can’t fool me.'</td>
</tr>
<tr>
<td>A30E</td>
<td>63 61 6E 27</td>
<td></td>
</tr>
<tr>
<td>A3E2</td>
<td>74 56 66 6F</td>
<td></td>
</tr>
<tr>
<td>A3E6</td>
<td>6F 6C 28 60</td>
<td></td>
</tr>
<tr>
<td>A3EA</td>
<td>65 2E</td>
<td></td>
</tr>
<tr>
<td>A3EC</td>
<td>0D 0A</td>
<td>DEFB 0DH,0AH</td>
</tr>
<tr>
<td>A3EE</td>
<td>59 6F 75 72</td>
<td>DEF M 'Your high pitch is lower than your low pitch.'</td>
</tr>
<tr>
<td>A3F2</td>
<td>20 68 6F 67</td>
<td></td>
</tr>
<tr>
<td>A3F6</td>
<td>68 20 70 69</td>
<td></td>
</tr>
<tr>
<td>A3FA</td>
<td>74 63 6F 20</td>
<td></td>
</tr>
<tr>
<td>A3FE</td>
<td>69 73 28 6C</td>
<td></td>
</tr>
<tr>
<td>A402</td>
<td>6F 77 65 72</td>
<td></td>
</tr>
<tr>
<td>A406</td>
<td>20 74 68 61</td>
<td></td>
</tr>
<tr>
<td>A40A</td>
<td>6E 20 79 6F</td>
<td></td>
</tr>
<tr>
<td>A40E</td>
<td>75 72 28 6C</td>
<td></td>
</tr>
<tr>
<td>A412</td>
<td>6F 77 68 70</td>
<td></td>
</tr>
<tr>
<td>A416</td>
<td>69 74 63 68</td>
<td></td>
</tr>
<tr>
<td>A41A</td>
<td>2E</td>
<td></td>
</tr>
<tr>
<td>A41B</td>
<td>00</td>
<td>DEFB 00H</td>
</tr>
<tr>
<td>A41C</td>
<td>54 68 6D 20</td>
<td>MSGMOD: DEF M 'The following options are available:'</td>
</tr>
<tr>
<td>A420</td>
<td>66 6F 6C 6C</td>
<td></td>
</tr>
<tr>
<td>A424</td>
<td>6F 77 6F 6E</td>
<td></td>
</tr>
<tr>
<td>A428</td>
<td>67 28 6F 70</td>
<td></td>
</tr>
<tr>
<td>A42C</td>
<td>74 69 6F 6E</td>
<td></td>
</tr>
<tr>
<td>A430</td>
<td>73 28 61 72</td>
<td></td>
</tr>
<tr>
<td>A434</td>
<td>65 28 61 76</td>
<td></td>
</tr>
<tr>
<td>A438</td>
<td>61 69 6C 61</td>
<td></td>
</tr>
<tr>
<td>A43C</td>
<td>62 6C 65 3A</td>
<td></td>
</tr>
<tr>
<td>A440</td>
<td>0D 0A 0A 89</td>
<td>DEF B 0DH,0AH,0AH,HT,04H</td>
</tr>
<tr>
<td>A444</td>
<td>04</td>
<td></td>
</tr>
<tr>
<td>A445</td>
<td>27 54 27 28</td>
<td>DEF M 'T' - Tunes'</td>
</tr>
<tr>
<td>A449</td>
<td>20 20 28 28</td>
<td></td>
</tr>
<tr>
<td>A44D</td>
<td>54 75 6E 65</td>
<td></td>
</tr>
<tr>
<td>A451</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>A452</td>
<td>0D 0A 09 04</td>
<td>DEF B 0DH,0AH,HT,04H</td>
</tr>
<tr>
<td>A456</td>
<td>27 46 27 28</td>
<td>DEF M 'F' - Free mode'</td>
</tr>
<tr>
<td>A45A</td>
<td>20 20 28 28</td>
<td></td>
</tr>
<tr>
<td>A45E</td>
<td>46 72 65 65</td>
<td></td>
</tr>
<tr>
<td>A462</td>
<td>20 60 6F 64</td>
<td></td>
</tr>
<tr>
<td>A466</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>A467</td>
<td>0D 0A 09 04</td>
<td>DEF B 0DH,0AH,HT,04H</td>
</tr>
<tr>
<td>A46B</td>
<td>27 52 27 28</td>
<td>DEF M 'R' - Reset your range'</td>
</tr>
<tr>
<td>A46F</td>
<td>20 20 28 28</td>
<td></td>
</tr>
<tr>
<td>A473</td>
<td>52 65 73 65</td>
<td></td>
</tr>
<tr>
<td>A477</td>
<td>74 26 79 6F</td>
<td></td>
</tr>
<tr>
<td>A47B</td>
<td>75 72 28 72</td>
<td></td>
</tr>
</tbody>
</table>
ROM DATA SECTION

A47F 61 6E 67 65 C
A483 8D 0A 89 84 C
A487 77 58 27 20 C
A48B 20 2D 20 20 C
A48F 45 78 69 74 C
A493 8D 0A 8A C
A496 53 65 6C 65 C
A499 63 74 20 74 C
A49E 68 65 28 6C C
A4A2 65 74 74 65 C
A4A6 72 20 6F 66 C
A4AA 20 74 68 65 C
A4AE 6F 70 74 C
A4B2 69 6F 6E 20 C
A4B6 79 6F 75 20 C
A4BA 77 6F 75 6C C
A4BE 64 28 6C 69 C
A4C2 68 65 3A 28 C
A4C6 20 C
A4C7 00 C
A4C8 54 46 52 58 C
A4CC FF C
A4CD 49 20 64 6F C
A4D1 20 6E 6F 74 C
A4D5 28 68 61 76 C
A4D9 65 28 61 6E C
A4DD 79 28 6D 6F C
A4E1 72 65 20 74 C
A4E5 75 6E 65 73 C
A4E9 28 77 68 69 C
A4ED 65 68 28 61 C
A4F1 72 65 28 73 C
A4F5 75 69 74 65 C
A4F9 64 C
A4FA 0D C
A4FB 74 6F 20 79 C
A4FF 6F 75 72 20 C
A503 65 75 72 72 C
A507 65 6E 74 20 C
A50B 72 61 4E 67 C
A50F 65 28 61 6E C
A513 64 28 6C 65 C
A517 76 65 6C 20 C
A51B 6F 66 28 65 C
A51F 78 70 65 72 C
A523 74 69 73 65 C
A527 2E C
A528 0D 0A 09 04 C

DEFB 0DH,0AH,HT,04H
DEFM 'Select the letter of the option you would like: '
DEFB 0DH,0AH,0AH
DEFM 'Select the letter of the option you would like: '
DEFB 00H
DEFB 'T','F','R','X'
DEFB 0FFH
MSGNMT: DEFM 'I do not have any more tunes which are suited'
DEFB 0DH
DEFM 'to your current range and level of expertise.'
ROM DATA SECTION

A52C 57 6F 75 6C C DEFM 'Would you like to:
A538 64 28 79 6F C
A534 75 28 6C 69 C
A538 68 65 28 74 C
A53C 6F 3A C
A53E 00 8A 08 88 C DEFB 0DH,0AH,HT,08H
A542 27 41 27 2C 0D 0A 08 89 DEFMB 'A' - Do the same tunes again'
A546 26 28 20 28 C
A54A 44 6F 28 74 C
A54E 66 65 28 73 C
A552 61 6D 65 28 C
A556 74 75 6E 65 C
A55A 73 28 61 67 C
A55E 61 69 6E C
A561 00 8A 09 88 C DEFB 0DH,0AH,HT,08H
A565 27 45 27 2C 0D 0A 08 89 DEFMB 'E' - Expand your range'
A569 28 28 20 28 C
A56D 45 78 78 61 C
A571 6E 64 28 79 C
A575 6F 75 72 28 C
A579 72 61 6E 67 C
A57D 65 C
A57E 00 8A 09 88 C DEFB 0DH,0AH,HT,08H
A582 27 4D 27 2C 0D 0A 08 89 DEFMB 'M' - Display main menu'
A586 28 28 28 2C
A58A 44 69 73 78 C
A58E 6C 61 79 28 C
A592 60 61 69 6E C
A596 28 6D 65 6E C
A59A 75 C
A59E 00 8A 8A 89 C DEFB 0DH,0AH,HT,04H
A59F 04 C
A5A4 53 65 6C 65 C DEFM 'Select the letter of the option you would like: '
A5A8 63 74 28 74 C
A5AC 6D 74 28 65 C
A5B0 72 28 6F 66 C
A5B4 28 74 68 65 C
A5B8 28 6F 78 74 C
A5BC 69 6F 6E 28 C
A5C0 79 6F 75 28 C
A5C4 77 6F 75 6C C
A5C8 64 28 6C 69 C
A5CC 68 65 3A 28 C
A5D0 28 C
A5D1 08 C DEFB 08H
A5D2 41 45 4D FF C DEFB 'A', 'E', 'M', 0FFH
A5D6 59 6F 75 28 C MSGETG: DEFM 'You have sung every tune that I have in this tune group.'
<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A5DA</td>
<td>68 61 76 65 C</td>
<td>ROM DATA SECTION</td>
</tr>
<tr>
<td>A5DE</td>
<td>20 73 75 6E C</td>
<td></td>
</tr>
<tr>
<td>A5E2</td>
<td>67 20 65 76 C</td>
<td></td>
</tr>
<tr>
<td>A5E6</td>
<td>65 72 79 20 C</td>
<td></td>
</tr>
<tr>
<td>A5EA</td>
<td>74 75 6E 65 C</td>
<td></td>
</tr>
<tr>
<td>A5EE</td>
<td>20 74 68 61 C</td>
<td></td>
</tr>
<tr>
<td>A5F2</td>
<td>74 28 49 28 C</td>
<td></td>
</tr>
<tr>
<td>A5F6</td>
<td>68 61 76 65 C</td>
<td></td>
</tr>
<tr>
<td>A5FA</td>
<td>20 69 6E 28 C</td>
<td></td>
</tr>
<tr>
<td>A5FE</td>
<td>74 68 69 73 C</td>
<td></td>
</tr>
<tr>
<td>A602</td>
<td>20 74 75 6E C</td>
<td></td>
</tr>
<tr>
<td>A606</td>
<td>65 20 67 72 C</td>
<td></td>
</tr>
<tr>
<td>A60A</td>
<td>6F 75 70 2E C</td>
<td></td>
</tr>
<tr>
<td>A60E</td>
<td>0D 0A 89 84 C</td>
<td>DEFB 0DH,0AH,HT,04H</td>
</tr>
<tr>
<td>A612</td>
<td>57 6F 75 6C C</td>
<td>DEFM 'Would you like to:'</td>
</tr>
<tr>
<td>A616</td>
<td>64 20 79 6F C</td>
<td></td>
</tr>
<tr>
<td>A61A</td>
<td>75 20 6C 69 C</td>
<td></td>
</tr>
<tr>
<td>A61E</td>
<td>68 65 20 74 C</td>
<td></td>
</tr>
<tr>
<td>A622</td>
<td>6F 3A C</td>
<td></td>
</tr>
<tr>
<td>A624</td>
<td>0D 0A 89 88 C</td>
<td>DEFB 0DH,0AH,HT,08H</td>
</tr>
<tr>
<td>A628</td>
<td>27 41 27 28 C</td>
<td>DEFM '&quot;A&quot; - Do the same tunes over again or'</td>
</tr>
<tr>
<td>A62C</td>
<td>20 2D 20 28 C</td>
<td></td>
</tr>
<tr>
<td>A630</td>
<td>44 6F 20 74 C</td>
<td></td>
</tr>
<tr>
<td>A634</td>
<td>68 65 20 73 C</td>
<td></td>
</tr>
<tr>
<td>A638</td>
<td>61 6D 65 26 C</td>
<td></td>
</tr>
<tr>
<td>A63C</td>
<td>74 75 6E 65 C</td>
<td></td>
</tr>
<tr>
<td>A640</td>
<td>73 20 6F 76 C</td>
<td></td>
</tr>
<tr>
<td>A644</td>
<td>65 72 20 61 C</td>
<td></td>
</tr>
<tr>
<td>A648</td>
<td>67 61 69 6E C</td>
<td></td>
</tr>
<tr>
<td>A64C</td>
<td>20 6F 72 C</td>
<td></td>
</tr>
<tr>
<td>A64F</td>
<td>0D 0A 89 88 C</td>
<td>DEFB 0DH,0AH,HT,08H</td>
</tr>
<tr>
<td>A653</td>
<td>27 4C 27 28 C</td>
<td>DEFM '&quot;L&quot; - Load new tune group'</td>
</tr>
<tr>
<td>A657</td>
<td>26 2D 26 26 C</td>
<td></td>
</tr>
<tr>
<td>A65B</td>
<td>4C 6F 61 64 C</td>
<td></td>
</tr>
<tr>
<td>A65F</td>
<td>26 6E 65 77 C</td>
<td></td>
</tr>
<tr>
<td>A663</td>
<td>20 74 75 6E C</td>
<td></td>
</tr>
<tr>
<td>A667</td>
<td>65 20 67 72 C</td>
<td></td>
</tr>
<tr>
<td>A66B</td>
<td>6F 75 70 C</td>
<td></td>
</tr>
<tr>
<td>A66E</td>
<td>0D 0A 89 88 C</td>
<td>DEFB 0DH,0AH,HT,08H</td>
</tr>
<tr>
<td>A672</td>
<td>27 4D 27 28 C</td>
<td>DEFM '&quot;M&quot; - Display main menu'</td>
</tr>
<tr>
<td>A676</td>
<td>26 2D 26 26 C</td>
<td></td>
</tr>
<tr>
<td>A67A</td>
<td>44 69 73 76 C</td>
<td></td>
</tr>
<tr>
<td>A67E</td>
<td>6C 61 79 28 C</td>
<td></td>
</tr>
<tr>
<td>A682</td>
<td>6D 61 69 6E C</td>
<td></td>
</tr>
<tr>
<td>A686</td>
<td>26 6D 65 6E C</td>
<td></td>
</tr>
<tr>
<td>A68A</td>
<td>75 C</td>
<td></td>
</tr>
<tr>
<td>A68B</td>
<td>0D 0A 0A 89 C</td>
<td>DEFB 0DH,0AH,0AH,HT,04H</td>
</tr>
<tr>
<td>A68F</td>
<td>04 C</td>
<td></td>
</tr>
<tr>
<td>A698</td>
<td>53 65 6C 65</td>
<td>C</td>
</tr>
<tr>
<td>A694</td>
<td>63 74 28 74</td>
<td>C</td>
</tr>
<tr>
<td>A69B</td>
<td>68 65 28 6C</td>
<td>C</td>
</tr>
<tr>
<td>A69C</td>
<td>65 74 74 65</td>
<td>C</td>
</tr>
<tr>
<td>A690</td>
<td>72 28 6F 66</td>
<td>C</td>
</tr>
<tr>
<td>A6A4</td>
<td>20 74 68 65</td>
<td>C</td>
</tr>
<tr>
<td>A6A8</td>
<td>28 6F 78 74</td>
<td>C</td>
</tr>
<tr>
<td>A6AC</td>
<td>69 6F 6E 28</td>
<td>C</td>
</tr>
<tr>
<td>A6B8</td>
<td>79 6F 75 28</td>
<td>C</td>
</tr>
<tr>
<td>A6B4</td>
<td>77 6F 75 6C</td>
<td>C</td>
</tr>
<tr>
<td>A6BB</td>
<td>64 20 6C 69</td>
<td>C</td>
</tr>
<tr>
<td>A6BC</td>
<td>6B 65 3A 28</td>
<td>C</td>
</tr>
<tr>
<td>A6C8</td>
<td>20</td>
<td>C</td>
</tr>
<tr>
<td>A6C1</td>
<td>00</td>
<td>C</td>
</tr>
<tr>
<td>A6C2</td>
<td>41 4C 4D FF</td>
<td>C</td>
</tr>
<tr>
<td>A6C6</td>
<td>57 68 61 74</td>
<td>C</td>
</tr>
<tr>
<td>A6CA</td>
<td>28 68 65 79</td>
<td>C</td>
</tr>
<tr>
<td>A6CE</td>
<td>20 69 73 28</td>
<td>C</td>
</tr>
<tr>
<td>A6D2</td>
<td>74 68 69 73</td>
<td>C</td>
</tr>
<tr>
<td>A6D6</td>
<td>20 74 75 6E</td>
<td>C</td>
</tr>
<tr>
<td>A6DA</td>
<td>65 20 69 6E</td>
<td>C</td>
</tr>
<tr>
<td>A6DE</td>
<td>3F</td>
<td>C</td>
</tr>
<tr>
<td>A6DF</td>
<td>00</td>
<td>C</td>
</tr>
<tr>
<td>A6E8</td>
<td>09 64</td>
<td>C</td>
</tr>
<tr>
<td>A6E2</td>
<td>4D 61 6A 6F</td>
<td>C</td>
</tr>
<tr>
<td>A6E6</td>
<td>72 20 26 2D</td>
<td>C</td>
</tr>
<tr>
<td>A6EA</td>
<td>20 26 55 70</td>
<td>C</td>
</tr>
<tr>
<td>A6EE</td>
<td>70 65 72 28</td>
<td>C</td>
</tr>
<tr>
<td>A6F2</td>
<td>63 61 73 65</td>
<td>C</td>
</tr>
<tr>
<td>A6F4</td>
<td>0D 69 84</td>
<td>C</td>
</tr>
<tr>
<td>A6F9</td>
<td>4D 69 6E 6F</td>
<td>C</td>
</tr>
<tr>
<td>A6FD</td>
<td>72 28 28 2D</td>
<td>C</td>
</tr>
<tr>
<td>A701</td>
<td>20 28 4C 6F</td>
<td>C</td>
</tr>
<tr>
<td>A705</td>
<td>77 65 72 28</td>
<td>C</td>
</tr>
<tr>
<td>A709</td>
<td>63 61 73 65</td>
<td>C</td>
</tr>
<tr>
<td>A70D</td>
<td>00 89 84</td>
<td>C</td>
</tr>
<tr>
<td>A710</td>
<td>53 68 61 72</td>
<td>C</td>
</tr>
<tr>
<td>A714</td>
<td>78 28 28 2D</td>
<td>C</td>
</tr>
<tr>
<td>A718</td>
<td>20 26 27 23</td>
<td>C</td>
</tr>
<tr>
<td>A71C</td>
<td>27 28 6F 72</td>
<td>C</td>
</tr>
<tr>
<td>A728</td>
<td>20 27 73 27</td>
<td>C</td>
</tr>
<tr>
<td>A72C</td>
<td>0D 89 84</td>
<td>C</td>
</tr>
<tr>
<td>A727</td>
<td>46 6C 61 74</td>
<td>C</td>
</tr>
<tr>
<td>A72B</td>
<td>28 20 28 2D</td>
<td>C</td>
</tr>
<tr>
<td>A72F</td>
<td>28 20 44 6F</td>
<td>C</td>
</tr>
<tr>
<td>A733</td>
<td>75 62 6C 65</td>
<td>C</td>
</tr>
<tr>
<td>A737</td>
<td>20 6C 65 74</td>
<td>C</td>
</tr>
<tr>
<td>A73B</td>
<td>74 65 72 28</td>
<td>C</td>
</tr>
</tbody>
</table>
A73F 6F 72 20 27 C
A743 66 27 C
A745 8D 8A 89 84 C
A749 45 78 61 6D C
A74D 78 6C 65 3A C
A751 20 20 20 23 C
A755 20 20 20 43 C
A759 28 73 68 61 C
A75D 72 78 20 6D C
A761 69 6E 6F 72 C
A765 0D 89 8E C
A768 41 41 20 2D C
A76C 28 41 28 66 C
A770 6C 61 74 28 C
A774 60 6A 6F 6E C
A778 72 C
A779 00 C
A77A 89 84 C
A77C 27 3F 27 20 C
A780 20 20 20 2C C
A784 53 68 6F 77 C
A788 20 74 68 65 C
A78C 20 61 6E 73 C
A790 75 65 72 C
A793 80 89 84 C
A796 45 53 43 20 C
A799 20 20 20 2C C
A79C 44 69 73 70 C
A79F 26 61 79 20 C
A7A3 6F 70 74 69 C
A7A7 6F 6E 20 6D C
A7AE 65 6E 75 C
A7B1 00 C
A7B2 57 72 6F 6E C
A7B4 67 2E 20 28 C
A7B8 5A 72 79 28 C
A7BC 61 67 61 69 C
A7C2 6E 2E C
A7C4 00 C
A7C5 4E 6F 2E 20 C
A7C9 20 5A 68 61 C
A7CD 74 27 73 28 C
A7D1 6E 6F 74 28 C
A7D5 72 69 67 68 C
A7D9 74 2E C
A7DE 00 C
A7DC 5A 68 61 74 C
A7EB 27 73 20 72 C
A7EA 69 67 6B 74 C
A7E9 2E C
A7E8 00 C DEFB 00H
A7EA 58 72 65 73 C MSGPAK: DEFM 'Press any key to continue.'
A7EE 73 20 61 6E C
A7F2 79 20 69 6E C
A7F6 79 20 74 6F C
A7FA 20 63 6F 6E C
A7FE 74 69 6E 75 C
A882 65 2E C
A884 00 C DEFB 00H
A885 58 72 6F 63 C MSGPPM: DEFM 'Processing. Please wait.'
A889 65 73 73 69 C
A88D 6E 67 2E 28 C
A811 20 50 6C 6F 69 73 C
A815 61 73 65 20 77 69 74 C
A81D 2E C
A81E 00 C DEFB 00H
A81F 49 20 64 6F 65 73 74 79 28 31 28 33 20 35 20 31 29 C KEYHDR: DEFM 'Key:
A855 20 20 20 20 C
A859 00 C DEFB 00H
A85A 53 69 6E 67 C MSGSTT: DEFM 'Sing the tonic triad (1 3 5 3 1).'
A85E 20 74 6E 65 C
A862 20 74 6F 6E C
A866 69 63 20 74 C
A86A 72 69 61 64 C
A86E 20 28 31 28 C
A872 33 28 35 28 C
A876 33 28 31 29 C
A87A 2E C
A87B 00 6A 09 83 C DEFB 0DH,0AH,HT,03H
A87F 49 20 77 69 C DEFM 'I will display the number of each pitch'
A883 6C 6C 28 64 C
A887 69 73 78 6C C
A88B 61 79 28 74 C
A88F 68 65 28 6E C
A893 75 6D 62 65 C
A897 72 28 6F 66 C
A89B 28 65 61 63 C
A89F 68 28 78 69 C
A8A3 74 63 68 C
A8A6 9D 69 03 C
A8A9 77 68 65 6E C
A8AD 67 28 6F 75 C
A8B1 28 73 69 6E C
A8B5 67 28 69 74 C
A8B9 28 63 6F 72 C
A8BD 72 65 63 74 C
A8C1 6C 79 2E C
A8C4 8D 0A 09 07 C
A8C8 27 58 27 28 C
A8CC 20 2D 28 28 C
A8D0 50 6C 61 79 C
A8D4 20 74 68 65 C
A8D8 28 74 72 69 C
A8DC 61 64 C
A8E1 8D 09 07 C
A8E5 28 2D 28 28 C
A8ED 6C 61 79 78 C
A8F1 6F 70 74 69 C
A8F5 6F 6E 28 60 C
A8FF 65 6E 75 C
A903 8D 09 05 C
A907 53 58 41 43 C
A90B 28 28 28 28 C
A90F 74 72 69 61 C
A913 64 28 6F 76 C
A917 65 78 28 61 C
A91B 67 61 69 6E C
A91F 02 C
A920 31 28 28 28 C
A924 8D 09 05 C
A928 33 28 28 28 C
A92C 8D 09 05 C
A930 35 28 28 28 C
A934 8D 09 05 C
A938 8D 09 05 C
A93C 8D 09 05 C
A940 8D 09 05 C
A944 8D 09 05 C
A948 8D 09 05 C
A94C 8D 09 05 C
A950 8D 09 05 C
A954 8D 09 05 C
A958 8D 09 05 C
A95C 8D 09 05 C
A960 8D 09 05 C
A964 8D 09 05 C
A968 8D 09 05 C
A96C 8D 09 05 C
A970 8D 09 05 C
A974 8D 09 05 C
A978 8D 09 05 C
A97C 8D 09 05 C
A980 8D 09 05 C
A984 8D 09 05 C
A988 8D 09 05 C
A98C 8D 09 05 C
A990 8D 09 05 C
A994 8D 09 05 C
A998 8D 09 05 C
A99C 8D 09 05 C
A9A0 8D 09 05 C
A9A4 8D 09 05 C
A9A8 8D 09 05 C
A9AC 8D 09 05 C
A9B0 8D 09 05 C
A9B4 8D 09 05 C
A9B8 8D 09 05 C
A9BC 8D 09 05 C
A9C0 8D 09 05 C
A9C4 8D 09 05 C
A9C8 8D 09 05 C
A9CC 8D 09 05 C
A9D0 8D 09 05 C
A9D4 8D 09 05 C
A9D8 8D 09 05 C
A9DC 8D 09 05 C
A9E0 8D 09 05 C
A9E4 8D 09 05 C
A9E8 8D 09 05 C
A9EC 8D 09 05 C
A9F0 8D 09 05 C
A9F4 8D 09 05 C
A9F8 8D 09 05 C
A9FC 8D 09 05 C
A9FF 8D 09 05 C
A9F0 8D 09 05 C
A9F4 8D 09 05 C
A9F8 8D 09 05 C
A9FC 8D 09 05 C
A9FF 8D 09 05 C
A92F 33 20 20 20 C  DEFB '3'
A933 00 C  DEFB 00H
A934 31 C  DEFB '1'
A935 00 C  DEFB 00H
A936 4C 4F 4B C  MSGLOT: DEFB 'Look over the tune for a minute and'
A93A 20 6F 76 65 C
A93E 72 20 74 68 C
A942 65 20 74 75 C
A946 6E 65 20 66 C
A94A 6F 72 20 61 C
A94E 28 6D 69 6E C
A952 75 74 65 20 C
A956 61 6E 64 C
A959 0D C  DEFB 00H
A95A 70 72 65 73 C  DEFB 'press any key when you are ready to sing it.'
A95E 73 20 61 6E C
A962 79 20 68 65 C
A966 79 20 77 68 C
A96A 65 6E 20 79 C
A96E 6F 75 20 61 C
A972 20 65 20 72 C
A976 65 61 64 79 C
A97A 28 74 6F 20 C
A97E 73 69 6E 67 C
A982 0D 6A 89 04 C  DEFB 0DH,0AH,HT,04H
A984 0D 0A 89 04 C  DEFM '"T" - Play tonic note'
A98A 27 54 27 20 C
A98E 20 20 20 20 C
A992 50 6C 61 79 C
A996 28 74 6F 6E C
A99A 69 63 20 6E C
A99E 6F 74 65 C
A9A1 0D 09 04 C  DEFB 0DH,HT,04H
A9A4 45 53 43 28 C  DEFM 'ESC - Display option menu'
A9A8 20 20 20 20 C
A9AC 44 69 73 70 C
A9B0 6C 61 79 20 C
A9B4 6F 70 74 69 C
A9B8 6E 6E 20 6D C
A9BC 65 6E 75 C
A9BF 00 C  DEFB 00H
A9C0 57 6F 75 6C C  MSGSM: DEFM 'Would you like to:
A9C4 64 20 79 6F C
A9C8 75 20 6C 6F C
A9CC 6B 65 20 74 C
A9D0 6F 3A C
A9D2 0D 8A 8A 09 C  DEFB 0DH,0AH,0AH,HT,04H
A9D6 04 C
<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A9D7</td>
<td>27 53 27 20</td>
<td>DEFM ‘‘S’’ - Skip this tune</td>
</tr>
<tr>
<td>A9D8</td>
<td>28 20 28 20</td>
<td></td>
</tr>
<tr>
<td>A9D9</td>
<td>53 68 69 70</td>
<td></td>
</tr>
<tr>
<td>A9D2</td>
<td>28 73 68 69</td>
<td></td>
</tr>
<tr>
<td>A9D7</td>
<td>73 73 74 75</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>66 65</td>
<td></td>
</tr>
<tr>
<td>A9E4</td>
<td>00 0A 09 04</td>
<td>DEFB 0DH,0AH,HT,04H</td>
</tr>
<tr>
<td>A9F1</td>
<td>27 52 27 20</td>
<td>DEFM ‘‘R’’ - Return to where you were</td>
</tr>
<tr>
<td>A9F5</td>
<td>28 20 20 20</td>
<td></td>
</tr>
<tr>
<td>A9F9</td>
<td>52 65 74 75</td>
<td></td>
</tr>
<tr>
<td>A9FD</td>
<td>72 6E 2B 74</td>
<td></td>
</tr>
<tr>
<td>AA01</td>
<td>66 26 77 68</td>
<td></td>
</tr>
<tr>
<td>AA05</td>
<td>65 72 65 20</td>
<td></td>
</tr>
<tr>
<td>AA09</td>
<td>79 6E 75 20</td>
<td></td>
</tr>
<tr>
<td>AA11</td>
<td>77 65 72 45</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>0D 0A 09 04</td>
<td>DEFB 0DH,0AH,HT,04H</td>
</tr>
<tr>
<td>A9E4</td>
<td>27 4D 27 20</td>
<td>DEFM ‘‘M’’ - Display main menu</td>
</tr>
<tr>
<td>A9F1</td>
<td>28 20 20 20</td>
<td></td>
</tr>
<tr>
<td>A9F9</td>
<td>44 69 73 70</td>
<td></td>
</tr>
<tr>
<td>A9FD</td>
<td>6C 61 79 20</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>60 61 69 6E</td>
<td></td>
</tr>
<tr>
<td>A9F9</td>
<td>20 6D 65 6E</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>A9E4</td>
<td>0D 0A 0A</td>
<td>DEFB 0DH,0AH,OAH</td>
</tr>
<tr>
<td>A9E0</td>
<td>53 65 6C 65</td>
<td>DEFM ‘‘Select the letter of the option you would like: ’</td>
</tr>
<tr>
<td>A9F9</td>
<td>63 74 28 74</td>
<td></td>
</tr>
<tr>
<td>A9F9</td>
<td>6C 6C 6C</td>
<td></td>
</tr>
<tr>
<td>A9F5</td>
<td>65 74 74 65</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>72 28 6F 66</td>
<td></td>
</tr>
<tr>
<td>A9F5</td>
<td>6D 74 68 65</td>
<td></td>
</tr>
<tr>
<td>A9F5</td>
<td>28 6F 78 74</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>67 6F 6E 20</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>79 6F 75 20</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>77 6F 75 6C</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>64 28 6C 69</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>6B 65 3A 20</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>2B</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>08</td>
<td>DEFB 08H</td>
</tr>
<tr>
<td>A9E0</td>
<td>53 52 4D FF</td>
<td>DEFB ‘‘S’, ‘R’, ‘M’, 0FFH</td>
</tr>
<tr>
<td>A9E0</td>
<td>57 6F 75 6C</td>
<td>MSGSRP: DEFM ‘Would you like to:’</td>
</tr>
<tr>
<td>A9E0</td>
<td>64 2B 79 6F</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>75 28 6C 69</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>6B 65 2B 74</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>6F 3A</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>80 0A 08 09</td>
<td>DEFB 0DH,0AH,0AH,HT,04H</td>
</tr>
<tr>
<td>A9E0</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>A9E0</td>
<td>27 53 27 20</td>
<td>DEFM ‘‘S’’ - Skip this tune</td>
</tr>
<tr>
<td>A9E0</td>
<td>28 20 28 20</td>
<td></td>
</tr>
</tbody>
</table>
AA86 53 68 69 78 C
AA8A 20 74 68 69 C
AA8E 73 26 74 75 C
AA92 4E 65 C
AA94 0D 0A 89 04 C
AA98 27 52 27 28 C
AA9C 20 20 28 28 C
AA9B 20 32 74 75 C
AA9D 20 74 54 26 C
AA9F 27 74 75 6E C
AAAF 77 65 72 65 C
AA8B 0D 0A 89 04 C
AA8D 27 58 27 28 C
AA8F 20 20 28 28 C
AA91 5C 61 79 79 C
AA95 20 74 68 65 C
AA99 20 74 75 6E C
AA9B 65 C
AA9D 0D 0A 89 04 C
AA9F 27 40 27 28 C
AA99 20 20 28 28 C
AA9D 44 69 73 70 C
AA9E 8C 61 79 20 C
AA9F 6D 61 69 6E C
AA9F 20 46 65 6E C
AAAC 75 C
AAAE 0D 0A 0A C
AAAF 53 65 66 65 C
AAAF 63 74 28 74 C
AAAF 48 65 20 6C C
AAFA 65 74 74 65 C
AAFD 72 20 6F 66 C
AAFE 28 74 68 65 C
AAFD 20 6F 70 74 C
AAFE 69 6F 6E 28 C
AB01 79 6F 75 28 C
AB07 77 6F 75 6C C
AB0D 64 28 6C 69 C
AB0F 68 65 3A 28 C
AB14 20 C
AB17 00 H
AB19 53 52 50 4D C
AB1E FF C
AB20 57 6F 75 6C C
AB22 64 28 79 6F C
AB28 75 28 6C 69 C

DEFB 0DH,0AH,HT,04H
DEFM "'R'" - Return to where you were'

DEFB 0DH,0AH,HT,04H
DEFM "'P'" - Play the tune'

DEFB 0DH,0AH,HT,04H
DEFM "'M'" - Display main menu'

DEFB 0DH,0AH,0AH
DEFM 'Select the letter of the option you would like: '

DEFB 00H
DEFB 'S','R','P','M',0FFH

MSGCAP: DEFM 'Would you like to:'
AB34  68 65 20 74  C  DEFB  0DH,0AH,0AH,HT,04H
AB38  6F 3A  C
AB3A  80 8A 8A 89  C  DEFM  "'C'" - Continue with another tune'
AB3E  84  C
AB3F  27 43 27 20  C  DEFM  "'A'" - Try this tune again'
AB43  28 2D 20 20  C
AB47  43 6F 6E 74  C
AB4B  69 6E 75 65  C
AB4F  28 77 69 74  C
AB53  68 20 61 6E  C
AB57  6F 74 68 65  C
AB5B  72 20 74 75  C
AB5F  6E 65  C
AB61  8D 8A 89 84  C  DEFB  0DH,0AH,HT,04H
AB65  27 41 27 20  C  DEFM  "'P'" - Play the tune'
AB69  28 2D 20 20  C
AB6D  54 72 79 20  C
AB71  74 68 69 73  C
AB75  28 74 75 6E  C
AB79  65 28 61 67  C
AB7D  61 69 6E  C
AB81  8D 8A 89 84  C  DEFB  0DH,0AH,HT,04H
AB85  75  C
AB89  8D 8A 89 84  C  DEFM  "'M'" - Display main menu'
AB8D  27 40 27 20  C
AB91  20 20 20 20  C
AB95  44 69 73 70  C
AB99  6C 61 79 20  C
AB9D  60 61 69 6E  C
ABB1  28 6D 65 6E  C
ABB5  69 6E 65 6C  C
ABBB  6D 69 6E 65  C
ABB9  53 65 6C 65  C
ABB9  63 74 20 74  C
ABCD  28 64 68 65  C
ABE1  64 28 6C 69  C
"You missed out of pitches."

'Score: %'

"Which clef would you like to use?"

"'T' - Treble"

"'B' - Bass"

'Select the letter of the clef you would like: '
ROM DATA SECTION

AC8F 75 6C 64 20 C
AC93 6C 69 6B 65 C
AC97 3A 20 20 C
AC9A 00 C
DEFB 00H
AC9B 54 42 FF C
DEFB 'T','B',0FFH
AC9E 57 68 61 74 C
MSGAKS: DEFB 'What key signature would you like:'
AC9F 20 6B 65 79 C
AC9F 79 6F 75 67 C
ACAE 72 65 20 77 C
ACBA 65 3A C
DEFB 00H
ACB1 66 61 74 75 C
ACB2 6E 75 64 64 C
ACB5 20 20 20 20 C
ACBE 65 3A C
DEFB 00H
ACBF 54 68 6F 20 C
MSGHAY: DEFB 'Who are you trying to kid?'
ACD3 61 72 65 20 C
ACD4 79 6F 75 20 C
ACD5 74 72 79 69 C
ACD6 6E 67 20 74 C
ACD7 6F 28 6B 69 C
ACD8 64 3F C
ACD9 8D 8A C
DEFB 0D,0AH
ACDB 0D 8A C
DEFB 0D,0AH
ACDC 54 68 61 74 C
DEFM 'Who are you trying to kid?'
ACDD 20 6B 65 20 C
ACDF 79 6F 75 67 C
ACFA 72 65 20 77 C
ACFB 65 3A C
DEFB 00H
ACFF 68 65 72 20 C
ACFD 68 65 79 2E C
ACFE 54 72 79 20 C
DEFM 'Try another key.'
ACF5 20 68 65 79 C
ACF6 69 73 74 2E C
ACF7 0D 8A C
DEFB 0D,0AH
ACF8 54 72 79 20 C
DEFM 'Try another key.'
ACF9 61 6E 6F 74 C
ACFA 68 65 72 20 C
ACFB 68 65 79 2E C
ACFC 00 C
DEFB 00H
ACFD 58 72 65 73 C
MSGAKS: DEFB 'Press any key to stop'
ACFE 73 20 61 6E C
AD00 79 20 68 65 C
AD01 79 20 74 4F C
AD02 20 73 74 4F C
AD03 20 20 20 20 C
AD04 70 C
AD05 00 C
DEFB 00H
AD06 49 4E 56 41 C
MSGINV: DEFB 'INVALID OPTION. Re-enter.'
AD07 4C 49 44 20 C
AD08 4F 50 54 49 C
AD09 4F 4E 2E 20 C
AD0A 20 20 52 65 C
ROM DATA SECTION

AD36  2D 65 6E 74  C
AD3A  65 72 2E  C
AD3D  08  C        DEFB  00H
               C
               ;
               ;  .DEPHASE
               ;
SUBTTL  PROGRAM SECTION
*EJECT
SIGHTSINGER'S MIRROR
PROGRAM SECTION

MACRO-88 3.44 09-Dec-81 PAGE 1-29

CSEG
ORG 0000H

; CORG EQU $ ; PHASE BASE+0E80H

CBASE EQU $ ; MACLIB B:SSM00.MAC

; COLD

; COLD START INITIALIZATION ROUTINE.

COLD: LD HL, TOPRAM-1 ; SET UP SORCERER SYSTEM AREA
AE03 22 F000 C LD (ENDRAM), HL ; AND STACK POINTER
AE06 FD 2A F000 C LD IY, (ENDRAM) ; PUT ADDRESS IN (IY)
AE0F 01 FF92 C LD BC, 0FF92H ; LENGTH OF SYSTEM AREA
AE10 FD 09 C ADD IY, BC ; POINT TO TOP OF STACK/MWA
AE0F FD F9 C LD SP, IY ; SET STACK POINTER
AE11 7D C LD A, L ; END OF AREA ADDRESS
AE12 FD E5 C PUSH IY
AE14 E1 C POP HL
AE15 36 00 C COLD01: LD (HL), 00H ; INITIALIZE SYSTEM AREA
AE17 23 C INC HL ; TO ALL ZERO
AE18 BD C CP L
AE19 28 FA C JR NZ, COLD01
AE1B FD 30 3D 40 C LD (IY+3DH), 40H ; DEFAULT TAPE/BAUD
AE1F FD 36 44 3E C LD (IY+44H), ’ ’ ; MONITOR PROMPT CHARACTER
AE23 11 BE9 C LD DE, KBINPT ; SET UP KEYBOARD DRIVER
AE26 CD E631 C CALL SETINP ; ADDRESS
AE29 11 E01B C LD DE, VIDEO ; SET UP VIDEO SCREEN
AE2C CD E615 C CALL SETOUT ; DRIVER ADDRESS
AE2F 21 C000 C LD HL, MCP ; POINT TO MCP ENTRY POINT
AE32 7E C LD A, (HL) ; GET FIRST INSTRUCTION
AE33 FE C3 C CP 0C3H ; TEST FOR JUMP INSTRUCTION
AE35 28 0E C JR Z, COLD02 ; JUMP IF MCP IS LOADED
AE37 3E 0C C LD A, FF ; LOAD FORM FEED
AE39 CD E01B C CALL VIDEO ; CLEAR SORCERER SCREEN
AE3C 21 A07B C LD HL, MSGMCP ; ‘MCP NOT LOADED’
AE3F CD BE11 C CALL MONPRT ; DISPLAY MESSAGE
AE42 C3 E003 C JP MONITR ; RETURN TO SORCERER MONITOR
AE45 21 00008 C COLD02: LD HL, SAVCMD ; INITIALIZE SAVE COMMAND
AE48 3E 20 C LD A, SPACE ; WITH INTER-SPACES
AE4A 06 12 C LD B, 12H ; LENGTH OF COMMAND
AE4C 77 C COLD03: LD (HL), A
AE4D 23 C INC HL
AE4E 10 FC C DJNZ COLD03 ; END COMMAND WITH CR
AE50 36 8D C LD (HL), CR ; END COMMAND WITH CR
AE52 61 0008 C LD BC, 0008H ; LENGTH OF DATA TO MOVE
AE55 11 FFF0 C LD DE, 0FF0H ; DESTINATION ADDRESS
AE58 21 A02D C LD HL, SPCHAR ; SPECIAL CHARACTERS ADDRESS
AE5B ED B9 C LDIR ; MOVE SPECIAL GRAPHIC CHARACTERS
AE5D IE 85 C LD E, 65H ; NUMBER OF PARTITIONS TO DEFINE
AE5F 21 A083 C LD HL, PART01 ; POINT TO FIRST PDB
AE62 E5 C COLD04: PUSH HL ; PUT ADDRESS OF PDB IN (BC)
AE63 C1 C POP BC
AE64 3E 00 C LD A, DEFPAR ; DEFINE PARTITION FUNCTION CODE
AE66 D5 C PUSH DE ; SAVE COUNT
AE67 CD C800 C CALL MCP
AE69 D1 C POP DE ; RESTORE COUNT
AE6B 01 0008 C LD BC, PDBLEN ; LOAD LENGTH OF PDB
AE6E 09 C ADD HL, BC ; POINT TO NEXT PDB
AE6F 1D C DEC E ; COUNT PARTITIONS
AE70 20 F0 C JR NZ, COLD04 ; JUMP IF NOT FINISHED
AE72 21 0000X C LD HL, RECBUF ; START TAPE RECORD BUFFER END
AE75 22 0000X C LD (RECEND), HL ; AT BEGINNING OF BUFFER
AE78 21 1936 C LD HL, 1936H ; STARTING SEED VALUE FOR
AE7B 22 0000X C LD (SEED1), HL ; RANDOM NUMBER GENERATOR
AE7E 22 0000X C LD (SEED2), HL
AE81 CD AF76 C CALL GETDAT ; GET TODAY'S DATE
AE84 CD AFA8 C CALL LODUSR ; LOAD USER FILE
AE87 CD B0DF C CALL SETTG ; LOAD TUNE GROUP
AE8A C3 AE1 C JP SSM ; START PROGRAM

; ; WARM
; ; WARM START INITIALIZATION
; ; GIVEN: NONE
; ; RETURNS: (IY) - ADDRESS OF TUNE GROUP HEADER
; ; WARN: PUSH DE
AE8E FD 21 0000X C LD IY, TUNGRP ; POINT TO TUNE GROUP HEADER
AE91 FD 7E 07 C LD A, (IY + RATONE) ; GET STARTING RATING
AE95 32 0000X C LD (USRRA1), A ; START USER AT THIS RATING
AE98 CD BCBD C CALL INITT ; INITIALIZE ALL TUNES AS UNUSED
AE9B AF C XOR A ; RESET TUNE COUNT
AE9C 32 0000X C LD (TUNCNT), A
AE9F 10 CD AEA1 C
SIGHTSINGER'S MIRROR PROGRAM SECTION

AEA0  C9  C  RET

; SIGHTSINGER'S MIRROR (SSM)
; CONTROLS LOGON PROCEDURE, GETS USER'S RANGE, AND CONTROLS EACH OPTION OF THE MAIN MENU.

AEA1  06  FF  C  SSM:  LD  B,0FFH  ; CLEAR ENTIRE SCREEN
AEA2  3E  82  C  LD  A,CLRPAR  
AEA3  CD  C080  C  CALL  MCP  
AEA4  81  8204  C  LD  BC,0204H  ; SET SORCERER CURSOR POSITION
AEA5  CD  DE5D  C  CALL  SETCUR  
AEA6  CD  AEF4  C  CALL  LOGON  ; WAIT FOR USER TO LOGON
AEA7  FE  83  C  CP  ETX  ; CONTROL/C
AEA8  DA  E003  C  JP  Z,MONITR  ; RETURN TO MONITOR
AEA9  CD  AFE8  C  CALL  LOGON  ; GET USER'S RANGE
AEBA  A7  C  AND  A  ; TEST FOR START OVER
 AEBC  3E  00  C  LD  A,TYPUSR  ; CREATE USER RECORD
AEBD  CD  BB37  C  CALL  CREREC
AECE  CD  AE8D  C  SSM02:  CALL  WARM
AEED  1A  A41C  C  LD  DE,MSGMOD  ; ASK FOR MODE
AEF0  CD  B023  C  CALL  QRYOPT
AEF1  FE  52  C  CP  'R'  ; RANGE
AEF2  28  E8  C  JR  Z,SSM01  ; JUMP IF HE WANTS TO START OVER
AEF3  FE  54  C  CP  'T'  ; TUNES
AEF4  28  0E  C  JR  Z,SSM04  ; FREE MODE
AEF5  FE  46  C  CP  'F'  ; FREE MODE
AEF6  28  05  C  JR  Z,SSM03  
AEF7  CD  BC19  C  CALL  WRTBUF  ; ELSE Flush BUFFER
AEF8  18  C6  C  JR  SSMB  ; AND LOGOFF
AEF9  CD  B1A4  C  SSM03:  CALL  CLTFRE  ; RUN FREE MODE
AEF0  18  E1  C  JR  SSMB2
AEF1  CD  B0DF  C  SSM04:  CALL  SETT6  ; LOAD TUNE GROUP
AEF2  A7  C  AND  A  ; TEST RETURN CODE
AEF3  28  08  C  JR  NZ,SSM02  ; MAIN MENU IF NOT ZERO
AEF4  3E  08  C  LD  A,TYPGRP  ; CREATE TUNE GROUP RECORD
AEF5  CD  BB37  C  CALL  CREREC
AEF6  FD  21  0000  C  LD  IY,TUNGRP
AEF7  CD  B2E3  C  CALL  CTLTUN  ; RUN TUNE MODE
AEF8  18  CD  C  JR  SSMB2

; LOGON Procedure. Does not return until either an
C ; AUTHORIZED USER LOGS ON OR A CONTROL/C IS ENTERED.
C ;
C ; GIVEN: NONE
C ; RETURNS: (A) - COMPLETION CODE
C ;
C ; = 00H - AUTHORIZED USER
C ;
C ; = 03H - CONTROL/C - RETURN TO MONITOR
C ;

AEF4 CD E1A2 C LOGON: CALL FNDUSRAA ; SET (IY) TO MONITOR AAA
AEF7 21 A283 C LD HL, MSGLOG
AEFA CD BE11 C CALL MONPRT ; DISPLAY 'LOGON'
AEF0 3E 7F C LD A, FLDFLG ; TRANSLATE LOWER TO UPPER CASE
AEFF CD BE1C C CALL RESPON ; WAIT FOR RESPONSE
AF02 7E C LD A, (HL) ; GET FIRST CHARACTER
AF03 FE 03 C CP ETX ; TEST FOR CONTROL/C
AF05 28 41 C JR Z, LOGN03 ; EXIT IF FOUND
AF07 06 05 C LD B, 05H ; LENGTH OF NAME FIELD
AF09 CD AF61 C CALL SPCFIL ; PAD ON THE RIGHT WITH SPACES
AF0C EB C EX DE, HL ; GET ADDRESS OF NAME IN (DE)
AF0D CD AF49 C CALL FNDUSR ; SEARCH TABLE FOR NAME
AF10 28 2D C JR NZ, LOGN01 ; JUMP IF NOT FOUND
AF12 22 0000X C LD (USRNAME), HL ; ELSE SAVE ADDRESS OF NAME
AF15 11 0005 C LD DE, 0005H ; LENGTH OF NAME
AF18 19 C ADD HL, DE ; POINT TO PASSWORD
AF19 E5 C PUSH H ; SAVE ADDRESS
AF1A 21 A2C1 C LD HL, MSGPSW ; ASK FOR PASSWORD
AF1D CD BE11 C CALL MONPRT
AF20 11 BEF8 C LD DE, NULDEV ; SET OUTPUT DEVICE NULL SO
AF23 CD E615 C CALL SETOUT ; PASSWORD WILL NOT BE SHOWN
AF26 3E 7F C LD A, FLDFLG ; TRANSLATE LOWER TO UPPER CASE
AF28 CD BE1C C CALL RESPON ; GET PASSWORD
AF2B 11 EB18 C LD DE, VIDEO ; RESET OUTPUT DEVICE DRIVER
AF2E CD E615 C CALL SETOUT
AF31 06 0B C LD B, 08H ; LENGTH OF PASSWORD FIELD
AF33 CD AF61 C CALL SPCFIL ; PAD ON RIGHT WITH SPACES
AF36 01 C POP DE ; RESTORE ADDRESS OF PASSWORD
AF37 81 0088 C LD BC, 0088H ; LENGTH OF PASSWORD
AF3A CD BF50 C CALL COMPAR ; TEST PASSWORD
AF3D 28 08 C JR Z, LOGN02 ; EXIT IF CORRECT
AF3F 21 A2CD C LOGN01: LD HL, MSGUAU ; 'UNAUTHORIZED USER'
AF42 CD BE11 C CALL MONPRT ; DISPLAY MESSAGE
AF45 18 AD C JR LOGON ; RESTART
AF47 AF C LOGN02: XOR A ; SET AUTHORIZED USER
AF48 C9 C LOGN03: RET
C ;
C ; FINDS USER NAME IN LOG TABLE AND EITHER SETS OR RESETS
THE ZERO FLAG ACCORDING TO WHETHER THE NAME WAS FOUND IN THE TABLE.

; GIVEN: (DE) - ADDRESS OF USER NAME
; RETURNS: (HL) - ADDRESS OF USER NAME IF FOUND
; Z - FOUND/NOT FOUND INDICATOR
; SET IF USER NAME FOUND
; RESET IF USER NAME NOT FOUND

FDUSR: PUSH BC
LD HL,LOGTBL ; POINT TO LOG TABLE
FNUS01: LD BC,0005H ; LENGTH OF USER NAME
CALL COMPAR ; COMPARE NAMES
JR Z,FNUS82 ; JUMP IF FOUND
LD BC,008DH ; POINT TO NEXT USER
ADD HL,BC ; NAME IN TABLE
LD A,(HL) ; END OF TABLE IS DENOTED
AND A ; BY A VALUE OF ZERO
JR NZ,FNUS01 ; JUMP IF NOT END OF TABLE
CP 0FFH ; ELSE RESET ZERO FLAG
POP BC
RET

SPCFIL
; PADS THE FIELD WHOSE ADDRESS AND LENGTH ARE GIVEN WITH SPACES ON THE RIGHT. THE STARTING POSITION IS INDICATED WHEN A CARRIAGE RETURN IS FOUND.
; GIVEN: (B) - LENGTH OF FIELD
; (HL) - ADDRESS OF FIELD
; RETURNS: NONE

SPCFIL: PUSH BC
PUSH HL
LD A,CARRET ; SEARCH FOR CARRIAGE RETURN
Spf101: CP (HL)
JR Z,Spf102 ; JUMP WHEN FOUND
INC HL
DJNZ Spf101
JR Spf104 ; EXIT IF NOT FOUND
LD A,SPACE
Spf102: LD (HL),A ; STORE SPACES
INC HL
DJNZ Spf102
INC HL
DJNZ Spf104
LD HL,Spf103
POP HL
POP BC
RET

GETDAT

GETS TODAY'S DATE FROM OPERATOR AND STORES IT IN THE 3 BYTES AT 'DATE'. NO ERROR CHECKING IS DONE.

GIVEN: NONE

RETURNS: NONE

GETDAT: LD B,01H ;ACTIVATE AND CLEAR SCREEN
CALL ANCPAR
LD BC,0104H ;SET MONITOR CURSOR POSITION
CALL SETCUR
LD A,83H
LD (TEMP01),A ;NUMBER OF BYTES
LD DE,DATE
LD HL,MSGPET
CALL MONPRT
LD A,FLDFLG
CALL RESPON
GTDT01: LD B,(HL) ;GET FIRST OF DIGIT PAIR
INC HL ;POINT TO NEXT CHARACTER
LD C,(HL) ;GET SECOND DIGIT OF PAIR
INC HL ;POINT TO SEPARATOR
INC HL ;POINT TO NEXT DIGIT PAIR
LD DE,(DE),A ;STORE IT
LD A,FLDFLG
JR NZ,GTDT01 ;JUMP IF NOT FINISHED
HOLE SCREEN PARTITION
ACTIVATE AND CLEAR IT

LODUSR: PUSH HL
LD B,01H ;WHOLE SCREEN PARTITION
CALL ANCPAR ;ACTIVATE AND CLEAR IT
LD BC,0102H ;X/Y COORDINATE
CALL SETCUR ;PUT CURSOR THERE
LD HL,MSGPET ;'LOAD USER FILE?'
CALL MONPRT ;DISPLAY MESSAGE
PROGRAM SECTION

AFBA 3E 7F  C  LD  A,FLDFLG  ;LOWER TO UPPER CASE
AFBC CD BE1C  C  CALL  RESPON  ;GET ANSWER
AFBF 7E  C  LD  A,(HL)
AFCA 7E  C  LD  A,(K L)
AFCD CD E009  C  CALL  GETCHR  ;WAIT TILL READY
AFDA CD E205  C  CALL  KILCUR  ;ERASE CURSOR
AFDA 7E  C  CALL  KILCUR  ;ERASE CURSOR
AFEE C5  C  GETRN:  PUSH  BC
AFEC D5  C  GETRN:  PUSH  DE
AFED E5  C  GETRN:  PUSH  HL
AFAE 86 00  C  LD  B,00H  ;NO OFFSET
AFAF 3E 26  C  LD  A,SETOFS  ;SET OFFSET FUNCTION CODE
AFAF CD C000  C  CALL  MCP
AFAD 21 A308  C  GTRN01:  LD  HL,MSGFXN  ;'FIND HIGHEST/LOWEST PITCH'
AFB0 CD BEF9  C  CALL  MOVMSG  ;MOVE TO RAM
AFBB 01 0007  C  LD  BC,0007H  ;LENGTH OF TEXT TO MOVE
AFBA 11 0009X  C  LD  DE,THMSG+09H  ;DESTINATION OF MOVE
B001 21 A2FA  C  LD  HL,LOWEST  ;TEXT TO MOVE
SIGHTSINGER'S MIRROR MACRO-88 3.44 09-Dec-81 PAGE I-36
PROGRAM SECTION

B084  ED B9  C  LDIR ;MOVE 'LOWEST' TO MESSAGE
B086  CD B05F  C  CALL QRYRNG ;ASK FOR LOWEST PITCH
B089  FE FD  C  CP KBINTR ;TEST FOR USER INTERRUPT
B08B  28 4E  C  JR 2,GTRN86 ;JUMP IF USER STOPS
B08D  32 0000X  C  LD (USRLOH),A ;ELSE STORE LOWEST PITCH
B090  01 0007  C  LD BC,0007H ;LENGTH OF NEXT TEXT MOVE
B093  11 0009X  C  LD DE,TMPMSG+09H ;DESTINATION OF MOVE
B096  21 A301  C  LD HL,HIGHEST ;TEXT TO MOVE
B099  ED B0  C  LDIR ;MOVE 'HIGHEST' TO MESSAGE
B09B  CD B05F  C  CALL QRYRNG ;ASK FOR HIGHEST PITCH
B09E  FE FD  C  CP KBINTR ;TEST FOR USER INTERRUPT
B0A0  28 39  C  JR 2,GTRN82 ;EXIT ON INTERRUPT
B0A3  32 0000X  C  GTRN82: LD (USRHI),A ;ELSE STORE HIGH PITCH
B0A5  47  C  LD B,A ;SAVE IT
B0A6  3A 0000X  C  LD A,(USRLOH) ;GET USER'S LOW NOTE
B0A9  90  C  SUB B ;GET DIFFERENCE
B0AA  ED 44  C  MOVE IT POSITIVE
B0AB  CB 7F  C  BIT 7,A ;MAKE SURE HIGH IS ABOVE LOW
B0AC  28 10  C  JR 2,GTRN83 ;JUMP IF IT IS
B0AD  01 0111  C  LD BC,0111H ;X/Y COORDINATE TO PRINT MESSAGE
B0AE  11 A30A  C  LD DE,MSGCF 'YOU CAN'T FOOL ME'
B0B1  3E 34  C  LD A,PRTEXT PRINT TEXT FUNCTION CODE
B0B4  CD C000  C  CALL MCP ;DISPLAY MESSAGE
B0B7  CD BEA3  C  CALL WAIT ;WAIT TILL USER IS READY
B0BB  18 B5  C  JR GTNR01 ;START OVER
B0BC  32 0000X  C  GTRN03: LD (USRANG),A ;STORE USER'S RANGE
B0BE  30 0D  C  CP 0DH ;TEST FOR MINIMUM OF 1 OCTAVE
B0C0  30 13  C  JR NC,GTRN05 ;EXIT IF LARGER
B0C3  01 0111  C  LD BC,0111H ;X/Y COORDINATE OF MESSAGE
B0C6  11 A398  C  LD DE,MSGCFR 'THAT'S NOT A VERY WIDE RANGE'
B0C9  3E 34  C  LD A,PRTEXT PRINT TEXT FUNCTION CODE
B0CD  CD C000  C  CALL MCP ;PRINT IT
B0D2  CD BEA3  C  GTRN04: CALL WAIT ;WAIT TILL USER IS READY
B0D5  86 01  C  LD B,01H ;WHOLE SCREEN PARTITION
B0D7  CD BD92  C  CALL ANCPAR ;CLEAR THE SCREEN
B0DA  AF  C  GTRN05: XOR A ;ZERO RETURN CODE
B0DB  EI  C  GTRN06: POP HL
B0DC  D1  C  POP DE
B0DD  C1  C  POP BC
B0DE  C9  C  RET

; QUERY USER FOR RANGE. THE USER IS ASKED TO SING EITHER
; THE HIGHEST OR LOWEST PITCH HE FEELS COMFORTABLE SINGING
; FOR THE SESSION. THE MESSAGE STORED AT 'MSGCFX' IS
; DISPLAYED TO REQUEST THE PITCH FROM THE USER. THE PITCH
C IS PLAYED BACK FOR HIS CONFIRMATION TO ALLOW HIM THE
C OPTION OF CHANGING IT.
C
C GIVEN: NONE
C
C RETURNS: (A) - INTEGER PITCH SUNG BY USER
C
C = KBINTR - USER INTERRUPT
C

B05F C5 C QRYANG: PUSH BC
B068 D5 C PUSH DE
B061 E5 C PUSH HL
B062 3E 2C C QYRN01: LD A,INITPT ;INITIALIZE INPUT PITCH ROUTINE
B064 CD C000 C CALL MCP
B067 06 01 C LD B,01H ;PARTITION NUMBER TO USE
B069 CD BD92 C CALL ANCPAR ;ACTIVATE AND CLEAR THIS PARTITION
B06C 3E 02 C LD A,02H ;DELAY FACTOR
B06E CD BECD C CALL DELAY ;ALLOW TIME FOR USER TO SEE CHANGE
B071 01 0103 C LD BC,0103H ;LOAD STARTING X/Y COORDINATE
B074 11 0000X C LD DE,TMPMSG ;'FIND HIGHEST/LOWEST PITCH...'
B077 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
B079 CD C000 C CALL MCP ;PRINT THE MESSAGE
B07C CD E009 C CALL GETCHR ;WAIT TILL USER IS READY
B07F 01 0108 C LD BC,0108H ;NEXT LINE/COLUMN FOR MESSAGE
B082 11 A359 C LD DE,MSGSTP ;'SING THAT PITCH'
B085 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
B087 CD C000 C CALL MCP ;PRINT MESSAGE
B08A 06 FF C QYRN02: LD B,NOISE ;LISTEN FOR PITCH
B08C 3E 27 C LD A,INITPT ;LISTEN UNTIL USER SINGS NOTE
B08E CD C000 C CALL MCP
B091 FE FD C CP KBINTR ;CHECK FOR USER INTERRUPT
B093 28 46 C JR Z,QYRN05 ;JUMP ON INTERRUPT
B095 FE FF C CP NOISE ;CHECK FOR NOISE
B097 28 F1 C JR Z,QYRN02 ;KEEP LISTENING IF NOISE
B099 06 04 C LD B,04H ;PREPARE TO PLAY BACK PITCH
B09B 21 0000X C LD HL,WKSPC ;SET UP PITCH IN WORK SPACE
B09E E5 C CALL MCP ;SAVE ADDRESS OF NOTE
B09F 36 01 C LD (HL),01H ;NUMBER OF NOTES TO PLAY
B0A1 23 C QYRN03: INC HL ;POINT TO NEXT VOICE
B0A2 77 C LD (HL),A ;STORE PITCH FOR NEXT VOICE
B0A3 10 FC C DJNZ QYRN03 ;CONTINUE FOR 4 VOICES
B0A5 23 C INC HL ;POINT TO RHYTHM BYTE
B0A6 36 10 C LD (HL),10H ;MAKE IT A HALF NOTE
B0A8 01 010B C LD BC,0108H ;LINE/COLUMN
B0AA 11 A36A C LD DE,MSGSSI ;'STOP SINGING'
B0AB 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
B0AD CD C000 C CALL MCP ;DISPLAY MESSAGE
B0B3 3E 03 C LD A,03H ;DELAY FACTOR
B0B5 CD BECD C CALL DELAY ;GIVE USER TIME TO STOP
B0B8 01 003C C LD BC,003CH ;SET ROOT NOTE AND TEMPO
B08B D1 C POP DE ; RESTORE ADDRESS OF NOTE
B08C 3E 24 C LD A,PLAREL ; PLAY NOTES RELATIVE FUNCTION CODE
B08E CD C800 C CALL MCP ; PLAY THE SAME PITCH
B0C1 01 010E C LD BC,010EH ; LINE/COLUMN OF MESSAGE
B0C4 11 A378 C LD DE,MSGTR ; 'IS THIS THE RIGHT PITCH?'
B0C7 3E 34 C LD A,PRTEXT ; PRINT TEXT FUNCTION CODE
B0C9 CD C800 C CALL MCP ; DISPLAY MESSAGE
B0CC CD BEDD C CALL KBFOLD ; WAIT FOR USER CONFIRMATION
B0CF FE 59 C CP 'Y' ; DOES HE SAY IT IS RIGHT?
B0C1 2B 05 C JR Z,QYRN04 ; JUMP IF HE DOES
B0D3 FE 0D C CP CARRET ; TEST FOR CARRIAGE RETURN
B0D5 C2 B862 C JP N2,QYRN01 ; START OVER IF NOT RIGHT
B0D8 3B 0001X C QYRN04: LD A,(WRKSPC+01H) ; GET INTEGER PITCH
B0DB E1 C QYRN05: POP HL
B0DC D1 C POP DE
B0DE C9 C POP BC
B0DF C5 C SETTG: PUSH BC
B0E0 D5 C PUSH DE
B0E1 E5 C PUSH HL
B0E2 06 01 C SETG01: LD B,01H ; CLEAR ENTIRE SCREEN
B0E4 CD B92 C CALL ANCPAR
B0E7 01 0103 C LD BC,0103H ; X/Y COORDINATE
B0EA CD BE5D C CALL SETCUR ; MOVE CURSOR
B0ED CD B187 C CALL CHNNAM ; CHECK FOR VALID NAME
B0F0 2B 12 C JR Z,SETG02 ; JUMP IF VALID
B0F2 21 A1AA C LD HL,MSGTRG ; 'NO TUNE GROUP IS LOADED'
B0F5 CD BE11 C CALL MNPRT
B0F8 3E 7F C LD A,FLDFLG
B0FA CD BE1C C CALL RESPON
B0FD 7E C LD A1,(HL) ; GET RESPONSE
B0FE FE 59 C CP 'Y' ; EXIT IF NOT YES
B100 2B 2B C JR Z,SETG03 ; ELSE LOAD TUNE GROUP
B102 1B 3E C JR SETG04
B104 21 A1F6 C SETG02: LD HL,MSGTGX ; 'TUNE GROUP IS NOW LOADED'
B107 CD BEF9 C CALL MOVMSG ; MOV TO RAM
LOAD DATA FROM TAPE UNIT 01. BAUD RATE IS SET TO 300.
THE FIRST FILE FOUND IS LOADED AT THE ADDRESS GIVEN
REGARDLESS OF THE LOAD ADDRESS ON THE TAPE ITSELF.
GIVEN: (HL) - ADDRESS OF DESTINATION OF LOAD
(IX) - ADDRESS OF MONITOR MNA
RETURNS: NONE
LOAD: PUSH AF
```assembly
B150  C5  C  PUSH  BC
B15E  D5  C  PUSH  DE
B15F  E5  C  PUSH  HL
B160  DD  E5  C  PUSH  IX
B162  FD  E5  C  PUSH  IY
B164  CD  E1A2  C  CALL  FINDMIA  ;SET (IY) TO MIA
B167  EB  C  EX  DE,HL  ;PASS ADDRESS IN (DE)
B168  FD  36  3D  00  C  LD  (IY+3DH),00H  ;SET BAUD RATE TO TAPE/300
B16C  06  01  C  LD  B,01H  ;TAPE UNIT NUMBER 1
B16E  21  B17E  C  LD  HL,LOAD01  ;PUT RETURN ADDRESS ON STACK
B171  E5  C  PUSH  HL
B172  AF  C  XOR  A  ;NOT ZERO FOR NON
B173  FE  7F  C  CP  7FH  ;AUTO EXECUTE FILE
B175  F5  C  PUSH  AF
B176  AF  C  XOR  A  ;ZERO FOR NO NAME MATCH
B177  F5  C  PUSH  AF
B178  FE  7F  C  CP  87FH  ;NOT ZERO FOR LOAD
B17A  F5  C  PUSH  AF  ;ADDRESS GIVEN IN (DE)
B17B  C3  E7B8  C  JP  TAPLOD  ;CALL MONITOR LOAD ROUTINE
B17E  FD  E1  C  LOAD01:  POP  IY
B180  DD  E1  C  POP  IX
B182  E1  C  POP  HL
B183  D1  C  POP  DE
B184  C1  C  POP  BC
B185  F1  C  POP  AF
B186  C9  C  RET
C
C  ;CHRONAM
C  ;CHECK TUNE GROUP NAME FOR PRINTABLE CHARACTERS AS
C  ;A TEST TO SEE IF A TUNE GROUP HAS BEEN LOADED.
C  ;ANY CHARACTERS BETWEEN '0' AND 'Z' ARE CONSIDERED
C  ;TO BE VALID. SPACES ARE VALID EXCEPT AS THE FIRST
C  ;CHARACTER.
C  ;GIVEN:  NONE
C  ;RETURNS: Z - ZERO FLAG VALIDITY INDICATOR
C  ;       SET - VALID
C  ;       RESET - INVALID
C  ;
B187  C5  C  CHRONAM:  PUSH  BC
B188  E5  C  PUSH  HL
B189  06  05  C  LD  B,05H
B18B  21  008EX  C  LD  HL,TUNGRP  ;POINT TO TUNE GROUP NAME
B18E  7E  C  LD  A,(HL)  ;GET FIRST CHARACTER
B19F  18  05  C  JR  CHRON02  ;PASS OVER SPACE TEST
B191  7E  C  CHRON01:  LD  A,(HL)  ;GET CHARACTER
```
PROGRAM SECTION

B192  FE 20  C  CP   SPACE ; TEST FOR SPACE
B194  28 86  C  JR   Z,CKM83 ; OK IF SPACE
B196  D6 30  C  CKM82; SUB 30H ; SHIFT VALID RANGE TO
B198  FE 48  C  CP   48H ; 00 - 4AH
B19A  30 04  C  JR   NC,CKM404 ; JUMP IF INVALID
B19C  23 C  CKM83; INC HL ; POINT TO NEXT CHARACTER
B19D  10 F2 C  DJNZ CKM101
B19F  AF C  XOR A ; SET VALID FLAG
B1A6  A7 C  CKNM84; AND A ; SET/RESET ZERO FLAG
B1A1  E1 C  POP HL
B1A2  C1 C  POP BC
B1A3  C9 C  RET

; MACLIB B:SSM02.MAC
; CTLFRE
; CONTROL FREE MODE.
; GIVEN: NONE
; RETURNS: (A) - INTERRUPT KEYBOARD CHARACTER

B1A4  C5 C  CTLFRE: PUSH BC
B1A5  D5 C  PUSH DE
B1A6  E5 C  PUSH HL
B1A7  CD B1B8 C  CALL QRYFRE ; GET PARAMETERS
B1AA  FE 1B C  CP   ESC ; CHECK FOR EXIT
B1AC  28 86 C  JR   Z,CTFR81 ; JUMP TO EXIT
B1AE  CD B240 C  CALL INITFM ; INITIALIZE FIELDS
B1B1  CD B283 C  CALL FREMOD ; LISTEN AND DISPLAY
B1B4  E1 C  CTFR81; POP HL
B1B5  D1 C  POP DE
B1B6  C1 C  POP BC
B1B7  C9 C  RET

; QUERY FREE MODE PARAMETERS. ASKS FOR USER'S CHOICE OF
; CLEF AND KEY.
; GIVEN: NONE
; RETURNS: (A) - COMPLETION CODE
; = 08H - SUCCESSFUL
; = ESC - USER KEYBOARD INTERRUPT


MACRO-88 3.44 09-Dec-81

SIGHTSINGER'S MIRROR

PROGRAM SECTION

B226  CD C000  C  CALL MCP
B229  A7  C  AND A  ;TEST FOR INVALID KEY
B22A  28 10  C  JR Z,GYFR05  ;JUMP IF OK
B22C  01 000F  C  LD BC,000FH  ;X/Y COORDINATE
B22F  11 ACCE  C  LD DE,MSGWAY  ;"WHO ARE YOU TRYING TO KID?"
B232  3E 34  C  LD A,PRTEXT  ;DISPLAY MESSAGE
B234  CD C000  C  CALL MCP
B237  CD BE30  C  CALL WAIT
B23A  18 96  C  JR QYFR02  ;DO KEY AGAIN
B23C  E1  C  QYFR05: POP HL
B23D  DI  C  POP DE
B23E  C1  C  POP BC
B23F  C9  C  RET

C  ;
C  ; INITFM
C  ; INITIALIZE FREE MODE. ASSUMES THAT 'FMCLEF' IS
C  ; INITIALIZED AND THE KEY HAS BEEN SET IN MCP.
C  ; GIVEN:  NONE
C  ; RETURNS:  NONE
C  ;
B240  C5  C  INITFM: PUSH BC
B241  00 01  C  LD B,01H  ;CLEAR THE SCREEN
B243  3E 02  C  LD A,CLRPAR
B245  CD C000  C  CALL MCP
B248  01 141E  C  LD BC,141EH  ;X/Y COORDINATE
B24B  11 A00C  C  LD DE,MSGAKS  ;"PRESS ANY KEY TO STOP"
B24E  3E 34  C  LD A,PRTEXT  ;DISPLAY TEXT
B250  CD C000  C  CALL MCP
B253  3E 33  C  LD A,ENASTO  ;ENABLE STORE IN TUNE TABLE
B255  CD C000  C  CALL MCP
B258  3E 02  C  LD A,02H
B25A  32 0000X  C  LD (PARNUM),A  ;START OFF IN TOP PARTITION
B25D  47  C  LD B,A  ;ACTIVATE TOP PARTITION
B25E  3E 01  C  LD A,ACTPAR
B260  CD C000  C  CALL MCP
B263  06 04  C  LD B,04H  ;STAFF ORIENTATION
B265  3A 0000X  C  LD A,(FM CLEF)  ;GET CLEF
B268  4F  C  LD C,A
B269  3E 04  C  LD A,STFCLEF  ;DISPLAY STAFF AND CLEF
B26B  CD C000  C  CALL MCP
B26E  3E 0A  C  LD A,KEYSIG  ;DISPLAY KEY SIGNATURE
B270  CD C000  C  CALL MCP
B273  AF  C  XOR A  ;RESET INTEGER PITCH OFFSET
B274  32 0000X  C  LD (1OFSET),A
B277  3E 2C  C  LD A,INITPT  ;INITIALIZE INPUT PITCH
CALL MCP
LD A,NOISE
LD (PREVIP),A
POP BC
RET

FREMQD:

LETS USER SING ANYTHING HE DESIRES AND DISPLAYS IT.
DOES NOT RETURN UNTIL USER PRESSES A KEY.

GIVEN: NONE
RETURNS: (A) INTERRUPT KEYBOARD CHARACTER

FREMQD: PUSH BC
PUSH DE
PUSH HL
LD L,00H
LD H,NPM
FRMD01: CALL LISTEN
CP KBINTR
JR Z.FRMD05
LD B,A
PUSH BC
PUSH DE
LD B,03H
LDA .tPAIM JM)
CP 02H
JR 2,FRMD04
LD B,03H
LD A,CLRPAR
CALL MCP
LD B,02H
JR FRMD01
FRMD02: LD A,PITCHI
DISPLAY PITCH

FRMD03: PUSH BC
SAVE UNDISPLAYED PITCH
PUSH DE
DATA
LD B,03H
SET UP OTHER PARTITION
LD A,(PARNUM)
GET CURRENT PARTITION
CP 02H
TEST FOR TOP PARTITION
JR 2,FRMD04
SET UP TOP PARTITION
LD B,03H
ELSE CLEAR BOTTOM ONE
LD A,CLRPAR
CALL MCP
LD B,02H
SET UP TOP PARTITION NEXT
PROGRAM SECTION

B2C1 78 C FRMD84: LD A,B ; STORE NEW CURRENT PARTITION
B2C2 32 0000x C LD (PARNUM),A ; NUMBER
B2C5 3E 01 C LD A,ACTPAR ; NOW ACTIVATE IT
B2C7 CD C000 C CALL MCP
B2CA 86 04 C LD B,04H ; STAFF ORIENTATION
B2CC 3A 0000x C LD A,(FMCLEF) ; CLEF IN USE
B2CF 4F C LD C,A
B2D0 3E 04 C LD A,STFCLF ; DISPLAY STAFF AND CLEF
B2D2 CD C000 C CALL MCP
B2D5 3E 0A C LD A,KEYSIG ; DISPLAY KEY SIGNATURE
B2DA U64 C LD B,04H ; STAFF ORIENTATION
B2D5 3E 0A C LD A,KEYSIG ; DISPLAY KEY SIGNATURE
B2D7 CD C000 C CALL MCP
B2DA D1 C POP DE ; RESTORE PITCH DATA
B2DB C1 C POP BC
B2DC 18 B8 C JR F H 0 8 2 ; JUMP TO DISPLAY PITCH
B2DE 79 C FRMD85: LD A,C ; GET KEYBOARD CHARACTER
B2DF E1 C POP HL
B2E0 D1 C POP DE
B2E1 C1 C POP BC
B2E2 C9 C POP BC
B2E4 C5 C ; MACLIB B:SSM03.MAC
B2E5 C5 C ; CTLTUN
B2E6 86 FF C ; CONTROL THE SELECTION, DISPLAY AND PERFORMANCE OF THE TUNE.
B2E8 3E 02 C ; ASSUMES THAT THE USER'S RATING, HIGHEST PITCH, LOWEST PITCH
B2ED CD C000 C ; AND RANGE HAVE BEEN INITIALIZED.
B2F0 D8 C ; GIVEN: (1Y) — ADDRESS OF TUNE GROUP HEADER
B2F4 CD B4E5 C ; RETURNS: NONE
B2F7 A7 C CTLTUN: PUSH BC
B2F8 28 3B C PUSH DE
B2FA FE 08 C PUSH HL
B2FC 28 16 C CTUN01: LD B,0FFH ; CLEAR ENTIRE SCREEN
B2F7 A7 C CTUN02: LD A,CLRPAR
B2F8 28 3B C CALL MCP
B2F9 8E 08 C ACTPAR ; ACTIVATE PARTITION IN WHICH
B2F1 CD C000 C ; TUNE WILL BE DISPLAYED
B2FF CD B4E5 C CALL SELTUN ; SELECT A TUNE
B2F7 A7 C AND A ; CHECK RETURN CODE
B2F8 28 3B C JR Z,CTUN05 ; JUMP IF GOOD TUNE FOUND
B2FA FE 08 C CP 08H ; CHECK OTHER RETURN CODES
B2FC 28 16 C JR NZ,CTUN02 ; JUMP IF END OF TUNE GROUP
B2FE 11 A4CD C LD DE,MSGMT ; 'NO MORE SUITABLE TUNES'
PROGRAM SECTION

B301 CD B023 C CALL QRYOPT ; ASK WHAT TO DO
B304 FE 41 C CP 'A' ; START AT THE BEGINNING?
B306 28 25 C JR Z,CNTUN04 ; JUMP TO START ALL OVER
B308 FE 45 C CP 'E' ; EXPAND HIS RANGE?
B30A 20 2F C JR NZ,CNTUN06 ; EXIT IF NOT
B30C CD AE8D C CALL GETRNG ; ELSE GET HIS RANGE AGAIN
B30F A7 C AND A ; TEST RETURN CODE
B310 28 D4 C JR Z,CNTUN01 ; START AGAIN IF ZERO
B312 18 27 C JR CNTUN06 ; ELSE EXIT
B314 11 A5D6 C CNTUN02: LD DE,MSGETG ; 'T H A T 'S ALL THE TUNES I HAVE'
B317 CD B023 C CALL QRYOPT ; ASK WHAT TO DO
B31A FE 4C C CP 'L' ; LOAD NEW TUNE GROUP?
B31C 20 0B C JR NZ, CNTON03 ; JUMP IF NOT
B31E CD B0DF C CALL SETT6 ; GET NEW TUNE GROUP
B321 A7 C AND A ; TEST RETURN CODE
B322 28 17 C JR NZ,CNTUN06 ; EXIT IF NOT ZERO
B324 CD AE8D C CALL WARM ; RE-INITIALIZE
B327 18 B0 C JR CNTUN01 ; RESTART
B329 FE 41 C CNTUN03: CP 'A' ; DO IT AGAIN?
B32B 28 0E C JR NZ,CNTUN06 ; EXIT IF NOT
B32D CD BCB0 C CNTUN04: CALL INITTG ; ELSE INITIALIZE TUNE GROUP
B330 CD B4BC C CALL FIXRAT ; ADJUST HIS RATING
B333 18 B1 C JR CNTUN01 ; AND START OVER
B335 CD B33F C CNTUN05: CALL LISTUN ; LISTEN TO TUNE
B338 A7 C AND A ; TEST RETURN CODE
B339 28 AB C JR NZ,CNTUN01 ; SELECT NEXT TUNE IF ZERO
B33B E1 C CNTUN06: POP HL
B33C D1 C POP DE
B33D C1 C POP BC
B33E C9 C RET
B33F CD B59D C LISTUN: CALL SHOTUN ; DISPLAY TUNE
B342 A7 C AND A ; TEST FOR SUCCESSFUL
B343 28 5D C JR NZ,LSTU06 ; ON ERROR SELECT ANOTHER TUNE
B345 CD B3A4 C CALL PRELUD ; ELSE INTRODUCE TUNE TO USER
B348 A7 C AND A ; TEST FOR CONTINUE FLAG
B349 28 0A C JR Z,LSTU01 ; JUMP IF READY TO CONTINUE
B34B FE 4D C CP 'M' ; DOES HE WANT TO EXIT?
B34D 28 4F C JR Z,LSTU05 ;JUMP IF SO
B34F FE 53 C CP ‘S’ ;SKIP THIS TUNE?
B351 28 4F C JR Z,LSTU06 ;JUMP IF SO
B353 18 EA C JR LISTUN ;ELSE RETURN TO SAME TUNE
B355 3E 10 C LSTU01: LD A,TYPTUN ;CREATE TUNE RECORD
B357 CD B837 C CALL CREREC
B35A CD B503 C LSTU02: CALL TRANSIT ;TRANSLATE TUNE TO INTEGER
B35D CD B306 C CALL SINGIT ;LISTEN TO HIM SING IT
B35F FE 53 C CP ‘S’ ;SKIP THIS TUNE?
B362 28 3E C JR Z,LSTU06 ;JUMP IF SO
B364 A7 C AND A ;DID HE FINISH IT?
B365 20 37 C JR NZ,LSTU05 ;EXIT IF NOT
B367 CD B522 C CALL EVALUT ;ELSE EVALUATE PERFORMANCE
B36A 3E 20 C LD A,TYPPER ;CREATE PERFORMANCE RECORD
B36C CD B837 C CALL CREREC
B36F CD B695 C CALL ADJRAT ;ADJUST USER RATING
B372 11 AB28 C LSTU03: LD DE,MSGCAP ;CONTINUE, AGAIN, PLAY, MENU
B375 CD B023 C CALL QRYOPT
B378 FE 43 C CP ‘C’ ;CONTINUE?
B37A 20 26 C JR Z,LSTU06 ;JUMP IF SO
B37C FE 50 C CP ‘P’ ;PLAY TUNE?
B37E 28 0C C JR Z,LSTU04 ;JUMP IF SO
B380 FE 41 C CP ‘A’ ;AGAIN?
B382 20 1A C JR NZ,LSTU05 ;EXIT IF NOT
B384 CD B590 C CALL SHOTUN ;RE-DISPLAY TUNE
B387 CD BA2B C CALL PRTHDR ;RE-DISPLAY HEADER
B38A 18 CE C JR LSTU02 ;AND LISTEN FOR IT
B38C CD B590 C LSTU04: CALL SHOTUN ;RE-DISPLAY THE TUNE
B38F CD BA2B C CALL PRTHDR ;RE-DISPLAY HEADER
B392 06 F0 C LD B,0F0H ;WHOLE=60 TEMPO
B394 3E 22 C LD A,MONOPH ;PLAY TUNE FUNCTION CODE
B396 CD C000 C CALL MCP
B399 CD BEA3 C CALL WAIT ;WAIT TILL USER IS READY
B39C 18 04 C JR LSTU03 ;ASK OPTIONS AGAIN
B39E 3E FF C LSTU05: LD A,0FFH ;SET EXIT CODE
B3A0 18 01 C JR LSTU07 ;AND EXIT
B3A2 AF C LSTU06: XOR A ;LOAD CONTINUE CODE
B3A3 C9 C LSTU07: RET
C ;
C ; PRELUD
C ;
C ; QUERY USER CONCERNING THE KEY OF THE TUNE AND HAVE
C ; HIM SING THE TONIC TRIAD BEFORE LETTING HIM SING
C ; THE TUNE.
C ;
C ; GIVEN:  NONE
C ; RETURNS: (A) - RETURN CODE
SIGHTSINGER'S MIRROR

PROGRAM SECTION

C ; = 00H - CONTINUE
C ; = 'S' - SKIP THIS TUNE
C ; = 'R' - RETURN TO WHERE LEFT OFF
C ; = 'M' - MAIN MENU

B3A4 C5 C PRELUD: PUSH BC
B3A5 D5 C PUSH DE
B3A6 E5 C PUSH HL
B3A7 CD B8F7 C CALL QRYKEY ; ASK FOR KEY OF TUNE
B3A8 A7 C AND A ; TEST FOR CORRECT ANSWER
B3A9 28 08 C JR Z,PRLD01 ; JUMP IF SO
B3AD 11 A9C0 C LD DE,MSGSRN ; SKIP, RETURN OR MENU?
B3B0 CD BD23 C CALL QRYOPT
B3B3 18 1D C JR PRLD02 ; EXIT WITH CODE S, R OR M
B3B5 CD BA28 C PRLD01: CALL PRLDR ; PRINT THE HEADER LINE
B3B8 3E 23 C LD A,PLAKEY ; AURALLY ESTABLISH KEY
B3BB CD C000 C CALL MCP
B3BD CD BA55 C CALL QRYTON ; GET HIM TO SING TONIC TRIAD
B3C0 A7 C AND A ; TEST FOR CONTINUE CODE
B3C1 28 0F C JR Z,PRLD02 ; JUMP IF ZERO
B3C3 11 A9C0 C LD DE,MSGSRH ; SKIP, RETURN OR MENU?
B3C6 CD BD23 C CALL QRYOPT
B3C9 FE 52 C CP 'R' ; TEST FOR START OVER
B3CB 28 05 C JR NZ,PRLD02 ; JUMP IF NOT STARTING OVER
B3CD CD B59D C CALL SHOTUN ; ELSE RE-DISPLAY TUNE
B3E0 18 E3 C JR PRLD01 ; AND HEADER
B3E2 E1 C PRLD02: POP HL
B3E3 D1 C POP DE
B3E4 C1 C POP BC
B3E5 C9 C RET

C ;

C ; SINGIT

C ; LISTEN TO USER'S PERFORMANCE OF TUNE AND CHANGE PITCHES
C ; TO BLACK WHOLE NOTES WHEN HE SINGS THEM RIGHT. ALSO
C ; DISPLAYS THE PERFORMANCE ON A SEPARATE STAFF BELOW THE
C ; ONE HE IS READING. ON THIS STAFF THE ERRORS ARE LEFT
C ; SO THAT THEY MAY BE REVIEWED UPON COMPLETION OF THE TUNE.
C ; THE TUNE IS ASSUMED TO BE ALREADY STORED IN INTEGER FORMAT
C ; STARTING AT 'WRKSPC'.

C ; GIVEN: (IX) - ADDRESS OF TUNE HEADER
C ; RETURNS: (A) - COMPLETION CODE
C ; = 'S' - SKIP THIS TUNE
C ; = 'M' - MAIN MENU
C ; = 00H - END OF TUNE REACHED
SIGHTSINGER'S MIRROR MACRO-80 3.44 09-Dec-81 PAGE 1-49

PROGRAM SECTION

B306 C5 C SINGIT: PUSH BC
B307 D5 C PUSH DE
B308 E5 C PUSH HL
B309 AF C XOR A ;ZERO OUT ATTEMPT NUMBER
B30A 32 0000X C LD (ATEMPT),A ;AND NUMBER OF PLAYINGS
B30D 32 0000X C LD (NUMPLA),A
B30E CD B447 C SGIT01: CALL PRETUN ;PREPARE TUNE FOR SINGING
B30F A7 C AND A ;TEST RETURN CODE
B314 20 5D C JR NZ,SGIT00 ;EXIT ON NON-ZERO
B316 CD BC83 C SGIT02: CALL LISTEN ;GET SING PITCH
B319 FE FD C CP KBINTR ;TEST FOR INTERRUPT
B31B 20 08 C JR NZ,SGIT03 ;JUMP IF NONE
B31D CD B4A3 C CALL MIDTUN ;MID-TUNE INTERRUPT HANDLER
B31F A7 C AND A ;ZERO MEANS START TUNE OVER
B320 28 ED C JR Z,SGIT01 ;JUMP TO START OVER
B323 18 4E C JR SGIT08 ;ELSE EXIT
B325 4F C SGIT03: LD C,A ;SAVE PITCH
B326 BE C CP (HL) ;COMPARE SING PITCH WITH TUNE
B328 20 0A C JR Z,SGIT04 ;JUMP IF SAME
B32A CD B4F1 C CALL PARNOT ;ELSE DISPLAY PARALLEL NOTE
B32C 3E FF C LD A,OFFH ;SET ERROR FLAG
B32D 32 0000X C LD (TEMP01),A
B32E 18 E3 C JR SGIT02 ;KEEP LISTENING
B330 CD B402 C SGIT04: CALL FILNOT ;BLACK OUT NOTE IN TOP STAFF
B334 3A 0000X C LD A,(NUMSNG) ;INCREMENT NUMBER NOTES SING
B338 3C C INC A
B339 32 0000X C LD (NUMSNG),A
B33B 3A 0000X C LD A,(TEMP01) ;CHECK ERROR FLAG
B33D A7 C AND A
B33E 20 09 C JR Z,SGIT05 ;JUMP IF NO ERROR
B340 3A 0000X C LD A,(ERROR) ;ELSE GET ERROR COUNT
B342 3C C INC A ;AND INCREMENT IT
B344 32 0000X C LD (ERROR),A
B346 18 04 C JR SGIT06 ;LEAVE NOTE IN ERROR SHOWING
B349 79 C SGIT05: LD A,C ;GET PITCH
B34A CD B4F1 C CALL PARNOT ;DISPLAY NOTE IN BOTTOM STAFF
B34C AF C SGIT06: XOR A ;RESET ERROR FLAG
B34D 32 0000X C LD (TEMP01),A
B34F CD B63F C CALL NOTPTR ;POINT (DE) TO NEXT NOTE
B351 FE FF C CP DBLBAR ;CHECK FOR END OF TUNE
B353 20 17 C JR Z,SGIT07 ;EXIT ON DOUBLE BAR
B355 86 02 C LD B,02H ;PARTITION NUMBER
B357 3E 01 C LD A,ACTPAR ;ACTIVATE TOP PARTITION
B359 CD C000 C CALL MCP
B35B 3E 31 C LD A,GETCP ;SET CURRENT POSITION
B35D CD C000 C CALL MCP
B35F 78 C LD A,B
B362 32 0000X C LD (TEMPCP),A ;SAVE IT
PROGRAM SECTION

B43B 79 C LD A,C ;SAVE STAFF ORIENTATION
B43C 32 0000X C LD (TEMPSO),A
B43F 23 C INC HL ;POINT TO NEXT INTEGER PITCH
B440 18 A4 C JR SGIT02 ;JUMP IF NOT
B442 AF C SGIT07: XOR A ;SET RETURN CODE IF COMPLETE
B443 E1 C SGIT08: POP HL
B444 D1 C POP DE
B445 C1 C POP BC
B446 C9 C RET

C ;
C ; INITIALIZES FIELDS AND WAITS TILL THE USER IS READY TO
C ; BEGIN SINGING. TONIC NOTE MAY BE PLAYED ANY NUMBER OF
C ; TIMES BY PRESSING THE 'T' KEY. THE ESCAPE KEY CAUSES
C ; AN OPTION MENU TO BE DISPLAYED. ANY OTHER KEY INDICATES
C ; THAT HE IS READY TO SING.
C ;
C ; GIVEN : (IX) - ADDRESS OF TUNE HEADER
C ; RETURNS : (A) - COMPLETION CODE
C ;
C ; = 08H - LISTEN TO TUNE
C ; = 'S' - SKIP THIS TUNE
C ; = 'M' - DISPLAY MAIN MENU
C ;
C ; (DE) - ADDRESS OF TUNE IN TUNE TABLE FORMAT
C ;
C ; (HL) - ADDRESS OF TUNE IN INTEGER FORMAT
C ;
B447 C5 C PRETUN: PUSH BC
B448 06 03 C PRET01: LD B,03H ;PARTITION NUMBER
B44A 3E 01 C LD A,ACTPAR ;ACTIVATE IT
B44C CD C000 C CALL MCP
B44F 01 0002 C LD BC,0002H ;X/Y COORDINATE
B452 11 AP36 C LD DE,MSGLOT ;'LOOK OVER THE TUNE...'
B455 3E 34 C LD A,PRTEXT
B457 CD C000 C CALL MCP
B459 DD E5 C PUSH IX ;PUT ADDRESS OF TUNE
B45C E1 C POP HL ; IN (HL)
B45D 11 0012 C LD DE,HDRLEN ;LOAD LENGTH OF TUNE HEADER
B460 19 C ADD HL,DE ;POINT TO FIRST NOTE
B461 EB C EX DE,HL ;PUT ADDRESS IN (DE)
B462 3E FF C LD A,NOISE ;INITIALIZE PREVIOUS PITCH
B464 32 0000X C LD (PREVIP),A ; TO NOISE
B467 3E 2C C LD A,INITPT ;INITIALIZE MCP INPUT PITCH
B469 CD C000 C CALL MCP
B46C CD BEED C PRET02: CALL KBFOLD ;WAIT FOR KEYBOARD RESPONSE
B46F FE 54 C CP 'T' ;TEST FOR REPLAY OF TONIC NOTE
B471 28 05 C JR NZ,PRET03 ;JUMP IF NOT
B473 CD BCA8 C CALL TONOTE ;ELSE PLAY TONIC NOTE AGAIN
MIDT01: CP 'P' ; TEST FOR PLAY TUNE
B4C0  28 0F  C  JR  NZ,MIDT03 ; EXIT IF NOT
B4C2  FE 50  C  CP 'M' ; TEST FOR MAIN MENU
B4C8  20 0F  C  JR  NZ,MIDT03 ; EXIT IF NOT
B4C4  18 12  C  JR  MIDT02 ; AND START OVER
B4C0  28 0F  C  JR  NZ,MIDT03 ; EXIT IF NOT

Given:
None
Returns: (A) - Completion Code
00H - Start Tune Over Again
'S' - Skip This Tune
'M' - Display Main Menu

MIDTUN: LD A,TYPPER ; Create Performance Record
CALL CRERECl
XOR A ; Reset Number of Playings
LD (NUMPLA),A
LD DE,MSGSRM ; Skip, Return, Play or Menu
CALL QRYOPT
FE 50 C CP 'P' ; Test for Play Tune
JR NZ,MIDT01 ; Jump if Not
SHOTUN ; Re-Display Tune
PRTHDR ; Re-Display Header
B4C8 18 12 C JR MIDT02 ; AND START OVER
B4C0 28 0F C JR NZ,MIDT03 ; EXIT IF NOT

MID-TUNE INTERRUPT HANDLER. An Option Menu is Displayed
Allowing 4 Options: Skip the Tune, Return to Tune, Play the Tune, and Display Main Menu. A Performance Record
Is Always Written to the Buffer.

Given: None
Returns: (A) - Completion Code
00H - Start Tune Over Again
'S' - Skip This Tune
'M' - Display Main Menu

B4A3  3E 28 C MIDTUN: LD A,TYPPER ; Create Performance Record
B4A5  CD B837 C CALL CRERECl
B4A8  AF C XOR A ; Reset Number of Playings
B4A9  32 0000X C LD (NUMPLA),A
B4AC  11 AA67 C LD DE,MSGSRM ; Skip, Return, Play or Menu
B4AF  CD BD23 C CALL QRYOPT
B4B2  FE 52 C CP 'R' ; Test for Return
B4B4  28 08 C JR NZ,MIDT01 ; Jump if Not
B4B6  CD B59D C CALL SHOTUN ; Re-Display Tune
B4B9  CD BA2B C CALL PRTHDR ; Re-Display Header
B4BC  18 12 C JR MIDT02 ; AND START OVER
B4BD  FE 50 C MIDT01: CP 'P' ; Test for Play Tune
B4C0  28 0F C JR NZ,MIDT03 ; EXIT IF NOT
SIGHTSINGER'S MIRROR MACRO-80 3.44 09-Dec-81 PAGE 1-52

PROGRAM SECTION

B4C2  CD B59D  C  CALL  SHOTUN  ;RE-DISPLAY TUNE
B4C5  86 F0  C  LD  B,0F0H  ;TEMPO
B4C7  3E 22  C  LD  A,MONOPH  ;PLAY TUNE (4 VOICES IN UNISON)
B4C9  CD C000  C  CALL  MCP
B4CC  21 0000x  C  LD  HL,NUMPLA  ;INCREMENT PLAYING COUNT
B4CF  34  C  INC  (HL)
B4D0  AF  C  MIDT02; XOR  A  ;SET START OVER FLAG
B4D1  C9  C  MIDT03; RET

C

C

C

C ; FILNOT
C

C ; CHANGE DISPLAYED NOTE TO BLACK WHOLE NOTE AT THE CURRENT
C ; POSITION IN PARTITION 2.
C

C ; GIVEN: (DE) - ADDRESS OF NOTE DATA
C ; (IX) - ADDRESS OF TUNE HEADER
C ; RETURNS: (A) - COMPLETION CODE
C ;     = 00H - SUCCESSFUL
C ;     = EOS - END OF STAFF; NOTE NOT DISPLAYED
C ;     = OORANG - OUT OF RANGE
C ;     = FFH - INVALID DATA
C

B4D2  C5  C  FILNOT; PUSH  BC
B4D3  D5  C  PUSH  DE
B4D4  E5  C  PUSH  HL
B4D5  06 02  C  LD  B,02H  ;PARTITION NUMBER
B4D7  3E 01  C  LD  A,ACTPAR  ;ACTIVATE IT
B4D9  CD C000  C  CALL  MCP
B4DC  62  C  LD  H,D  ;SAVE ADDRESS OF NOTE
B4DD  6B  C  LD  L,E
B4DE  23  C  INC  HL  ;POINT TO RHYTHM BYTE
B4DF  23  C  INC  HL
B4E0  23  C  INC  HL
B4E1  7E  C  LD  A,(HL)  ;GET RHYTHM
B4E2  47  C  LD  B,A  ;SAVE IT
B4E3  36 00  C  LD  (HL),BLKWHL  ;MAKE IT A BLACK WHOLE NOTE
B4E5  EB  C  EX  DE,HL  ;RESET (HL) TO ADDRESS OF NOTE
B4E6  CD B628  C  CALL  DISPNO  ;DISPLAY THIS NOTE
B4E9  4F  C  LD  C,A  ;SAVE RETURN CODE
B4EA  78  C  LD  A,B  ;RESET OLD RHYTHM
B4EB  12  C  LD  (DE),A
B4EC  79  C  LD  A,C  ;RESTORE RETURN CODE
B4ED  E1  C  POP  HL
B4EE  D1  C  POP  DE
B4EF  C1  C  POP  BC
B4F0  C9  C  RET
C  ; PARNOT
C  ;
C  ; DISPLAYS NOTE IN THE PARALLEL PARTITION AT THE CURRENT
C  ; POSITION OF THE TOP PARTITION TO KEEP THE NOTES LINED
C  ; UP. IF ANY PITCH IS ALREADY STORED IN THIS POSITION
C  ; IT IS ERASED. THE LAST PITCH SUNG IN ERROR IS LEFT
C  ; DISPLAYED IN THIS PARALLEL PARTITION.
C  ;
C  ; GIVEN:  (A) - INTEGER PITCH
C  ;  (DE) - ADDRESS OF NOTE DATA
C  ; RETURNS: (A) - COMPLETION CODE
C  ;         = 00H - SUCCESSFUL
C  ;         = EOS - END OF STAFF; NOTE NOT DISPLAYED
C  ;         = ORANG - OUT OF RANGE
C  ;         = FFH - INVALID DATA
C  ;
B4F1  C5  C  PARNOT:  PUSH  BC
B4F2  D5  C  PUSH  DE
B4F3  E5  C  PUSH  HL
B4F4  EB  C  EX  DE,HL           ;PUT NOTE ADDRESS IN (HL)
B4F5  57  C  LD  D,A             ;SAVE INTEGER PITCH
B4F6  06 03  C  LD  B,03H         ;ACTIVATE BOTTOM PARTITION
B4F8  3E 01  C  LD  A,ACTPAR
B4FA  CD  C000  C  CALL  MCP
B4FD  3A 0000X  C  LD  A,(TEMPCP)    ;GET POSITION OF TOP PARTITION
B500  47  C  LD  B,A             ;SET CURRENT POSITION OF BOTTOM
B501  3A 0000X  C  LD  A,(TEMPSO)    ;PARTITION TO SAME
B504  4F  C  LD  C,A
B505  3E 30  C  LD  A,SETCP
B507  CD  C000  C  CALL  MCP
B50A  0E 03  C  LD  C,03H         ;ERASE THE 3 COLUMNS STARTING
B50C  3E 19  C  LD  A,ERACOL      ;AT THE POSITION GIVEN IN
B50E  CD  C000  C  CALL  MCP      ;(B)
B511  42  C  LD  B,D             ;DISPLAY GIVEN INTEGER PITCH
B512  0E 01  C  LD  C,WHNOT       ;AS WHOLE NOTE
B514  23  C  INC  HL             ;POINT TO CPI OF ORIGINAL NOTE
B515  23  C  INC  HL
B516  23  C  INC  HL
B517  23  C  INC  HL
B518  56  C  LD  D,(HL)          ;GET CPI
B519  3E 17  C  LD  A,PITCHI      ;DISPLAY PITCH
B51B  CD  C000  C  CALL  MCP
B51E  E1  C  POP  HL
B51F  D1  C  POP  DE
B520  C1  C  POP  BC
B521  C9  C  RET
PROGRAM SECTION

C ; EVALUT
C ; EVALUATE PERFORMANCE OF TUNE.
C ;
C ; GIVEN: NONE
C ;
C ;

B522 C5 C EVALUT: PUSH BC
B523 D5 C PUSH DE
B524 E5 C PUSH HL
B525 21 ABF0 C LD HL,MSG00X "YOU MISSED NNN OUT OF..."
B526 CD BEF9 C CALL MOVMSG ;MOVE TO RAM
B527 CD BD66 C CALL STATMSG ;SET UP STAT MESSAGE FIELDS
B528 06 04 C LD B,04H ;PARTITION NUMBER
B530 CD BD22 C CALL ANCPAR ;ACTIVATE AND CLEAR IT
B532 01 0101 C LD BC,0101H ;X/Y COORDINATE
B534 11 0006X C LD DE,THPMSG ;"YOU MISSED NNN OUT OF..."
B536 3E 34 C LD A,PRTEXT ;DISPLAY MESSAGE
B538 CD C000 C CALL MCP
B53E CD BEA3 C CALL WATT
B541 06 04 C LD B,04H ;CLEAR OUT PARTITION
B543 3E 02 C LD A,CCLRPAR
B545 CD C000 C CALL MCP
B548 E1 C POP HL
B549 D1 C POP DE
B54A C1 C POP BC
B54B C9 C RET

C ; MACLIB B:SM04.MAC
C ;
C ; SETPAR
C ; SET UP PARALLEL PARTITION. DISPLAYS STAFF AND CLEF IN THE
C ; PARTITION BELOW THE TUNE DISPLAYED WITH THE SAME KEY
C ; SIGNATURE. THE PARTITION IN WHICH THE TUNE IS DISPLAYED
C ; IS ACTIVATED BEFORE EXITING.
C ;
C ; GIVEN: (IX) - ADDRESS OF TUNE HEADER
C ;
C ;

B54C C5 C SETPAR: PUSH BC
B54D D5 C PUSH DE
B54E E5 C PUSH HL
B550 06 03 C LD B,03H ;BOTTOM PARTITION NUMBER
B551 3E 01 C LD A,ACTPAR ;ACTIVATE THE PARTITION
SHOTUN

SHOW TUNE TO USER AND TRANSLATE IT TO A SERIES OF INTEGER PITCHES. IF THE TUNE CANNOT BE DISPLAYED FOR ANY REASON THE ERROR BIT IN THE TUNE HEADER IS SET AND THE APPROPRIATE ERROR CODE IS RETURNED.

GIVEN: (IX) - ADDRESS OF TUNE HEADER

RETURNS: (A) - COMPLETION CODE

= 00H - SUCCESSFUL
C ; = EOP - END OF PARTITION FOUND BEFORE END
C ; OF TUNE
C ; = FDH - INVALID TUNE TYPE
C ; = FFH - INVALID KEY IN HEADER
C ; = FFH - INVALID NOTE IN TUNE

B59D C5 C SHOTUN: PUSH BC
B59E D5 C PUSH DE
B59F E5 C PUSH HL
B5A0 CD B503 C CALL TRANST ;TRANSLATE TUNE TO INTEGER
B5A3 A7 C AND A ;TEST RETURN CODE
B5A4 20 1F C JR NZ,SHOW01 ;EXIT IF ERROR FOUND
B5A6 DD E5 C PUSH IX ;PUT TUNE ADDRESS IN (DE)
B5A8 D1 C POP DE
B5A9 13 C INC DE ;SKIP OVER LENGTH OF TUNE
B5AA 13 C INC DE
B5AB 3E 33 C LD A,ENASTO ;ENABLE STORE IN TUNE TABLE
B5AD CD C000 C CALL MCP
B5B0 01 A02B C LD BC,PRLST ;POINT TO PARTITION LIST
B5B3 3E 8F C LD A,NEXTUN ; TO USE TO DISPLAY THE
B5B5 CD C000 C CALL MCP ; TUNE
B5B8 06 00 C LD B,00H ;SET RETURN CODE TO SUCCESSFUL
B5BA FE 3A C CP EOTUNE ;TEST FOR SUCCESSFUL RETURN
B5BC 20 07 C JR NZ,SHOW01 ;JUMP IF NOT SUCCESSFUL
B5BE 3E 32 C LD A,INHIBT ;PROTECT THE TUNE TABLE
B5C5 CD C000 C CALL MCP
B5C8 18 09 C JR SHOW02 ;EXIT SUCCESSFUL
B5CD 47 C SHOW01: LD B,A ;SAVE RETURN CODE
B5CE DD 7E 09 C LD A,(IX+BITFLG) ;GET BIT FLAGS
B5CF CB F7 C SET ERRBIT,A ;SET ERROR BIT
B5C9 DD 77 09 C LD (IX+BITFLG),A ;PUT IT BACK
B5CE 76 C SHOW02: LD A,B ;RESTORE RETURN CODE
B5D0 E1 C POP HL
B5D1 D1 C POP DE
B5D2 C1 C POP BC
B5D3 C9 C RET
C ; ; TRANST
C ; ; TRANSLATE TUNE TO SERIES OF INTEGER PITCHES.
C ; ; GIVEN: (IX) - ADDRESS OF TUNE HEADER
C ; RETURNS: (A) - COMPLETION CODE
C ; = 00H - SUCCESSFUL
C ; = FFH - ERROR IN DATA

B5D3 C5 C TRANST: PUSH BC
SIGHTSINGER'S MIRROR  MACRO-80  3.44  09-Dec-81  PAGE 1-57

PROGRAM SECTION

B5D4  D5  C  PUSH DE
B5D5  E5  C  PUSH HL
B5D6  DD E5  C  PUSH IX
B5D8  21 0008  C  LD HL,WRKSPC ;DESTINATION ADDRESS
B5DB  D5  C  LD A,(IX+TUNITYP) ;GET TUNE TYPE
B5DE  32 0008  C  LD (TEMP02),A ;SAVE IT
B5D1  10 0012  C  LD DE,HLRDLEN ;LOAD LENGTH OF TUNE HEADER
B5D4  DD 19  C  TRNT01: ADD IX,DE ;POINT TO NEXT NOTE
B5D6  DD 7E 00  C  LD A,(IX+SCLLET) ;GET SCALE DEGREE/LETTER NAME
B5D8  FE FF  C  CP DBLBAR ;TEST FOR END OF TUNE
B5D9  20 33  C  JR Z,TRNT04 ;JUMP IF END OF TUNE
B5DA  FE FE  C  CP SNGBAR ;TEST FOR SINGLE BAR
B5DB  20 2A  C  JR Z,TRNT03 ;JUMP IF SINGLE BAR
B5D1  47  C  LD B,A ;ELSE IT IS A PITCH
B5D2  DD 4E 01  C  LD C,(IX+MODACD) ;GET MODIFIER/ACCIDENTAL
B5D5  3A 0008  C  LD A,(TEMP02) ;GET TUNE TYPE
B5D8  FE A0  C  CP TYPSCL ;TEST FOR SCALE DEGREE TYPE
B5D9  20 07  C  JR NZ,TRNT02 ;JUMP IF LETTER NAME TYPE
B5DA  3E 1C  C  LD A,CVTSTL ;CONVERT SCALE DEGREE TO LETTER
B5DB  CD C000  C  CALL MCP
B601  42  C  LD B,D ;MOVE RESULT
B602  4B  C  LD C,E
B603  DD 56 02  C  TRNT02: LD D,(IX+OCTAVE) ;GET OCTAVE OF PITCH
B606  3E 1E  C  LD A,CVTLTA ;CONVERT LETTER NAME TO
B608  CD C000  C  CALL MCP ;TO ABSOLUTE PITCH
B60B  FE FF  C  CP 0FFH ;TEST FOR ERROR
B60D  28 12  C  JR Z,TRNT05 ;JUMP ON ERROR
B60F  47  C  LD B,A ;ELSE CONVERT ABSOLUTE PITCH
B610  3E 20  C  LD A,CVTATI ;TO INTEGER PITCH
B612  CD C000  C  CALL MCP
B615  FE FF  C  CP 0FFH ;TEST FOR ERROR
B617  28 08  C  JR Z,TRNT05 ;JUMP ON ERROR
B619  77  C  LD (HL),A ;STORE INTEGER PITCH
B61A  23  C  INC HL ;POINT TO NEXT POSITION
B61B  11 0005  C  TRNT03: LD DE,NOTLEN ;LOAD LENGTH OF NOTE GROUP
B61E  18 04  C  JR TRNT01 ;GO GET NEXT NOTE
B620  AF  C  TRNT04: XOR A ;ZERO OUT ACCUMULATOR
B621  77  C  TRNT05: LD (HL),A ;AS END OF TUNE FLAG
B622  DD E1  C  POP IX
B624  E1  C  POP HL
B625  D1  C  POP DE
B626  C1  C  POP BC
B627  C9  C  RET
C ;
C ;
C ; DISPNO
C ;
C ; DISPLAY NOTE FROM EITHER SCALE DEGREE OR LETTER NAME
C ; DATA AT THE CURRENT POSITION IN THE ACTIVE PARTITION.
C ;
C ; GIVEN: (HL) - ADDRESS OF NOTE DATA STREAM
C ; (IX) - ADDRESS OF TUNE HEADER
C ;
C ; RETURNS: (A) - COMPLETION CODE
C ; = 00H - SUCCESSFUL
C ; = EOS - END OF STAFF; NOTE NOT DISPLAYED
C ; = ODRANG - OUT OF RANGE
C ; = FFH - INVALID DATA
C ;
B628 C5 C DISPNO: PUSH BC
B629 D5 C PUSH DE
B62A E5 C PUSH HL
B62B 06 14 C LD BC,PTCHST ;FIRST ASSUME SCALE DEGREE
B62D DD 7E 03 C LD A,(IX+TUNTYP) ;GET TUNE TYPE
B630 FE A0 C CP TYPSCL ;IS IT SCALE DEGREE TYPE?
B632 28 02 C JR Z,DISP01 ;JUMP IF IT IS
B634 06 15 C LD BC,PTCHLT ;ELSE LOAD FUNCTION CODE
B636 78 C DISP01: LD A,IX ;GET MCP FUNCTION CODE
B637 EB C EX DE,HL ;PUT ADDRESS OF NOTE IN (DE)
B638 CD 0000 C CALL MCP ;DISPLAY THE NOTE
B63B E1 C POP HL
B63C D1 C POP DE
B63D C1 C POP BC
B63E C9 C RET
C ;
C ; NXTPTR
C ;
C ; GET POINTER TO NEXT NOTE IN TUNE SKIPPING OVER BAR
C ; LINES AND ADJUST ACTIVE PARTITION'S CURRENT POSITION
C ; WHEN A BAR LINE IS FOUND.
C ;
C ; GIVEN: (DE) - ADDRESS OF CURRENT NOTE
C ;
C ; RETURNS: (DE) - ADDRESS OF NEXT NOTE
C ;
C ; (A) - COMPLETION CODE
C ; = 00H - NORMAL
C ; = DBLBAR - END OF TUNE REACHED
C ;
B63F C5 C NXTPTR: PUSH BC
B640 E5 C PUSH HL
B641 CD B652 C CALL NXTNOT ;GET ADDRESS OF NEXT NOTE
B644 FE FF C CP DBLBAR ;END OF TUNE FOUND?
B646 28 97 C JR Z,NXPT01 ;JUMP ON END OF TUNE
B648 A7 C AND A ;TEST FOR SINGLE BAR LINE
B649 28 94 C JR Z,NXPT01 ;JUMP IF NOT FOUND
B64A CD B66D C CALL BARADJ ;LINE UP BAR LINES
B64F AF C XOR A ;AND SET NORMAL RETURN CODE
GET ADDRESS OF NEXT NOTE IN TUNE.

GIVEN: (DE) - ADDRESS OF CURRENT NOTE

RETURNS: (DE) - ADDRESS OF NEXT NOTE IN TUNE

(A) - BAR LINE INDICATOR

- 00H - NO INTERVENING BAR LINE
- DBLBAR - END OF TUNE
- SNGBAR - BAR LINE CROSSED

BARADJ

ADJUSTS CURRENT POSITION IN THE MODEL AND PARALLEL PARTITIONS SO THAT THE BAR LINES WILL BE LINED UP.

GIVEN: NONE

RETURNS: NONE

BARADJ: PUSH BC

ACTIVATE TOP PARTITION
AOJRAT  
ADJUST USER RATING ACCORDING TO USERS PERFORMANCE ON TUNE. THE RATING ADJUSTMENT IS A FUNCTION OF HIS SCORE:

<table>
<thead>
<tr>
<th>SCORE</th>
<th>RATING ADJUSTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>98-100</td>
<td>+5</td>
</tr>
<tr>
<td>80-89</td>
<td>+2</td>
</tr>
<tr>
<td>78-79</td>
<td>0</td>
</tr>
<tr>
<td>68-69</td>
<td>-5</td>
</tr>
<tr>
<td>60-69</td>
<td>-10</td>
</tr>
</tbody>
</table>

GIVEN: None  
RETURNS: None

ADJRAT: PUSH BC  
LD A,(SCORE) ;GET SCORE  
LD B,05H ;LOAD ADJUSTMENT  
CP 5AH ;TEST FOR 90-100  
JR NC,ADRT01 ;JUMP IF IN RANGE  
LD B,02H ;ELSE LOAD NEXT LEVEL  
CP 50H ;TEST FOR 80-89  
JR NC,ADRT01 ;JUMP IF IN RANGE  
LD B,0FH ;ELSE LOAD NEXT LEVEL  
CP 46H ;TEST FOR 78-79  
JR NC,ADRT01 ;JUMP IF IN RANGE  
LD B,0FH ;LOAD NEXT LEVEL  
CP 3CH ;TEST FOR 68-69  
JR NC,ADRT01 ;JUMP IF IN RANGE  
LD B,0FH ;LOAD BOTTOM LEVEL
ADRT81: LD A,(USRRAT) ;GET CURRENT USER RATING
ADD A,B ;ADD ADJUSTMENT
LD (USRRAT),A ;STORE IT
POP BC
RET

; FIXRAT

; ADJUST USER RATING ACCORDING TO ITS CURRENT VALUE
; RELATIVE TO THE STARTING VALUE. IF THE CURRENT VALUE
; IS LARGER THAN THE STARTING VALUE PLUS 10, THE RATING
; ISRESET TO THE STARTING RATING. IF IT IS SMALLER
; THAN THE SAME VALUE, THE RATING IS SET TO ITS CURRENT
; RATING MINUS 10. THIS IS DONE WHEN NO TUNES CAN BE
; FOUND IN THE TUNE GROUP WHICH MEET THE CURRENT
; SELECTION CRITERIA.

; GIVEN: (IY) - ADDRESS OF TUNE GROUP HEADER
; RETURNS: (A) - ADJUSTED USER RATING

FIXRAT: PUSH BC
LD A,(IY+RATONE> GET STARTING RATING
LD C,A SAVE IT
ADD A,BA H ADD 10 TO IT
LD B,A STORE RESULT
LD A,(USRRAT) COMPARE CURRENT USER RATING
CP B WITH STARTING RATING + 10
JR C,FXRT01 JI WP IF M S NOT IMPROVED
LD A,C ELSE SET RATING TO
JR FXRT82 STARTING RATING
FXRT01: SUB 0AH SUBTRACT 10 FROM CURRENT RATING
JR NC,FXRT82 JUMP IF RESULT IS POSITIVE
XOR A ELSE SET IT TO ZERO
FXRT02: LD (USRRAT),A STORE ADJUSTED RATING
POP BC
RET

INITPR
; Initializes Performance Record Fields.
; GIVEN: None
; RETURNS: None

; INITPR: XOR A ;Zero Out Fields
LD (NUMSNG),A ;Number Notes Sung
B6D5 32 0000h  C  LD (ERRCNT),A  ;NUMBER NOTES MISSED  
B6DE 32 0000h  C  LD (SCORE),A  ;SCORE  
B6E1 32 0000h  C  LD (TEMP01),A  ;ERROR FLAG  
B6E4 C9  C  RET  

MACLIB B:SSM05.MAC  

; SELECT TUNE  
; SEARCHES THE CURRENT TUNE GROUP FOR THE NEXT TUNE WHICH  
; MEETS THE SELECTION CRITERIA. THE SELECTION SCHEME MAY  
; BE RANDOM, LINEAR OR BASED ON A DIFFICULTY RATING.  
;  
; GIVEN: (IY) - ADDRESS OF TUNE GROUP HEADER  
; RETURNS: (A) - COMPLETION CODE  
; - 00H - SUCCESSFUL  
; - 08H - NO TUNE FOUND WHICH MEETS CRITERIA  
; - 10H - NO REMAINING UNUSED TUNES  
; (IX) - ADDRESS OF TUNE HEADER  

B6E5 C5  C  SELTUN: PUSH BC  
B6E6 D5  C  PUSH DE  
B6E7 E5  C  PUSH HL  
B6E8 FD 7E 06  C  SLTN01: LD A,(IY+SELSCH)  ;GET SELECTION SCHEME  
B6EB FE 08  C  CP RATECH  ;TEST FOR RATING SCHEME  
B6E9 28 08  C  JR NZ,SLTN02  ;JUMP IF NOT  
B6EF CD B76B  C  CALL RATESEL  ;GET NEXT TUNE BY RATING  
B6F2 A7  C  AND A  ;TEST RETURN CODE  
B6F3 26 14  C  JR Z,SLTN04  ;JUMP IF GOOD  
B6F5 18 59  C  JR SLTN06  ;ELSE EXIT  
B6F7 FE 10  C  SLTN02: CP RNDSCH  ;TEST FOR RANDOM SELECT SCHEME  
B6F9 28 08  C  JR NZ,SLTN03  ;JUMP IF NOT RANDOM  
B6FB CD B7A8  C  CALL RNDSEL  ;GET RANDOM TUNE  
B6FE A7  C  AND A  ;TEST RETURN CODE  
B6FF 28 08  C  JR Z,SLTN04  ;JUMP IF GOOD  
B701 18 40  C  JR SLTN06  ;ELSE EXIT  
B703 CD B754  C  SLTN03: CALL LINESEL  ;GET NEXT AVAILABLE TUNE  
B706 A7  C  AND A  ;TEST FOR GOOD RETURN CODE  
B707 26 47  C  JR NZ,SLTN06  ;EXIT IF NOT GOOD  
B709 DD 7E 09  C  SLTN04: LD A,(IX+BITLEG)  ;GET TUNE USED FLAG  
B70C CB FF  C  SET USEDBT,A  ;SET TUNE-USED BIT  
B70E DD 77 09  C  LD (IX+BITLEG),A  ;STORE IT BACK  
B711 3A 0000h  C  LD A,(TUNCNT)  ;INCREMENT NUMBER OF TUNES  
B714 3C  C  INC A  ; SELECTED  
B715 32 0000h  C  LD (TUNCNT),A
LINSEL

SELECTS A TUNE FROM THE TUNE GROUP USING A LINEAR SEARCH FOR THE FIRST UNUSED TUNE WITHOUT ITS ERROR BIT SET.

GIVEN:
(IY) - ADDRESS OF TUNE GROUP HEADER

RETURNS:
(A) - COMPLETION CODE

= 00H - SUCCESSFUL

= 10H - NO REMAINING UNUSED TUNES IN GROUP

(IX) - ADDRESS OF TUNE HEADER

B754 E5 C LINSSEL: PUSH HL
B755 FD E5 C PUSH IY ;PUT TUNE ADDRESS IN (HL)
B757 E1 C POP HL
B758 11 0020 C LD DE,TGLEN ;POINT TO FIRST TUNE
B75B 19 C ADD HL,DE
B75C CD B84D C CALL GETUNE ;GET NEXT TUNE
B75F 7C C LD A,H ;TEST FOR END OF GROUP
B76B B5 C OR L
B761 3E 10 C LD A,10H ;ASSUME END OF GROUP
B763 28 01 C JR Z,LINS01 ;JUMP IF END OF GROUP
B765 AF C XOR A ;ELSE SET SUCCESSFUL FLAG
B766 E5 C LINS01: PUSH HL ;PUT ADDRESS OF TUNE
B767 DD E1 C POP IX ; IN (IX)
B769 E1 C POP HL
B76A C9 C RET ;

RATSEL ;
;
;
;
;
;

B76B C5 C RATSEL: PUSH BC
B76C D5 C PUSH DE
B76D E5 C PUSH HL
B76E FD E5 C PUSH IY ;PUT GROUP HEADER ADDRESS
B770 E1 C POP HL ; IN (HL)
B771 11 0820 C LD DE,TGLEN ;LENGTH OF GROUP HEADER
B774 19 C ADD HL,DE ;POINT TO FIRST TUNE
B775 CD B84D C CALL GETUNE ;GET FIRST UNUSED TUNE
B778 7C C LD A,H ;TEST FOR END OF GROUP
B779 B5 C OR L
B77A 28 04 C JR NZ,RATS01 ;JUMP IF NOT END OF GROUP
B77C 3E 10 C LD A,10H ;ELSE LOAD END FLAG
B77E 10 27 C JR RATS03 ; AND EXIT
B780 E5 C RATS01: PUSH HL ;PUT HEADER ADDRESS IN (IX)
B781 DD E1 C POP IX
B783 3A 0000X C LD A,(USRAT) ;GET USER'S CURRENT RATING
B786 C6 06 C ADD A,6H ;GET CURRENT MAXIMUM RATING
B788 47 C LD B,A ;SAVE IT
B789 DD 7E 88 C LD A,(IX+RATING) ;GET TUNE RATING
B78C 88 C CP B ;COMPARE WITH USER RATING
B78D 30 88 C JR NC,RATS02 ;JUMP IF TUNE IS TOO HARD
B78F C6 8B C ADD A,6BH ;ELSE TEST FOR TOO EASY
RANDOMLY SELECTS A TUNE FROM THE TUNE GROUP. IF THE TUNE SELECTED BY A RANDOM NUMBER HAS ALREADY BEEN USED OR HAS AN ERROR, A LINEAR SCAN IS DONE FROM THAT POINT. IF THE END OF THE TUNE GROUP IS REACHED THE SEARCH IS STARTED AT THE BEGINNING.

GIVEN: (IY) - ADDRESS OF TUNE GROUP HEADER
RETURNS: (A) - COMPLETION CODE
= 08H - SUCCESSFUL
= 10H - NO REMAINING UNUSED TUNES
(IX) - ADDRESS OF TUNE HEADER

RNDSEL: PUSH BC
PUSH DE
PUSH HL
RNDSEL: LD B,(IY+NUMTUN) ;GET NUMBER OF TUNES IN GROUP
LD A,(TUNCNT) ;GET NUMBER TUNES SELECTED
LD B,A ;TEST FOR ALL TUNES DONE
JR NC,RNDS04 ;JUMP IF END OF TUNE GROUP
LD A,B ;ELSESEND NUMBER AS RANGE
CALL RNDS01 ;GET RANDOM NUMBER
LD B,A ;SEND IT IN (B)
CALL SCANTG ;GET ADDRESS OF THAT TUNE
AND A ;TEST FOR ERROR
JR 2,RNDS02 ;JUMP IF NO ERROR
DEC B ;DECREMENT INVALID NUMBER
LD (IY+NUMTUN),B ;CORRECT NUMBER TUNES VALUE
JR RNDS01 ;START OVER
SIGHTSINGER'S MIRROR  
MACRO-88 3.44  09-Dec-81  PAGE 1-66

PROGRAM SECTION

B7C8  DD 7E 09  C  RND02: LD  A, (IX+BITFLG) ;GET BIT FLAGS
B7CB  CB 7F  C  BIT  USBDT,A ;TEST IF ALREADY USED
B7CD  20 07  C  JR  NZ,RND03 ;BYPASS IF SO
B7CF  CB 77  C  BIT  ERRBIT,A ;TEST ERROR BIT
B7D0  20 03  C  JR  NZ,RND03 ;JUMP IF TUNE HAS ERROR
B7D3  AF  C  XOR  A ;ELSE SET SUCCESSFUL CODE
B7D4  20 20  C  JR  RND06 ;AND EXIT
B7D6  DD 05  C  RND03: PUSH IX ;PUT ADDRESS IN (HL)
B7D8  E1  C  CALL GETUNE ;DO LINEAR SEARCH FROM HERE
B7DC  7C  C  LD  A,H ;TEST FOR END OF GROUP
B7DD  65  C  OR  L
B7DE  20 12  C  JR  NZ,RND05 ;JUMP IF TUNE FOUND
B7E0  11 0200  C  LD  HL,TUNGRP ;CONTINUE LINEAR SEARCH
B7E3  11 0020  C  LD  DE, TGHLEN ;FROM BEGINNING OF
B7E6  19  C  ADD  HL, DE ;TUNE GROUP
B7E7  CD 080D  C  CALL GETUNE ;GET NEXT AVAILABLE TUNE
B7EA  7C  C  LD  A,H ;TEST FOR NONE AVAILABLE
B7EB  B5  C  OR  L
B7EC  20 04  C  JR  NZ,RND05 ;JUMP IF TUNE FOUND
B7EE  3E 10  C  RND04: LD  A, 18H ;ELSE SET END OF GROUP FLAG
B7F0  18 04  C  RND05: XOR  A ;SET SUCCESSFUL FLAG
B7F2  AF  C  RND05: POP  HL ;PUT ADDRESS OF TUNE
B7F3  E5  C  POP  IX ;IN (IX)
B7F4  DD 01  C  RND06: POP  DE
B7F7  D1  C  POP  BC
B7F8  C9  C  RET

B7FA  C5  C  SCANTG: PUSH BC
B7FB  D5  C  PUSH DE
B7FC  E5  C  PUSH HL
B7FD  DD 21 0000  C  LD  IX,TUNGRP ;POINT TO START OF TUNE GROUP
B801  11 0020  C  LD  DE, TGHLEN ;ADD LENGTH OF HEADER
B804  DD 19  C  SCAN01: ADD  IX, DE ;POINT TO TUNE
PROGRAM SECTION

BB06 18 03 C DJNZ SCAN02 ;COUNT NUMBER OF TUNES
BB08 AF C XOR A ;SET SUCCESSFUL CODE
BB09 18 0C C JR SCAN03 ;EXIT
BB0B DD 56 01 C SCAN02: LD D, (IX+01H) ;ELSE GET LENGTH OF TUNE
BB0E DD 5E 00 C LD E, (IX+00H)
BB11 7A C LD A, D ;TEST FOR END OF GROUP
BB12 B3 C OR E
BB13 20 EF C JR NZ, SCAN01 ;JUMP IF NOT END OF GROUP
BB15 3E FF C LD A, (IX+FFH) ;SET INVALID FLAG
BB17 E1 C SCAN03: POP HL
BB18 D1 C POP DE
BB19 C1 C POP BC
BB1A C9 C RET

; RNDNUM
; FINDS A RANDOM NUMBER IN THE RANGE FROM 1 TO THE
; VALUE GIVEN, INCLUSIVE.
; GIVEN: (A) - LARGEST VALUE TO RETURN
; RETURNS: (A) - RANDOM NUMBER BETWEEN 1 AND THE VALUE GIVEN

BB1B C5 C RNDNUM: PUSH BC
BB1C D5 C PUSH DE
BB1D 4F C LD C, A ;SAVE MAX VALUE
BB1E CD B02B C CALL RANDOM ;GET 16-BIT RANDOM NUMBER
BB21 06 00 C LD B, 00H ;PREPARE FOR DIVIDE
BB23 CD BFA8 C CALL DIVIDE ;MOD BY GIVEN VALUE
BB26 79 C LD A, C ;GET REMAINDER
BB27 3C C INC A ;MAKE RELATIVE TO 1
BB28 D1 C POP DE
BB29 C1 C POP BC
BB2A C9 C RET

; RANDOM
; GETS A 16-BIT RANDOM NUMBER.
; GIVEN: NONE
; RETURNS: (DE) - 16-BIT RANDOM NUMBER

BB2B C5 C RANDOM: PUSH BC
BB2C E5 C PUSH HL
BB2D ED 5F C LD A, R ;USE REFRESH VALUE IN (BC)
BB2F 4F C LD C, A
BB30 06 00 C LD B, 00H
B832 0F C RRC A ; PICK A SEED VALUE DEPENDING
B833 38 06 C JR C, RAND01 ; ON THE LOW ORDER BIT
B835 4F C LD C, A
B836 21 0000 X C LD HL, SEED1
B839 18 03 C JR RAND02
B83B 21 0000 X C RAND01: LD HL, SEED2
B83E 5E C RAND02: LD E, (HL) ; LOAD THE SEED VALUE
B83F 23 C INC HL
B840 56 C LD D, (HL)
B841 EB C EX DE, HL
B842 09 C ADD HL, BC ; ADD SEED TO REFRESH
B843 EB C EX DE, HL
B844 CD 8F 8E C CALL MLTPLY ; MULTIPLY SUM BY REFRESH VALUE
B847 72 C LD (HL), D ; STORE RESULT AS NEW SEED
B848 2B C DEC HL
B849 73 C LD (HL), E
B84A E1 C POP HL
B84B C1 C POP BC
B84C C9 C RET

;; GETUNE
;; GET NEXT UNUSED TUNE IN TUNE GROUP AND RETURN ADDRESS
;; OF TUNE HEADER IN (HL).
;; GIVEN: (HL) - STARTING TUNE HEADER ADDRESS
;; RETURNS: (HL) - ADDRESS OF HEADER OF NEXT UNUSED TUNE
;; = 0809H - END OF TUNE GROUP FOUND

B84D 05 C GETUNE: PUSH DE
B84E DD E5 C PUSH IX
B850 E5 C GETNO1: PUSH HL ; PUT ADDRESS IN (IX)
B851 DD E1 C POP IX
B853 DD 56 01 C LD D, (IX+01H) ; GET LENGTH OF TUNE
B855 DD 5E 00 C LD E, (IX+00H) ; TEST FOR END OF TUNE GROUP
B859 7A C LD A, D
B85A B3 C OR E
B85A 2B 0E C JR Z, GETNO3 ; JUMP IF END OF GROUP
B85D DD 7E 09 C LD A, (IX+BITFLG) ; GET FLAG BYTE
B868 CB 77 C BIT ERRBIT, A ; TEST FOR ERROR BIT
B862 20 04 C JR NZ, GETNO2 ; BYPASS IF ERROR FOUND
B864 CB 7F C BIT USEDBIT, A ; TEST USED BIT
B866 20 04 C JR Z, GETNO4 ; JUMP IF NOT USED
B868 19 C GETNO2: ADD HL, DE ; POINT TO NEXT TUNE
B869 10 E5 C JR GETNO1 ; TRY NEXT TUNE
B86B FB C GETNO3: EX DE, HL ; PUT ZERO IN (HL)
B86C DD E1 C GETNO4: POP IX
SETUNE: PUSH BC

SETUNE: PUSH DE

SETUNE: PUSH HL

LD B,(IX+KEYLET) ; GET KEY LETTER
LD C,(IX+KEYACD) ; GET KEY ACCIDENTAL
LD D,(IX+KEYMOD) ; GET KEY MODE
LD A,SETKEY ; SET KEY FUNCTION CODE
CALL MCP ; ESTABLISH KEY
AND A ; TEST FOR ZERO RETURN CODE
JR NZ, SETNO1 ; JUMP IF UNSUCCESSFUL
LD B,(IX+TUNETYPE) ; GET TUNE FORMAT TYPE
LD DE,KLEN ; LENGTH OF TUNE HEADER
LD A,HILO ; HIGHEST/LOWEST NOTE FUNCTION CODE
CALL MCP ; GET HIGHEST/LOWEST NOTES IN TUNE
LD A,C ; GET LOWEST PITCH
LD (TUNLOW),A ; SAVE IT
LD A,B ; GET HIGHEST PITCH
LD (TUNHI),A ; SAVE IT
SUB C ; GET RANGE
LD (TUNRNG),A ; SAVE IT
CALL CNTRNG ; CENTER TUNE IN USER'S RANGE
CALL KEYNOT ; SET KEY NOTE VALUES
XOR A ; SET SUCCESSFUL CODE
SETNO1: POP HL
POP DE
RET

; CNTRNG
C
; CENTER RANGE
C
; CENTERS THE RANGE OF THE TUNE IN THE USER'S VOCAL RANGE.
C
; ASSUMES THAT THE RANGE OF THE TUNE (TUNRNG) AND THE USER'S
C
; RANGE (USRANG) HAVE BEEN CALCULATED.
C
; GIVEN:  NONE
C
; RETURNS:  (A) - COMPLETION CODE
C
; = 00 - SUCCESSFUL
C
; = FF - USER RANGE SMALLER THAN TUNE RANGE
C

BB0A C5 C
CTRANG: PUSH BC

BB0C 3A 0000x C
LD A,(TUNRNG) ;GET RANGE OF TUNE

BB0F 47 C
LD B,A ;SAVE IT

BB18 3A 0000x C
LD A,(USRANG) ;GET USER RANGE

BB1C 90 C
SUB B ;GET DIFFERENCE

BB1E 3A 0004 C
JR NC,CTRAN1 ;JUMP IF USER RANGE SMALLER

BB1E 3E FF C
LD A,0FFH ;USER RANGE TOO SMALL FOR TUNE

BB20 18 18 C
JR CTRN02 ;EXIT WITH ERROR FLAG

BB22 CB 2F C
CTRAN1: SRA A ;PUT HALF THE RANGE DIFFERENCE

BB26 47 C
LD B,A ;AT EACH END OF USERS RANGE

BB2D 3A 0000x C
LD A,(USRLO) ;GET USER'S LOWEST NOTE

BB30 80 C
ADD A,B ;ADD HALF THE DIFFERENCE

BB34 47 C
LD B,A ;SAVE IT

BB38 3A 0000x C
LD A,(TUNLO) ;GET LOWEST NOTE OF TUNE

BB3C 90 C
SUB B ;SUBTRACT LOWEST USER NOTE

BB40 32 0000x C
LD (OFFSET),A ;STORE USER'S INPUT OFFSET

BB44 ED 44 C
NEG ;OPPOSITE DIRECTION FOR PLAYBACK

BB48 47 C
LD B,A ;SEND TO MCP IN (B)

BB4C 3E 26 C
LD A,SETOFS ;SET OFFSET FUNCTION CODE

BB50 CD 0000 C
CALL MCP

BB54 AF C
XOR A ;ZERO RETURN CODE

BB58 C1 C
CTRAN2: POP BC

BB5B C9 C
RET

C
; KEYNOT
C
; CALCULATE KEY NOTE INTEGER AND ADJUSTED KEY NOTE INTEGER
C
; AND STORE THEM IN 'KEYNT3' AND 'KEYN0F' RESPECTIVELY.
C
; GIVEN: (IX) - ADDRESS OF TUNE HEADER
C
; RETURNS: (A) - INTEGER KEY NOTE PLUS USER RANGE OFFSET
C

BB64 C5 C
KEYNOT: PUSH BC

BB68 D5 C
PUSH DE

BB6C DD 46 05 C
LD B,(IX+KEYLET) ;GET TUNE KEY LETTER

BB70 DD 4E 06 C
LD C,(IX+KEYACD) ;GET TUNE KEY ACCIDENTAL
B8DC 16 03 C LD D,3H ;SET IN OCTAVE 3
B8DE 3E 1E C LD A,CVTLTA ;GET ABSOLUTE PITCH
B8E0 CD C000 C CALL MCP
B8E3 47 C LD B,A ;PASS RESULT IN (B)
B8E4 3E 20 C LD A,CVTAI ;CONVERT TO INTEGER
B8E6 CD C000 C CALL MCP
B8E9 32 0000x C LD (KEYNT3),A ;STORE KEY NOTE
B8EC 4F C LD C,A ;SAVE IT
B8ED 3A 0000x C LD A,(OFFSET) ;GET OFFSET FOR USER RANGE
B8F0 81 C ADD A,C ;ADD IT TO KEY NOTE
B8F1 32 00003 C LD (KEYNOF),A ;STORE IT
B8F4 D1 c POP DE
B8F5 Cl c POP BC
B8F6 C9 c RET

; MACLIB B:SSM86.MAC
; QRYKEY
; QUERY KEY
; ASKS THE USER FOR THE KEY OF THE TUNE THAT IS DISPLAYED AND
; CHECKS TO SEE IF IT IS RIGHT. THE USER MUST MAKE AT LEAST
; ONE ATTEMPT; THEREAFTER, HE MAY ASK TO BE TOLD THE ANSWER.
; THIS ROUTINE DOES NOT RETURN UNTIL THE KEY IS KNOWN OR THE
; ESCAPE KEY IS PRESSED.
; GIVEN: (IX) - ADDRESS OF TUNE HEADER
; RETURNS: (A) - COMPLETION CODE
; = 00H - SUCCESSFUL
; = ESC - INTERRUPT

B8F7 C5 C QRYKEY: PUSH BC
B8F8 D5 C PUSH DE
B8F9 E5 C PUSH HL
B8FA 06 05 C QRYKEY: LD B,35H ;PARTITION NUMBER
B8FC CD BD92 C CALL ANCPAR ;ACTIVATE AND CLEAR PARTITION
B8FF 3E 02 C LD A,2H ;ALLOW USER TO SEE CHANGE
B901 CD BECD C CALL DELAY
B904 01 0001 C LD BC,0001H ;X/Y COORDINATE
B907 11 A6C6 C LD DE,MSGW ;WHAT KEY IS THIS TUNE IN?
B90A 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
B90C CD C000 C CALL MCP ;ASK FOR THE KEY
B90F 01 0003 C LD BC,0003H ;X/Y COORDINATE
B912 11 A680 C LD DE,MSGM ;MAJOR - UPPER CASE
B915 3E 34 C LD A,PRTEXT ;DISPLAY TEXT
B917 CD C000 C CALL MCP
B91A 01 000B C LD BC,000BH ; X/Y COORDINATE
B91D 11 A77A C LD DE, MSGSTA ' ' ?' - SHOW THE ANSWER'
B920 3E 34 C LD A, PRTEXT ; DISPLAY TEXT
B922 CD C000 C CALL MCP
B925 3E 83 C LD A, 03H ; PARTITION NUMBER
B927 CD B0A1 C CALL GETPXY ; GET STARTING X/Y COORDINATE
B92A 78 C LD A, B ; GET COLUMN NUMBER
B92B C6 1A C ADD A, IAH ; POINT TO END OF QUESTION
B92D 47 C LD B, A
B92E 80 C DEC C
B92F CD BE5D C CALL SETCUR ; SET SORCERER CURSOR POSITION
B932 3E 08 C LD A, NOFOLD ; DO NOT TRANSLATE LOWER CASE
B934 CD BE1C C CALL RESPON ; GET USER RESPONSE
B937 CD BE84 C CALL KILCUR ; ERASE MONITOR CURSOR
B93A 7E C LD A, (HL) ; CHECK FIRST CHARACTER
B93B FE IB C cp ESC ; CHECK FOR ESCAPE KEY
B93D 28 44 C jr Z, QYKY06 ; EXIT IF ESCAPE KEY
B93F FE 3F C cp ' ?' ; DOES HE WANT TO BE TOLD?
B941 28 2F C jr Z, QYKY04 ; JUMP IF HE DOES
B943 CD B987 C CALL READKY ; ELSE READ HIS ANSWER
B946 FE FF C cp OFFH ; TEST FOR INPUT ERROR
B948 28 16 C jr Z, QYKY02 ; JUMP ON INPUT ERROR
B94A CD B901 C CALL CMPKEY ; ELSE SEE IF ANSWER IS RIGHT
B94D 01 03E C LD BC, 03Eh ; X/Y COORDINATE OF MESSAGE
B950 11 A7B2 C LD DE, MSGSTA 'WRONG. TRY AGAIN.'
B953 A7 C and A ; TEST RETURN CODE
B954 28 10 C jr NZ, QYKY03 ; JUMP IF WRONG ANSWER
B956 11 A7DC C LD DE, MSGTRI ' THAT'S RIGHT'
B959 3E 34 C LD A, PRTEXT ; PRINT TEXT FUNCTION CODE
B95B CD C000 C CALL MCP ; DISPLAY MESSAGE
B95E 18 18 C jr QYKY05 ; AND EXIT
B966 3E 34 C QYKY03: LD A, 03H ; DELAY FACTOR
B968 CD C000 C CALL MCP ; DISPLAY MESSAGE
B96B 18 88 C jr QYKY01 ; START OVER
B96E 01 03E C QYKY02: LD BC, 03Eh ; X/Y COORDINATE
B972 CD B98E C CALL PRTKEY ; TELL HIM THE ANSWER
B975 CD B9EC C CALL WAIT ; WAIT TILL USER IS READY
B978 CD BEA3 C QYKY05: CALL WAIT ; WAIT TILL USER IS READY
B97B 06 05 C LD B, 05H ; PARTITION NUMBER
B97D 3E 02 C LD A, CLRPAR ; CLEAR PARTITION FUNCTION CODE
B97F CD C000 C CALL MCP ; CLEAR IT
B982 AF C xor A ; SET CONTINUE CODE
B985 C1 C QYKY06: POP HL
B984 D1 C POP DE
B983 E1 C POP HL
PROGRAM SECTION

B986  C9  C  RET
    ;
    ; READKY
    ; READS THE CHARACTER INPUT BUFFER POINTED TO BY (HL) TO
    ; DETERMINE THE KEY LETTER, ACCIDENTAL AND MODE INDICATED
    ; BY THE USER.
    ;
    ; GIVEN:  (HL) - ADDRESS OF CHARACTER INPUT BUFFER
    ; RETURNS:  (A) - COMPLETION CODE
    ;          = 00H - SUCCESSFUL
    ;          = FFH - INVALID INPUT
    ;          (B) - KEY LETTER (ALWAYS UPPER CASE)
    ;          (C) - ACCIDENTAL (-1 TO +1)
    ;          (D) - KEY MODE (08-MAJOR; 01-MINOR)

B996  C9

B997  E5  C  READY: PUSH  HL
B998  16 00  C  LD  D,08H  ;START MODE AT MAJOR
B999  7E  C  LD  A,(HL)  ;GET FIRST CHARACTER
B99A  FE 61  C  CP  ‘a’  ;TEST FOR UPPER/LOWER CASE
B99B  30 03  C  JR  C,RDKY01  ;JUMP IF UPPER CASE
B99C  D6 20  C  SUB  20H  ;ELSE CONVERT TO UPPER CASE
B99D  14  C  INC  D  ;MAKE MODE MINOR
B99E  47  C  RDKY01:  LD  B,A  ;SAVE KEY LETTER
B99F  23  C  INC  HL  ;POINT TO ACCIDENTAL
B9A0  7E  C  LD  A,(HL)  ;GET CHARACTER
B9A1  E8 00  C  LD  C,00H  ;FIRST ASSUME NATURAL
B9A2  FE 80  C  CP  CARRET  ;TEST FOR END OF INPUT
B9A3  20 2F  C  JR  Z,RDKY98  ;EXIT IF THAT'S ALL
B9A4  FE 20  C  CP  SPACE  ;TEST FOR SPACE
B9A5  28 04  C  JR  Z,RDKY02  ;JUMP IF FOUND
B9A6  FE 20  C  CP  ‘ ‘  ;CHECK FOR DASH
B9A7  28 02  C  JR  NZ,RDKY83  ;JUMP IF NOT SPACE OR DASH
B9A8  23  C  RDKY02:  INC  HL  ;POINT TO NEXT CHARACTER
B9A9  7E  C  LD  A,(HL)  ;GET IT
B9AA  FE 53  C  RDKY03:  CP  ‘s’  ;TEST FOR SHARP
B9AB  20 08  C  JR  Z,RDKY04  ;JUMP IF SHARP
B9AC  FE 73  C  CP  ‘s’  ;TEST UPPER AND LOWER CASE
B9AD  28 04  C  JR  Z,RDKY04
B9AE  FE 23  C  CP  ‘0’  ;TEST FOR SHARP SYMBOL
B9AF  20 03  C  JR  NZ,RDKY05  ;JUMP IF NOT SHARP
B9B0  8C  C  RDKY04:  INC  C  ;MAKE KEY ACCIDENTAL SHARP
B9B1  18 16  C  JR  RDKY00  ; AND EXIT
B9B2  FE 46  C  RDKY05:  CP  ‘F’  ;TEST FOR UPPER AND LOWER CASE
B9B3  20 11  C  JR  Z,RDKY07  ; FLAT SYMBOLS
B9B4  FE 66  C  CP  ‘f’
;CMPKEY

CMPKEY: PUSH BC
C

LD E,0FFH ;FIRST ASSUME NOT EQUAL
B9D2 C5 C

LD A,(IX+KEYLET) ;GET TUNE KEY LETTER
B9D3 D5 C

LD A,(IX+KEYACD) ;GET TUNE KEY ACCIDENTAL
B9D4 E FF C

LD A,(IX+KEYNOD) ;GET TUNE KEY MODE
B9D5 DD 7E 05 C

LD A,E ;COMPARE WITH GIVEN LETTER
B9D6 DD 7E 06 C

JR NZ,CMPKY01 ;JUMP IF NOT EQUAL
B9D7 B8 C

LD A,(IX+KEYACD) ;GET TUNE KEY ACCIDENTAL
B9D8 B9 C

JR NZ,CMPKY01 ;JUMP IF NOT EQUAL
B9D9 20 0D C

LD A,(IX+KEYNOD) ;GET TUNE KEY MODE
B9DA 20 0E C

LD A,E ;COMPARE WITH GIVEN LETTER
B9DB DD 7E 07 C

JR NZ,CMPKY01 ;JUMP IF NOT EQUAL
B9DC 20 07 C

LD A,(IX+KEYMODE) ;GET TUNE KEY MODE
B9DE 20 06 C

INC E ;ELSE SET EQUAL INDICATOR
B9DF BA C

INC E ;ELSE SET EQUAL INDICATOR
B9E0 20 01 C

INC E ;ELSE SET EQUAL INDICATOR
B9E1 1C C

INC E ;ELSE SET EQUAL INDICATOR
B9E2 7B C

INC E ;ELSE SET EQUAL INDICATOR
B9E3 1C C

INC E ;ELSE SET EQUAL INDICATOR
B9E4 7B C

INC E ;ELSE SET EQUAL INDICATOR
B9E5 1C C

INC E ;ELSE SET EQUAL INDICATOR
B9E6 C9 C

INC E ;ELSE SET EQUAL INDICATOR
B9E7 C9 C

INC E ;ELSE SET EQUAL INDICATOR
B9E8 C9 C

INC E ;ELSE SET EQUAL INDICATOR
B9E9 C9 C

INC E ;ELSE SET EQUAL INDICATOR
B9E8 7B C

INC E ;ELSE SET EQUAL INDICATOR
B9E9 1C C

INC E ;ELSE SET EQUAL INDICATOR
B9ED E FF C
C ; PRTKKEY
C ;
C ; PRINT MESSAGE 'THIS TUNE IS IN THE KEY OF:' AND SHOW
C ; THE KEY.
C ;
C ; GIVEN:  (B) - X COORDINATE OF STARTING POSITION OF TEXT
C ;   (C) - Y COORDINATE OF STARTING POSITION OF TEXT
C ;
C ; RETURNS: NONE
C ;
B9EC D5 C PRTKKEY: PUSH DE
B9ED E5 C PUSH HL
B9EE C5 C PUSH BC ; SAVE LINE/COLUMN
B9EF 21 A833 C LD HL, MSGtti ; 'THIS TUNE IS IN THE KEY OF'
B9F2 CD BEF9 C CALL MQVM SG ; MOVE TO RAM
B9F5 CD BA0C C CALL ASCKEY ; GET KEY CHARACTERS
B9F8 78 C LD A, B ; GET KEY LETTER/MODE
B9F9 32 00 18 C LD (TMPMSG+1BH), A ; STORE IT IN MESSAGE TEXT
B9FC 79 C LD A, C ; GET KEY ACCIDENTAL
B9FD 32 00 1CH C LD (TMPMSG+1CH), A ; STORE IT IN MESSAGE TEXT
BA01 C1 C POP BC ; RESTORE X/Y COORDINATE
BA04 3E 34 C LD A, PRTEXT ; PRINT TEXT FUNCTION CODE
BA06 00 00 C CALL MCP ; DISPLAY THE MESSAGE
BA09 C1 C POP HL
BA0C DD 46 05 C ASCKEY: LD B, (IX+KEYLET) ; GET TUNE KEY LETTER
BA0F DD 7E 07 C LD A, (IX+KEYMOD) ; GET TUNE KEY MODE
BA12 A7 C AND A ; TEST FOR MAJOR
BA13 78 C LD A, B ; GET LETTER
BA14 28 02 C JR z, ASKYO1 ; JUMP IF MAJOR KEY
BA16 C6 20 C ADD A, 20H ; ELSE CONVERT TO LOWER CASE
BA18 47 C ASKYO1: LD B, A ; SAVE LETTER
BA19 00 20 C LD C, SPACE ; START ACCIDENTAL WITH NONE
BA1B DD 7E 06 C LD A, (IX+KEYACD) ; GET TUNE KEY ACCIDENTAL
BA1E FE 01 C CP 01H ; TEST FOR SHARP
BA20  38 08  C  JR   C,ASKY03 ;JUMP IF NATURAL
BA22  28 04  C  JR   Z,ASKY02 ;JUMP IF SHARP
BA24  8E 0E  C  LD   C,MLFLT ;ELSE ASSUME FLAT
BA26  18 02  C  JR   ASKY03 ; AND EXIT
BA28  8E 23  C  ASKY02; LD   C,'#' ;LOAD SHARP CHARACTER
BA2A  C9  C  ASKY03; RET

BA2B  C5  C  PRTHDR; PUSH  BC
BA2C  D5  C  PUSH  DE
BA2D  E5  C  PUSH  HL
BA2E  86 01  C  LD   B,01H ;WHOLE SCREEN PARTITION NUMBER
BA2F  3E 01  C  LD   A,ACTPAR ;ACTIVATE PARTITION
BA31  CD  C000  C  CALL  MCP
BA32  21  AB51  C  LD   HL,KEYHDR ;'KEY: '
BA33  CD  BEF9  C  CALL  MQVHSG ;M0VE TO RAM
BA34  CD  BA8C  C  CALL  ASCKEY ;GET KEY IN 'ASCII'
BA35  78  C  LD   A,B ;GET KEY LETTER
BA36  32 006H$  C  LD   (TMPMSG+06H),A ;STORE KEY LETTER
BA37  79  C  LD   A,C ;GET ACCIDENTAL
BA38  32 008H$  C  LD   (TMPMSG+08H),A ;STORE ACCIDENTAL
BA39  11 000H$  C  LD   DE,TMPMSG ;'KEY: '
BA3A  3E 34  C  LD   A,PRTEXT ;PRINT TEXT FUNCTION CODE
BA3B  CD  C000  C  CALL  MCP
BA3C  E1  C  POP   HL
BA3D  D1  C  POP   DE
BA3E  C1  C  POP   BC
BA40  C9  C  RET

MACLIB  B:SSM07.MAC

QRYTON

QUERY USER TO SING TONIC TRIAD. GIVES HIM THE OPTION OF
HEARING IT PLAYED FOR HIM UP TO THREE TIMES BY PRESSING
THE 'P' KEY. IF HE CAN'T SING IT RIGHT IN THREE TRIYS,
IT CONTINUES AS IF HE HAD.
PROGRAM SECTION

C ; GIVEN: (IX) = ADDRESS OF TUNE HEADER
C ; RETURNS: (A) = COMPLETION CODE
C ;                           = 08H - SUCCESSFUL
C ;                           = ESC - USER INTERRUPT
C ;
BA55 C5 C QRYTON: PUSH BC
BA56 D5 C PUSH DE
BA57 E5 C PUSH HL
BA58 3E 03 C LD A,03H ;MAXIMUM NUMBER OF TIMES
BA59 32 0000X C LD (TEMP00),A ; TO SING IT WRONG
BA5A 06 05 C QYTO01: LD B,05H ;PARTITION NUMBER
BA5F CD BD92 C CALL ANCPR ;ACTIVATE AND CLEAR THE PARTITION
BA62 3E 02 C LD A,02H ;ALLOW USER TIME TO SEE CHANGE
BA64 CD BECD C CALL DELAY
BA67 01 0102 C LD BC,0102H ;X/Y COORDINATE
BA6A 11 A85A C LD DE,MSGSTD ;"SING TONIC TRIAD"
BA6D 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
BA6F CD 0000 C CALL MCP ;DISPLAY MESSAGE
BA72 01 048C C LD BC,048CH ;NEXT LINE/COLUMN START
BA75 11 A079 C LD DE,MSGNUL ;NULL MESSAGE
BA78 3E 34 C LD A,PRTEXT ;SET X/Y COORDINATE
BA7A CD 0000 C CALL MCP
BA7D CD BAF4 C CALL MAKRDR ;SET UP TRIAD IN OCTAVE 3
BA80 21 0000X C LD HL,WRKSPC ; AT THIS ADDRESS
BA83 CD BA8B C CALL GRTD ;LISTEN TO TRIAD
BA86 A7 C AND A ;TEST FOR GOOD TRIAD
BA87 28 18 C JR 2,QYTO03 ;JUMP IF GOOD
BA89 FE 50 C CP 'P' ;CHECK FOR PLAY TRIAD
BA8B 28 86 C JR 2,QYTO02 ;JUMP TO PLAY TRIAD
BA8D FE 1B C CP ESC ;CHECK FOR ESCAPE
BA8F 28 26 C JR 2,QYTO05 ;EXIT IF ESCAPE
BA91 18 CA C JR QYTO01 ;ELSE START TRIAD OVER AGAIN
BA93 CD BB17 C QYTO02: CALL PLTRD ;PLAY THE TRIAD FOR HIM
BA96 3A 0000X C LD A,(TEMP00) ;SET LOOP COUNT
BA99 3D C DEC A ;IF HE CAN'T SING IT RIGHT IN
BA9A 32 0000X C LD (TEMP00),A ; 3 TIMES, GO ON ANYWAY
BA9D 20 BE C JR NZ,QYTO01 ;JUMP TO TRY AGAIN
BA9F 18 BB C JR QYTO04 ;ELSE EXIT
BAA1 01 048E C QYTO03: LD BC,048EH ;X/Y COORDINATE
BAA4 11 A7DC C LD DE,MSGSTRI ;"THAT'S RIGHT"
BAA7 3E 34 C LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
BA9C CD C000 C CALL MCP ;DISPLAY THE MESSAGE
BAAC 3E 04 C QYTO04: LD A,04H ;DELAY FACTOR
BAAE CD BECD C CALL DELAY
BA81 06 05 C LD B,05H ;PARTITION NUMBER
BA83 CD BD92 C CALL ANCPR ;CLEAR THE PARTITION
BA86 AF C XOR A ;SET SUCCESSFUL CODE
BA87 E1 C QYTO05: POP HL
BABB D1 C POP DE
BABB C1 C POP BC
BABB C9 C RET

; GETRAD
;
; GETS TRIAD AS SUNG BY USER AND CHECKS IT TO SEE IF IT IS
; RIGHT. AS EACH PITCH IS SUNG CORRECTLY, ITS SCALE DEGREE
; NUMBER IS DISPLAYED ON THE SCREEN. THE ROUTINE DOES NOT
; RETURN UNTIL EITHER THE TRIAD IS SUNG CORRECTLY OR THE
; ESCAPE KEY HAS BEEN Pressed. ANY OF THE PITCHES MAY BE
; SUNG IN ANY OCTAVE.
;
; GIVEN: (HL) - ADDRESS OF INTEGER TRIAD IN OCTAVE 3
; RETURNS: (A) - COMPLETION CODE
; = 80H - SUCCESSFUL
; = ASCII KEYBOARD CHARACTER ON INTERRUPT
;
BABB C5 C GETRAD; PUSH BC
BABC D5 C PUSH DE
BABB E5 C PUSH HL
BABE 3E 2C C LD A,INITPT ; INITIALIZE INPUT PITCH
BAB8 CD C000 C CALL MCP
BAC9 3E FF C LD A,NOISE ; SET PREVIOUS PITCH FIELD
BAC9 32 0000X C LD (PREVIP),A ; WITH NOISE VALUE
BACB 06 05 C LD B,05H ; NUMBER OF NOTES
BACA 11 A920 C LD DE,MSG135 ; '1 3 5 3 1'
BACD C5 C GTRD01: PUSH BC ; SAVE NOTE COUNT
BACE CD BC83 C GTRD02: CALL LISTEN ; GET NEXT DIFFERENT PITCH
BADD FE FD C CP KBINTR ; CHECK FOR KEYBOARD INTERRUPT
BAD3 20 04 C JR NZ,GTRD03 ; JUMP IF FOUND
BAD5 79 C LD A,C ; GET KEYBOARD CHARACTER
BAD6 C1 C POP BC ; RESTORE STACK POINTER
BAD7 18 17 C JR GTRD07 ; AND EXIT
BAD9 4F C GTRD03: LD C,A ; SAVE INTEGER PITCH
BAAD BE C GTRD04: CP (HL) ; IS IT CORRECT?
BAAB 20 06 C JR Z,GTRD05 ; JUMP IF SO
BAAD 3E EF C JR NC,GTRD02 ; JUMP IF IT IS TOO HIGH
BAAD C6 0C C ADD A,0CH ; ELSE ADD 1 OCTAVE
BAAE 18 F7 C JR GTRD04 ; AND COMPARE AGAIN
BAAB 23 C GTRD05: INC HL ; POINT TO NEXT PITCH
BAAC 01 0000 C LD BC,0000H ; SET X/Y TO CURRENT SETTING
BAAD 3E 34 C LD A,PRTEXT ; PRINT TEXT FUNCTION CODE
BAAE CD C000 C CALL MCP ; PRINT NUMBER OF NOTE JUST SUNG
BAAF C1 C GTRD06: POP BC ; RESTORE NOTE COUNT
BAAD 18 DE C DJNZ GTRD01 ; CONTINUE FOR ALL NOTES
BAAE AF C XOR A ; SET SUCCESSFUL CODE
BAF0  E1  C  GTR087:  POP  HL
BAF1  D1  C  POP  DE
BAF2  C1  C  POP  BC
BAF3  C9  C  RET

BAF4  C5  C  MAKR0:  PUSH  BC
BAF5  D5  C  PUSH  DE
BAF6  E5  C  PUSH  HL
BAF7  B6  05  C  LD  B,05H  ;COUNT NOTES
BAF9  3A  0008  C  LD  A,(KEYNT3)  ;GET KEY NOTE IN OCTAVE 3
BAFC  4F  C  LD  C,A  ;SAVE IT
BAFD  21  A035  C  LD  DE,MAJR0D  ;POINT TO MAJOR TRIAD
BB03  DD  7E  07  C  LD  A,(IX+KEYMOD)  ;GET KEY MODE OF TUNE
BB06  A7  C  AND  A  ;TEST FOR MAJOR
BB07  28  03  C  JR  2,MKR01  ;JUMP IF MAJOR KEY
BB09  11  A03A  C  LD  DE,MINCR0  ;ELSE POINT TO MINOR TRIAD
BB0C  1A  C  MKR01:  LD  A,(DE)  ;GET NOTE OFFSET FROM TONIC
BB0D  13  C  INC  DE  ;POINT TO NEXT OFFSET
BB0E  B1  C  ADD  A,C  ;ADD IT TO TONIC NOTE
BB0F  77  C  LD  (HL),A  ;STORE IT
BB10  23  C  INC  HL  ;POINT TO NEXT LOCATION
BB11  18  F9  C  DJNZ  MKR01  ;CONTINUE FOR ALL 5 PITCHES
BB13  E1  C  POP  HL
BB14  D1  C  POP  DE
BB15  C1  C  POP  BC
BB16  C9  C  RET

BB17  C5  C  PLATR0:  PUSH  BC
BB18  D5  C  PUSH  DE
BB19  E5  C  PUSH HL
BB1A  11 A045  C  LD  DE,RMJ135  ;POINT TO MAJOR TRIAD (RELATIVE)
BB1D  DD 7E 07  C  LD  A,(IX+KEYMOD)  ;GET MODE OF KEY
BB20  A7  C  AND  A  ;TEST FOR MAJOR
BB21  2B 03  C  JR  Z,PLTR01  ;JUMP IF MAJOR KEY
BB23  11 A05F  C  LD  DE,RMN135  ;ELSE POINT TO MINOR TRIAD
BB26  3A 0000%  C  PLTR01:  LD  A,(KEYNT3)  ;GET KEY NOTE INTEGER
BB29  D6 24  C  SUB  24H  ;SUBTRACT 3 OCTAVES
BB2B  47  C  LD  B,A  ;PASS AS ROOT NOTE
BB2C  0E 3C  C  LD  C,3CH  ;TEMPO - QUARTER = 1 SECOND
BB2E  3E 24  C  LD  A,PLAREL  ;PLAY RELATIVE FUNCTION CODE
BB30  CD C000  C  CALL  MCP  ;PLAY THE TRIAD
BB33  E1  C  POP  HL
BB34  D1  C  POP  DE
BB35  C1  C  POP  BC
BB36  C9  C  RET

MACLIB B:SSM08.MAC

CREATE USER RECORD OF SESSION STATISTICS. CREATES 1 OF 4 RECORD TYPES: USER, TUNE GROUP, TUNE AND PERFORMANCE. ALL RECORDS ARE WRITTEN TO A 256 BYTE BUFFER WHICH IS WRITTEN TO TAPE WHEN IT IS FILLED. EACH OF THE 4 RECORD TYPES ARE A FIXED LENGTH OF 16 BYTES, EXCEPT THE TUNE GROUP RECORD WHICH IS 32.

GIVEN: (A) - TYPE RECORD TO CREATE
RETURNS: (A) - COMPLETION CODE
= 00H - SUCCESSFUL
= VALUE GIVEN IF INVALID

BB37  C5  C  CREREC:  PUSH  BC
BB38  D5  C  PUSH  DE
BB39  E5  C  PUSH  HL
BB3A  2A 0000%  C  LD  HL,(RECN0)  ;POINT TO NEXT AVAILABLE
BB3D  EB  C  EX  DE,HL  ;ADDRESS IN BUFFER
BB3E  06 00X  C  LD  B,HL,(RECBUF)  ;DETERMINE IF BUFFER IS FULL
BB40  0E F1  C  LD  C,0F1H  ;ALL TYPES BUT TUNE GROUP
BB42  FE 00  C  CP  TYPGRP  ;NEED ONLY 16 BYTES
BB44  20 02  C  JR  NZ,CREC01  ;JUMP IF NOT TUNE GROUP
BB46  0E 11  C  LD  C,0E1H
BB48  CD 8FCA  C  CREC01:  CALL  DCOMP  ;TEST FOR END OF BUFFER
BB4B  38 85  C  JR  C,CREC02  ;JUMP IF NOT OUT OF ROOM
BB4D  F5  C  PUSH  AF  ;SAVE RECORD TYPE
C5 USEREC: PUSH BC
B987 D5 C PUSH DE
B988 E5 C PUSH HL
B989 2A 0000H C LD HL,(RECEV) ;GET FIRST AVAILABLE SLOT
B98C 3E 00 C LD A,TYPUSR ;STORE TYPE RECORD
B98E 77 C LD (HL),A ; AS FIRST BYTE
B98F 00 C LD HL,(USRNAD) ;PUT BUFFER ADDRESS IN (DE)
B991 01 0000H C INC DE
B994 01 0005 C LD BC,0005H ;LENGTH OF NAME
BB97 ED B0 C LDIR
BB99 21 0000X C LD HL,USRANG ;STORE HIS RANGE, HIGHEST
BB9C 01 0003 C LD BC,0003H ;3 FIELDS TO MOVE
BB9F ED B0 C LDIR
BBA1 21 0000X C LD HL,DATE ;MOVE TODAY'S DATE
BBA4 01 0003 C LD BC,0003H ;LENGTH
BBA7 ED B0 C LDIR
BBB9 EB C EX DE,HL ;PUT BUFFER ADDRESS IN (HL)
BBBB 06 04 C LD B,04H
BBBC 36 00 C USRE01: LD (HL),00H
BBB8 23 C INC HL
BBBF 10 FB C DJNZ USRE01
BBB1 E1 C POP HL
BBB2 D1 C POP DE
BBB3 C1 C POP BC
BBB4 C9 C RET

BBB5 C5 C
BBB6 D5 C
BBB7 E5 C
BBB8 2A 0000X C LD HL,(RECEND) ;POINT TO POSITION IN BUFFER
BBB9 36 00 C LD (HL),TYPGRP ;STORE TYPE RECORD
BBBA 23 C INC HL
BBBB 11 0000X C LD DE,TUNGRP ;COPY ALL OF TUNE GROUP
BBBC EB C EX DE,HL ; HEADER, EXCEPT LAST BYTE
BBBD 01 001F C LD BC,TGHLEN-1 ; TO THE BUFFER
BBBE ED B0 C LDIR
BBBF E1 C POP HL
BBBG D1 C POP DE
BBBH C1 C POP BC
BBBB C9 C RET

BBB85 C5 C GRPREC: PUSH BC
BBB86 D5 C PUSH DE
BBB87 E5 C PUSH HL
BBB88 2A 0000X C LD HL,(RECEND) ;POINT TO POSITION IN BUFFER
BBB89 36 00 C LD (HL),TYPGRP ;STORE TYPE RECORD
BBB9A 23 C INC HL
BBB9B 11 0000X C LD DE,TUNGRP ;COPY ALL OF TUNE GROUP
BBB9C EB C EX DE,HL ; HEADER, EXCEPT LAST BYTE
BBB9D 01 001F C LD BC,TGHLEN-1 ; TO THE BUFFER
BBB9E ED B0 C LDIR
BBB9F E1 C POP HL
BBBG D1 C POP DE
BBBH C1 C POP BC
BBB9C9 C9 C RET

BBB885 C5 C
BBB886 D5 C
BBB887 E5 C
BBB888 2A 0000X C
BBB889 36 00 C
BBB89A 23 C
BBB89B 11 0000X C
BBB89C EB C
BBB89D 01 001F C
BBB89E ED B0 C
BBB89F E1 C
BBB89G D1 C
BBB89H C1 C
BBB89C9 C9 C

; TUNREC

; CREATES A RECORD FOR IDENTIFYING THE TUNE IN USE.
; THE RECORD CONSISTS OF THE RECORD TYPE INDICATOR
FOLLOWED BY ALL BUT THE LAST BYTE OF THE TUNE HEADER.

GIVEN: NONE

RET

PERREC: PUSH BC

PERREC


GIVEN: (IX) - ADDRESS OF TUNE HEADER

RET

PERREC

; GIVEN: NONE

RET
The image contains a page from a document with code listings and comments. The code appears to be written in a programming language, possibly Assembly, and is related to a music program named "SIGHTSINGER'S MIRROR". The comments explain the purpose of the program and its functions, such as storing numbers sung, notes missed, score, user rating, number of playings, and writing these to a tape using the Sorecerer or Monitor TapeSave routine. The code includes instructions for checking if the record buffer is empty, setting a wait message, and using PUSH and POP operations for saving and restoring registers.
CALL DCOMP
JR Z, WRTB02 ;JUMP IF EMPTY
LD DE, SAVCHD+FILENAM ;MOVE USER'S LOGON NAME
LD HL, (USRNAM) ;TO FILE NAME POSITION
LD BC, 0005H ;IN COMMAND LINE
LD ED B8 ;
LD (IY+4D), 55H ;SET FILE TYPE TO 'U'
LD (IY+3D), 48H ;SET BAUD RATE TO 1200
LD DE, RECBUF ;STORE STARTING ADDRESS OF
LD HL, SAVCHD+SAVFIRM ;DATA TO SAVE IN COMMAND
CALL CUTHTA ;LINE IN ASCII
LD HL, (RECEND) ;CHECK FOR FULL BUFFER
LD A, L ;CONDITION AND STORE AN
AND A ;END OF BUFFER INDICATOR
JR Z, WRTB01 ;IF NOT FULL
LD (HL), 0FFH
WRTB01: DECH L ;POINT TO LAST BYTE OF DATA
LD DE, SAVETO ;STORE ENDING ADDRESS OF
EX DE, HL ;DATA IN COMMAND LINE
CALL CUTHTA
LD A, '2' ;SAVE TO TAPE UNIT #2
LD (SAVCHD+UNITND)+A
LD HL, SAVCHD ;MOVE COMMAND TO MONITOR
LD HL, RECBUF ;RESET BUFFER END ADDRESS
LD (RECEND), HL
HRTB62: CALL NOTBSY ;RESET BUSY SIGN
POP IY
POP IX
POP DE
POP BC
RET

LISTEN
LISTENS FOR NEXT PITCH THAT IS DIFFERENT FROM THE PREVIOUS
ONE. THAT IS, DOES NOT ALLOW REPEATED NOTES. WHEN A NEW
PITCH IS FOUND, IT IS STORED IN 'PREV1P' AND THEN THE
INTEGER OFFSET IS ADDED.
SIGHTSINGER'S MIRROR
PROGRAM SECTION

MACRO-80 3.44 09-Dec-81 PAGE 1-86

C ; GIVEN: NONE
C ; RETURNS: (A) - NEXT INTEGER PITCH HEARD
C ; = KBINTR - KEYBOARD INTERRUPT
C ; (B) - PREVIOUS PITCH
C ; (C) - ASCII KEYBOARD CHARACTER ON INTERRUPT
C ; = 00H - IF NO KEYBOARD INTERRUPT
C ;

BC83 FD E5 C LISTEN: PUSH IY ;CALL MONITOR ROUTINE TO
BC85 CD E1A2 C CALL SYNC ; SYNCHRONIZE WITH VIDEO TO
BC88 FD E1 C POP IY ; AVOID SCREEN FLICKER
BC8A 3A 0000% C LD A,(PREVIP) ;GET PREVIOUS INTEGER PITCH
BC8D 47 C LD B,A ;SEND TO ROUTINE
BC8E 3E 27 C LSTN01: LD A,NXTPIT ;NEXT PITCH FUNCTION CODE
BC90 CD C000 C CALL MCP ;LISTEN FOR NEXT PITCH
BC93 FE FD C CP KBINTR {TEST FOR KEYBOARD INTERRUPT
BC95 C8 C RET Z {EXIT IF FOUND
BC96 FE FF C CP NOISE {TEST FOR NOISE
BC98 28 F4 C JR Z,LSTN01 {IGNORE IT
BC99 28 F1 C JR Z,LSTN01 {JUMP IF SAME
BC9D 32 0000% C LD (PREVIP),A ;SAVE NEW INTEGER PITCH
BC9F 4F C LD C,A ;VALID PITCH
BCA1 3A 0000% C LD A, (OFFSET) ;GET INTEGER OFFSET
BCA4 B1 C ADD A,C ;ADD OFFSET
BCA5 8E 00 C LD C,00H ;SET NO KEYBOARD FLAG
BCA7 C9 C RET

C ;
C ; TONOTE
C ;
C ; PLAY TONIC NOTE.
C ;
C ; GIVEN: NONE
C ; RETURNS: NONE
C ;

BCA8 C5 C TONOTE: PUSH BC
BCA9 D5 C PUSH DE
BCA A 11 A03F C LD DE,ONENOT ;POINT TO SINGLE PITCH
BCA D 3A 0000% C LD A, (KEYNT3) ;GET KEY NOTE IN OCTAVE 3
BCB 0 D6 24 C SUB 24H ;PUT IN OCTAVE 0
BCB 2 47 C LD B,A ;SEND AS ROOT NOTE
BCB 3 0E 16 C LD C,16H ;TEMPO IS QUARTER=22
BCB 5 3E 24 C LD A,PLAREL ;PLAY RELATIVE
BCB 7 CD C000 C CALL MCP
BCB A 01 C POP DE
BCB B C1 C POP BC
BCC BC C9 C RET
C

; INITIALIZE TUNE GROUP SO THAT THE USED BIT IN THE
C
; BIT FLAG BYTE OF EACH TUNE IS RESET AND THEREFORE
C
; AVAILABLE FOR USE AGAIN. THE ERROR BIT IS NOT
C
; RESET.
C
; GIVEN: (IY) - ADDRESS OF TUNE GROUP HEADER
C
; RETURNS: NONE
C

BCBD D5 C INITTG: PUSH DE
BCE6 E5 C PUSH HL
BCEF FD E5 C PUSH IY ;PUT TUNE GROUP HEADER
BCC1 CD B187 C CALL CHKNM ;SEE IF TUNE GROUP LOADED
BCC4 A7 C AND A ;NON-ZERO IS NOT LOADED
BCC5 20 19 C JR NZ,INTG02 ;EXIT IF NOT LOADED
BCC7 11 0020 C LD DE,THLEN ;TUNE GROUP HEADER LENGTH
BCCA FD 19 C INTG01: ADD IY,DE ;POINT TO FIRST TUNE
BCCC FD 56 01 C LD D,(IY+01H) ;GET LENGTH OF TUNE
BCCF FD 5E 08 C LD E,(IY+00H)
BCD2 7A C LD A,D ;TEST FOR END OF GROUP
BCD3 B3 C OR E
BCD4 28 84 C JR Z,INTG02 ;EXIT AT END
BCD6 FD 7E 09 C LD A,(IY+BITFLG) ;GET BIT FLAGS BYTE
BCD9 CB BF C RES USEDBT,A ;RESET USED BIT
BCDB FD 77 09 C LD (IY+BITFLG),A
BCEE 18 EA C JR INTG01 ;GO GET NEXT TUNE
BCE8 FD E1 C INTG02: POP IY
BCE2 E1 C POP HL
BCE3 D1 C POP DE
BCE4 C9 C RET

C

; BUSY
C
; DISPLAYS BUSY MESSAGE WHEN PERFORMING ANY PROCESS
C
; THAT REQUIRES A LENGTHY AMOUNT OF TIME.
C
; GIVEN: NONE
C
; RETURNS: NONE
C

BCE5 F5 C BUSY: PUSH AF
BCE6 C5 C PUSH BC
BCE7 D5 C PUSH DE
BCE8 E5 C PUSH HL
BCE9 3E 03 C LD A,CURPAR ;GET CURRENT PARTITION
NOTBSY: ERASES BUSY MESSAGE ON BOTTOM LINE OF SCREEN.

GIVEN:  NONE
RETURNS: NONE

NOTBSY: PUSH AF

LD HL,0F7C0H  ;ADDRESS OF LAST LINE
LD D,H       ;POINT (DE) TO NEXT
INC DE       ;POSITION
LD BC,LEN-1  ;LENGTH OF SCREEN LINE
LD A,SPACE   ;BLANK OUT FIRST SPACE
LD (HL),A    ;BLANK OUT REST OF LINE
DIR
POP HL
POP DE
POP BC
POP AF
RET
MACLIB B:SSM10.MAC
C ; QRYOPT
C
C ; QUERY USER FOR CHOICE OF ACTION AS EXPLAINED IN THE
C ; TEXT DISPLAYED. THE ADDRESS OF THE TEXT IS GIVEN AND
C ; IS DISPLAYED STARTING ON THE SECOND LINE AND FOURTH
C ; COLUMN OF THE ENTIRE SCREEN. THE WHOLE SCREEN IS
C ; ERASED ON ENTRY TO AND EXIT FROM THIS ROUTINE. THE
C ; USER'S RESPONSE IS ASSUMED TO BE LIMITED TO ONE
C ; CHARACTER. THE LIST OF POSSIBLE ANSWERS IS ASSUMED TO
C ; BE PROVIDED IN THE SPACE IMMEDIATELY FOLLOWING THE END
C ; OF STRING CHARACTER AT THE END OF THE MESSAGE. THIS
C ; LIST MAY BE ANY LENGTH BUT MUST BE TERMINATED BY FFH.
C ; IF HE RESPONDS WITH A CARRIAGE RETURN ONLY, THE
C ; DEFAULT OPTION, ALWAYS THE FIRST ELEMENT IN THE LIST,
C ; IS RETURNED.
C
C ; GIVEN: (DE) - ADDRESS OF MESSAGE TO DISPLAY
C ; RETURNS: (A) - CHARACTER FOR CHOSEN OPTION
C ;
BD23 C5 C QRYOPT: PUSH BC
BD24 D5 C PUSH DE
BD25 E5 C PUSH HL
BD26 ED 53 0000X C LD (TEMP01),DE ;SAVE ADDRESS OF MESSAGE
BD2A 06 01 C QYOP01: LD B,01H ;ACTIVATE AND CLEAR ENTIRE
BD2C CD BD92 C CALL ANCPAR ; SCREEN
BD2F 3E 02 C LD A,02H ;DELAY FACTOR
BD31 CD BECD C CALL DELAY
BD34 01 0002 C LD BC,0002H ;X/Y COORDINATE
BD37 3E 34 C LD A,PRTEXT ;DISPLAY MESSAGE
BD39 CD C000 C CALL MCP
BD3C 3E 31 C LD A,GETCP ;GET X/Y COORDINATE
BD3E CD C000 C CALL MCP ; WHERE LEFT OFF
BD41 05 C DEC B
BD42 0D C DEC C
BD43 CD BESD C CALL SETCUR ;PUT CURSOR THERE
BD46 3E 7F C LD A,FLDFLG ;FOLD LOWER CASE TO UPPER CASE
BD48 CD BEIC C CALL RESPOP ;GET USER RESPONSE
BD4B 7E C LD A,(HL) ;GET RESPONSE CHARACTER
BD4C EB C EX DE,HL ;PUT VALID RESPONSE LIST IN (HL)
BD4D 57 C LD D,A ;KEEP RESPONSE
BD4E FE 8D C CP CARRET ;DEFAULT VALUE?
BD58 20 03 C JR NZ,QYOP02 ;JUMP IF NOT
BD52 56 C LD D,(HL) ;GET DEFAULT ANSWER
BD53 18 20 C JR QYOP03 ;EXIT
BD55 47 C QYOP02: LD B,A ;SEND RESPONSE CHARACTER
BD56 CD BD91 C CALL INLIST ;SEE IF RESPONSE IS VALID
BD59 A7 C AND A ;TEST FOR VALID
BD5A 28 19 C JR Z,QYOP03 ;JUMP IF VALID
INLIST  

CHECK TO SEE IF A GIVEN BYTE IS INCLUDED ANYWHERE IN THE LIST WHOSE ADDRESS IS ALSO GIVEN.

GIVEN:  (B) - CHARACTER UNDER TEST
         (HL) - ADDRESS OF LIST (TERMINATED BY FFH)

RETURNS: (A) - INCLUSION FLAG
         = 00H - CHARACTER IS IN LIST
         = FFH - CHARACTER IS NOT IN LIST

**Program Section**

```assembly
BD5C CD BE84 C CALL KILCUR ; ERASE CURSOR
BD5F 01 041D C LD BC,041D ; X/Y COORDINATE
BD62 11 AD22 C LD DE,MSGINV ; 'INVALID OPTION'
BD65 3E 34 C LD A,PRTEXT ; DISPLAY MESSAGE
BD67 CD C000 C CALL MCP
BD6A 3E 05 C LD A,05H ; DELAY FACTOR
BD6C CD B000 C CALL MCP
BD6F ED 5B 0000x C LD DE,(TEMP81) ; RESET MESSAGE ADDRESS
BD73 18 85 C JR QYOP81 ; DO IT OVER AGAIN
BD75 06 FF C QYOP83: LD B,0FFH ; CLEAR SCREEN
BD77 3E 02 C LD A,CLRPAR
BD79 CD C000 C CALL MCP
BD7C 7A C LD A,0 ; GET ANSWER
BD7D E1 C POP HL
BD7E D1 C POP DE
BD7F C1 C POP BC
BD80 C9 C RET
C ;
C ; INLIST
C ;
C ; CHECK TO SEE IF A GIVEN BYTE IS INCLUDED ANYWHERE IN THE LIST WHOSE ADDRESS IS ALSO GIVEN.
C ;
C ; GIVEN:  (B) - CHARACTER UNDER TEST
C ;         (HL) - ADDRESS OF LIST (TERMINATED BY FFH)
C ;
C ; RETURNS: (A) - INCLUSION FLAG
C ;         = 00H - CHARACTER IS IN LIST
C ;         = FFH - CHARACTER IS NOT IN LIST
C ;
BD81 C5 C INLIST: PUSH BC
BD82 E5 C PUSH HL
BD83 7E C INLS01: LD A,(HL) ; GET LIST ELEMENT
BD84 FE FF C CP 0FFH ; TEST FOR END OF LIST
BD86 28 07 C JR Z,INLS03 ; EXIT AT END OF LIST
BD88 BB C CP B ; TEST WITH GIVEN CHARACTER
BD89 28 03 C JR Z,INLS02 ; JUMP IF SAME
BD8B 23 C INC HL ; POINT TO NEXT LIST ELEMENT
BD8C 18 F5 C JR INLS01 ; TRY NEXT CHARACTER
BD8E AF C INLS02: XOR A ; SET FOUND FLAG
BD8F E1 C INLS03: POP HL
BD90 C1 C POP BC
BD91 C9 C RET
C ;
C ; ANCPAR
C ;
C ; ACTIVATE AND CLEAR PARTITION
```
PROGRAM SECTION

; GIVEN: (B) - PARTITION NUMBER
; RETURNS: NONE

BD92 C5 C ANCPAR: PUSH BC
BD93 D5 C PUSH DE
BD94 3E 01 C LD A,ACTPAR ;ACTIVATE PARTITION FUNCTION CODE
BD95 CD C000 C CALL MCP ;ACTIVATE THE GIVEN PARTITION
BD96 3E 82 C LD A,CLRPAR ;CLEAR PARTITION FUNCTION CODE
BD97 CD C000 C CALL MCP ;CLEAR THE GIVEN PARTITION
BD98 D1 C POP DE
BD99 C1 C POP BC
BDAA C9 C RET

; GETPXY
; GET PARTITION X/Y COORDINATE
; GETS THE STARTING LINE AND COLUMN POSITIONS OF THE PARTITION
; RELATIVE TO THE PHYSICAL SCREEN FROM THE ORIGINAL PARTITION
; DEFINITION. ASSUMES THAT ALL PARTITION DEFINITION BLOCKS ARE CONTIGUOUS AND SEQUENTIAL.

BDAC D5 C GETPXY: PUSH DE
BDAD E5 C PUSH HL
BDAE 47 C LD B,A ;PUT PARTITION NUMBER IN B
BDAF 11 0008 C LD DE,PDBLEN ;PARTITION DEFINITION LENGTH
BDAG 21 A004 C LD HL,PARTR010+01H ;POINT TO FIRST PARTITION DEFINITION
BDAH 05 C DEC B ;MAKE RELATIVE TO 0
BDAB 28 83 C JR Z,GTXY02 ;JUMP IF FIRST PARTITION
BDAC 19 C GTXY01: ADD HL,DE ;POINT TO NEXT PARTITION
BDAD 18 FD C DJNZ GTXY01
BDAE 46 C GTXY02: LD B,(HL) ;GET STARTING COLUMN OF PARTITION
BDAF 23 C INC HL ;POINT TO ROW
BDAG 4E C LD C,(HL) ;GET STARTING ROW OF PARTITION
BDAH E1 C POP HL
BDAB D1 C POP DE
BDAC C9 C RET

; STATMS
; STATISTICS MESSAGE. GETS THE NUMBER OF NOTES MISSED AND TOTAL NOTES IN TUNE AND CALCULATES THE PERCENTAGE
MACRO-80 3.44 09-Dec-81 PAGE 1-92

SIGHTSINGER'S MIRROR

PROGRAM SECTION

; CORRECT, THE VALUES ARE CONVERTED TO ASCII DIGITS AND
; STORED IN THE APPROPRIATE FIELDS FOR DISPLAY.
; GIVEN: IX - ADDRESS OF TUNE HEADER
; RETURNS: NONE

; STATMSG: PUSH BC
; PUSH DE
; PUSH HL
; LD A, (ERRCNT) ; GET NUMBER IN ERROR
; LD C, A ; SAVE IT FOR LATER
; LD E, A ; PUT COUNT IN MESSAGE FIELD
; LD D, 00H
; LD HL, TMPMSG+8DH ERROR COUNT FIELD IN MESSAGE
; CALL SETNUM ; PUT ASCII DIGITS IN FIELD
; LD A, (IX+NUMNOT) ; DO THE SAME WITH THE TOTAL
; LD E, A ; NUMBER OF NOTES IN TUNE
; SUB C ; NUMBER OF NOTES RIGHT
; LD D, 88H
; LD HL, TMPMSG+18H CALL SETNUM
; LD B, 00H MULTIPLY CORRECT COUNT
; CALL HLTPLY RESULT BY THE TOTAL
; LD A, (IX+NUMNOT) NUMBER OF NOTES IN TUNE
; LD C, A TO GET PERCENT CORRECT
; LD B, 88H
; PUSH BC SAVE TOTAL NOTES
; CALL DIVIDE
; EX DE, HL PUT PERCENTAGE IN (HL)
; INC HL DECREMENT FOR COMPARE
; SLA C DOUBLE THE REMAINDER TO
; RL B COMPARE WITH THE DIVISOR
; CALL DCOMP TO TEST FOR ROUND UP
; JR NC, STHG81 JUMP IF NO ROUND UP
; INC HL ELSE ROUND IF
; LD A, L GET FINAL PERCENTAGE
; LD (SCORE), A SAVE IT
; EX DE, HL STORE IT IN MESSAGE
; LD HL, TMPMSG+30H CALL SETNUM
; POP H
; POP DE
; POP BC
; RET
SETNUM

CONVERTS BINARY VALUE GIVEN TO ASCII AND STORES THE ASCII DIGITS STARTING AT THE ADDRESS GIVEN.

GIVEN: (DE) - BINARY VALUE

(FL) - STARTING ADDRESS FOR DIGITS

RETURNS: (FL) - CHANGED

MACLIB B:SSM11.MAC

MONPRT

MONITOR PRINT TEXT ROUTINE. SETS (IY) TO THE ADDRESS OF THE SORCERER MONITOR WORK AREA (MMW) AND CALLS THE MONITOR ROUTINE WHICH PRINTS TEXT POINTED TO BY (HL) ON THE SCREEN AT THE CURRENT CURSOR POSITION. THE STRING MUST BE TERMINATED BY 08H.

GIVEN: (HL) - ADDRESS OF STRING

RETURNS: (HL) - ADDRESS OF END OF STRING INDICATOR

WMACLIB B:SSM11.MAC

RESPON

GET USER KEYBOARD RESPONSE AND STORE CHARACTERS IN BUFFER USING THE SORCERER MONITOR KEYBOARD INPUT ROUTINE. ASSUMES
THAT THE SORCERER CURSOR POSITION HAS BEEN SET AND DISPLAYED. THE INPUT BUFFER LOCATION IS MAINTAINED BY THE SORCERER MONITOR IN THE MONITOR WORK AREA (MWA). THE ROUTINE FIRST CHECKS FOR ONE AS THE FIRST CHARACTER AND STORES IT IF FOUND. OTHERWISE, IT GIVES CONTROL TO THE MONITOR ROUTINE. THE TRANSLATION FROM LOWER CASE TO UPPER CASE CAN BE REQUESTED BY SENDING THE APPROPRIATE INDICATOR IN (A).

GIVEN: (A) - KEYBOARD INPUT DRIVER FLAG
          = NOFOLD - NORMAL KEYBOARD INPUT DRIVER
          = FLDFLG - FOLD LOWER CASE TO UPPER CASE

RETURNS: (HL) - ADDRESS OF CHARACTER INPUT BUFFER

BE1C C5 C RESPON: PUSH BC
BE1D D5 C PUSH DE
BE1E FD E5 C PUSH IY
BE20 CD E1A2 C CALL FNDDWA ;SET (IY) TO ADDRESS OF MWA
BE23 11 BEDD C LD DE,KBFOLD ;KEYBOARD INPUT DRIVER
BE26 FE 7F C CP FLDFLG ;TEST FOR FOLD OVER FLAG
BE28 28 83 C JR Z,RSPN01 ;JUMP IF FOLDING
BE2A 11 BEE9 C LD DE,KBINPT ;ELSE USE NORMAL DRIVER
BE2D CD E631 C RSPN01: CALL SETINP
BE30 FD 36 43 00 C LD (IY+43H),08H ;SET NON-BATCH MODE FLAG
BE34 FD E5 C PUSH IY ;PUT MWA ADDRESS IN (HL)
BE36 E1 C POP HL
BE37 3E 3C C LD A,3CH ;BUFFER LENGTH
BE39 85 C ADD A,L ;LAST BUFFER ADDRESS
BE3A 4F C LD C,A ;SAVE IT IN C
BE3B 45 C LD B,L ;KEEP STARTING POSITION
BE3C CD E009 C CALL GETCHR ;GET STARTING POSITION
BE3F FE 83 C CP ETX ;CHECK FOR CONTROL/C
BE41 28 89 C JR Z,RSPN02 ;JUMP IF FOUND
BE43 FE 1B C CP ESC ;ELSE CHECK FOR ESCAPE
BE45 28 85 C JR Z,RSPN02 ;JUMP IF FOUND
BE47 CD E14D C CALL MNINP ;CALL MONITOR INPUT ROUTINE
BE4A 18 83 C JR RSPN03 ;EXIT
BE4C FD 77 80 C RSPN02: LD (IY+08H),A ;STORE ETX/ESC IN BUFFER
BE4F 11 BEE9 C RSPN03: LD DE,KBINPT ;RESET KEYBOARD DRIVER
BE52 CD E631 C CALL SETINP
BE55 FD E5 C PUSH IY ;PUT ADDRESS OF BUFFER IN (HL)
BE57 E1 C POP HL
BE59 FD E1 C POP IY
BE5A D1 C POP DE
BE5B CI C POP BC
BE5C C9 C RET
PROGRAM SECTION

SETCUR

; SET CURSOR POSITION FOR SORCERER MONITOR AND DISPLAY IT AT THAT POSITION.

; GIVEN: (B) - X COORDINATE OF CURSOR ON PHYSICAL SCREEN

; (C) - Y COORDINATE OF CURSOR ON PHYSICAL SCREEN

; RETURNS: NONE

BE50 C5 C SETCUR: PUSH BC
BE5E D5 C PUSH DE
BE5F FD E5 C PUSH IY
BE61 CD E1A2 C CALL FNDMWA ; GET MONITOR WORK AREA ADDRESS
BE64 FD 70 6A C LD (IY+6AH),B ; STORE COLUMN IN MWA
BE67 FD 36 6B 00 C LD (IY+6AH),00H
BE6B 59 C LD E,C ; SET ROW AND MULTIPLY IT
BE6C 16 80 C LD D,00H ; BY THE LINE LENGTH
BE6E 06 86 C LD B,66H
BE70 CB 23 C STCR01: SLA E ; SHIFT LEFT 6 BITS
BE72 CB 12 C RL D
BE74 18 FA C DJNZ STCR01
BE76 FD 73 6B C LD (IY+6BH),E ; STORE RESULT FOR COLUMN
BE79 FD 72 69 C LD (IY+6BH),D ; POSITION IN MWA
BE7C CD E9CC C CALL MOVCUR ; PUT CURSOR AT THAT PLACE
BE7F FD E1 C POP IY
BE81 D1 C POP DE
BE82 C1 C POP BC
BE83 C9 C RET

KILCUR

; ERASE SORCERER MONITOR CURSOR.

; GIVEN: NONE

; RETURNS: NONE

BE84 D5 C KILCUR: PUSH DE
BE85 E5 C PUSH HL
BE86 FD E5 C PUSH IY
BE88 CD E1A2 C CALL FNDMWA ; SET (IY) TO MWA
BE8B FD 5E 6B C LD E,(IY+6AH) ; GET DISPLACEMENT OF
BE8E FD 56 69 C LD D,(IY+69H) ; CURRENT CURSOR POSITION
BE91 FD 6E 6A C LD L,(IY+6AH) ; FROM TOP OF SCREEN
BE94 FD 66 6B C LD H,(IY+6BH)
BE97 19 C ADD HL,DE
DISPLAY THE MESSAGE 'PRESS ANY KEY TO CONTINUE' ON THE BOTTOM LINE OF THE SCREEN AND WAIT FOR ANY KEY TO BE PRESSED.

GIVEN: NONE

RETURNS: (A) - UNCHANGED

WAIT: PUSH AF
       PUSH BC
       PUSH DE
       PUSH HL
       LD A,CURPAR ;QUERY CURRENT ACTIVE PARTITION
       CALL MCP
       LD H,A ;SAVE IT
       LD B,01H ;ACTIVATE WHOLE SCREEN
       LD A,ACTPAR
       CALL MCP
       LD BC,141EH ;LAST LINE OF SCREEN
       LD DE,MSGPAK ;'PRESS ANY KEY TO CONTINUE'
       LD A,PRTEXT ;PRINT TEXT FUNCTION CODE
       CALL MCP
       LD H,0 ;RESTORE PREVIOUS PARTITION
       LD A,ACTPAR
       CALL MCP
       CALL GETCHR ;WAIT FOR KEYBOARD RESPONSE
       POP HL
       POP DE
       POP BC
       POP AF
       RET

; DELAY
; DELAYS FURTHER PROCESSING BY AN AMOUNT OF TIME THAT IS A FUNCTION OF THE VALUE GIVEN. THE SMALLER THE VALUE GIVEN, THE SHORTER THE TIME DELAYED.
PROGRAM SECTION

C
C; GIVEN: (A) - DELAY FACTOR
C; RETURNS: NONE

BEC0 C5
C DELAY: PUSH BC
BEC3 D5
C PUSH DE
BECF 47
C LD B4A
BED0 11 0000
C DLAY01: LD DE,0000H
BED3 1B
C DLAY02: DEC DE
BED4 7A
C LD A4D
BED5 B3
C OR E
BED6 28 FB
C JR NZ,DLAY02
BED8 1B F6
C DJNZ DLAY01
BEDA D1
C POP DE
BEDB C1
C POP BC
BEDC C9
C RET

; KBGOLD

; KEYBOARD DRIVER WITH LOWER TO UPPER CASE TRANSLATION.
; EITHER THIS ROUTINE OR 'KBINPT' IS SET AT ALL TIMES
; AS THE SYSTEM KEYBOARD DRIVER. BOTH OF THESE DO NOT
; RETURN UNTIL A CHARACTER HAS BEEN PRESS.

; GIVEN: NONE
; RETURNS: (A) - CHARACTER

BEE9 CD E018
C KBINPT: CALL KEYBRD ;CALL MONITOR KEYBOARD DRIVER
BEEC 2B FB
C JR 2,KBINPT ;WAIT UNTIL KEY IS Pressed
FILTER OUT FORM FEEDS SINCE THESE RESET ALL GRAPHICS

TRANSLATE UNSHIFTED RUBOUT TO SHIFTED RUBOUT FOR CONVENIENCE

NULL DEVICE DRIVER USED FOR GETTING THE USER'S PASSWORD.
SETTING THE OUTPUT DEVICE DRIVER TO THIS ADDRESS KEEPS OUTPUT FROM BEING DISPLAYED ON THE SCREEN.

MOVE MESSAGE FROM ROM TO RAM SO THAT PARTS OF IT MAY BE EDITED BEFORE DISPLAY. MESSAGE MUST BE TERMINATED BY 00H. THE MESSAGE IS ALWAYS MOVED TO TMPMSG.

GIVEN: (HL) - ADDRESS OF MESSAGE TO MOVE
RETURNS: NONE

DESTINATION OF MOVE

STORE IN RAM

EXIT WHEN FOUND

POINT TO NEXT CHARACTER

CONTINUE

CONVERTS 4 HEX DIGITS TO ASCII AND STORES THE RESULT
STARTING AT THE ADDRESS GIVEN IN (HL).

GIVEN: (DE) - 4 HEX DIGITS

(HL) - DESTINATION ADDRESS OF RESULT

RETURNS: NONE

CVTHTA: PUSH BC

PUSH DE

PUSH HL

LD B,D ;GET HEX DATA

LD C,E

LD A,B ;CONVERT 2 DIGITS AT A TIME

CALL HEXASC ;CONVERT

LD (HL),D ;STORE RESULT

INC HL

LD (HL),E

INC HL

LD A,C ;GET NEXT 2 DIGITS

CALL HEXASC ;CONVERT

LD (HL),D ;STORE RESULT

LD (HL),E

INC HL

INC HL

LD (HL),D

LD (HL),E

POP HL

POP DE

POP BC

RET

HEXASC: PUSH BC

PUSH HL

LD HL,TEMP ;WORK AREA

LD (HL),A ;SAVE HEX DATA

LD B,02H ;NUMBER OF HEX DIGITS

RLD ;GET HEX DIGIT

AND 0FH ;STRIP OFF HIGH DIGIT

LD C,30H ;FIRST ASSUME 0 - 9

CP 0AH ;TEST FOR THAT RANGE

JR C,HEXAS02 ;JUMP IF 0 - 9

SUB 09H ;ELSE ADJUST 1 TO 6

LD C,40H ;MUST BE LETTER A - F

HEXAS01: RLD
BF39 B1 C HXAS82: OR C ;MAKE ASCII
BF3A 53 C LD D,E ;STORE RESULT
BF3B 5F C LD E,A
BF3C 10 ED C DJNZ HXAS81 ;NEXT DIGIT
BF3E E1 C POP HL
BF3F C1 C POP BC
BF40 C9 C
BF41 C5 C PACK: PUSH BC ;GET LOW ORDER DIGIT
BF42 79 C LD A,C ;STRIP OFF HIGH ORDER 4 BITS
BF43 E6 0F C AND 0FH ;MOVE LOW ORDER 4 BITS
BF45 CB 20 C SLA B ;OF (B) TO HIGH ORDER
BF46 CB 20 C SLA B ;POSITION
BF47 CB 20 C SLA B
BF48 CB 20 C SLA B
BF49 B0 C OR B ;PUT THE 2 TOGETHER
BF4E C1 C POP BC
BF4F C9 C
BF4A C5 C
BF4B D5 C COMPARE: PUSH BC
BF4C E5 C PUSH HL
BF4D 1A C CMR01: LD A,(DE) ;GET BYTE OF STRING 1
BF4E 8E C CP (HL) ;COMPARE TO STRING 2
BF4F 20 07 C JR NZ,CMR02 ;EXIT IF NOT EQUAL
CONCERTS BINARY VALUE TO ASCII DIGITS AND STORES THE RESULT
IN 'DIGITS' RIGHT JUSTIFIED. A MAXIMUM OF 5 DIGITS IS ALLOWED.

GIVEN: (DE) - BINARY VALUE TO CONVERT
RETURNS: (A) - NUMBER OF SIGNIFICANT DIGITS
          (HL) - ADDRESS OF FIRST NON-ZERO DIGIT

BINASC: PUSH BC
        PUSH DE
        LD HL, DIGITS+04H ; POINT TO LOW-ORDER DIGIT
        LD A, 05H ; NUMBER OF DIGITS
        CALL DIVIDE
        LD A, C ; GET REMAINDER
        OR 38H JCOWERT IT TO ASCII
        #ID 3FH ; GET RID OF ANY GARBAGE
        LD (HL), A ; STORE DIGIT
        DEC HL ; POINT TO NEXT DIGIT
        LD A, (TEMP82) ; GET DIGIT COUNT
        DEC A ; COUNT DOWN TO ZERO
        JR NZ, BNASC01 ; JUMP IF NOT 5 YET
        INC B ; POINT TO LAST DIGIT IF ZERO
        INC HL ; POINT TO NEXT DIGIT
        CP (HL) ; TEST FOR '0'
        JR NZ, BNASC83 ; JUMP IF NOT ZERO
        DJNZ BNASC82 ; ELSE TRY NEXT DIGIT
        INC B ; POINT TO LAST DIGIT IF ZERO
        LD A, B ; GET NUMBER OF DIGITS
        POP DE
        POP BC
        RET
C ;
C ;
C ; MLTPLY
C ;
C ; MULTIPLY (BC) BY (DE).
C ;
C ; GIVEN: (BC) - MULTIPLIER
C ; (DE) - MULTIPLICAND
C ;
C ; RETURNS: (A) - UNCHANGED
C ; (BC) - CHANGED
C ; (DE) - PRODUCT
C ;
BF0E F5 C MLTPLY: PUSH AF
BF0F E5 C PUSH HL
BF10 21 0000 C LD HL,0000H
BF11 3E 10 C LD A,10H
BF12 CB 43 C MULT01: BIT 0,E
BF13 28 01 C JR Z,MULT02
BF14 09 C ADD HL,DC
BF15 CB 2C C MULT02: SRA H
BF16 CB 1D C RR L
BF17 CB 1A C RR D
BF18 CB 1B C RR E
BF19 3D C DEC A
BF1A 20 F0 C JR NZ,MULT01
BF1B E1 C POP HL
BF1C F1 C POP AF
BF1D C9 C RET
C ;
C ; DIVIDE
C ;
C ; DIVIDE (DE) BY (BC) RETURNING THE QUOTIENT AND REMAINDER.
C ;
C ; GIVEN: (BC) - DIVISOR
C ; (DE) - DIVIDEND
C ;
C ; RETURNS: (A) - UNCHANGED
C ; (BC) - REMAINDER
C ; (DE) - QUOTIENT
C ;
BF1E F5 C DIVIDE: PUSH AF
BF1F E5 C PUSH HL
BF20 21 0000 C LD HL,0000H
BF21 3E 10 C LD A,10H
BF22 CB 23 C DIV01: SLA E
BF23 CB 12 C RL D
BF24 CB 15 C RL L
BF25 CB 14 C RL H
```assembly
BFB7 A7 C AND A
BFB8 ED 42 C SBC HL,BC
BFBA CB 7C C BIT 7,H
BFBC 20 03 C JR NZ,DVID02
BFBE 13 C INC DE
BFBF 18 01 C JR DVID03
BFC1 09 C DVID02: ADD HL,BC
BFC2 3D C DVID03: DEC A
BFC3 20 EA C JR NZ,DVID01
BFC5 44 C LD B,H
BFC6 40 C LD C,L
BFC7 E1 C POP HL
BFC8 F1 C POP AF
BFC9 C9 C RET

; DCOMP
; DOUBLE COMPARE: (DE) TO (BC)
; GIVEN: (BC) - VALUE 1
; (DE) - VALUE 2
; RETURNS: (A) - UNCHANGED
; C - CARRY FLAG SET/RESET ACCORDING TO COMPARISON
; Z - ZERO FLAG SET/RESET ACCORDING TO COMPARISON

BFCA 32 0000% C DCOMP: LD (TEMP02),A
BFCD 7A C LD A,D
BFCF B8 C CP B
BFCF 20 02 C JR NZ,DMP01
BFD1 7B C LD A,E
BFD2 B9 C CP C
BFD3 3A 0000% C DMP01: LD A,(TEMP02)
BFD6 C9 C RET

; DEPHASE
; ORG CORG+11FAH ;ASSUMES CORB AT XE00H

11FA' C3 A000 JP BASE
11FD' C3 AE00 JP COLD

; SUBTTL SYMBOL TABLE
; END
```
<table>
<thead>
<tr>
<th>Symbols:</th>
<th>Macros:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0009 ACDLIM</td>
<td>002F ACDNKY</td>
</tr>
<tr>
<td>0006 ACK</td>
<td>001A ACDNTL</td>
</tr>
<tr>
<td>B683 ADRT01</td>
<td>B695 ADJRAT</td>
</tr>
<tr>
<td>BA18 ASKY01</td>
<td>BAF2 ASKY03</td>
</tr>
<tr>
<td>0024 ASTRSK</td>
<td>B660 BARADJ</td>
</tr>
<tr>
<td>0008 BALIN</td>
<td>B608 BLOOHL</td>
</tr>
<tr>
<td>BF62 BINASC</td>
<td>BF3A BINAS03</td>
</tr>
<tr>
<td>BF69 BNAS01</td>
<td>BF3A BINAS03</td>
</tr>
<tr>
<td>0008 BS</td>
<td>B608 BLOOHL</td>
</tr>
<tr>
<td>000D CARRET</td>
<td>B187 CHKNAM</td>
</tr>
<tr>
<td>B191 CKMK01</td>
<td>B19C CKMK03</td>
</tr>
<tr>
<td>B1A0 CKMK04</td>
<td>B197 CKMK04</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B198 CKMK05</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B199 CM{{{X2</td>
</tr>
<tr>
<td>0008 BS</td>
<td>B19A CM{{{X3</td>
</tr>
<tr>
<td>000D CARRET</td>
<td>B19B CM{{{X4</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B19C CM{{{X5</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B19D CM{{{X6</td>
</tr>
<tr>
<td>0008 BS</td>
<td>B19E CM{{{X7</td>
</tr>
<tr>
<td>000D CARRET</td>
<td>B19F CM{{{X8</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A0 CM{{{X9</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A1 CM{{{X10</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A2 CM{{{X11</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A3 CM{{{X12</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A4 CM{{{X13</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A5 CM{{{X14</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A6 CM{{{X15</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A7 CM{{{X16</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A8 CM{{{X17</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A9 CM{{{X18</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A0 CM{{{X19</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A1 CM{{{X20</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A2 CM{{{X21</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A3 CM{{{X22</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A4 CM{{{X23</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A5 CM{{{X24</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A6 CM{{{X25</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A7 CM{{{X26</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A8 CM{{{X27</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A9 CM{{{X28</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X29</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X30</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X31</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X32</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X33</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X34</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X35</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X36</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X37</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X38</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X39</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X40</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X41</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X42</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X43</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X44</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X45</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X46</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X47</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X48</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X49</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X50</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X51</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X52</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X53</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X54</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X55</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X56</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X57</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X58</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X59</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X60</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X61</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X62</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X63</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X64</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X65</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X66</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X67</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X68</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X69</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X70</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X71</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X72</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X73</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X74</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X75</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X76</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X77</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X78</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X79</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X80</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X81</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X82</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X83</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X84</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X85</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X86</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X87</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X88</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X89</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X90</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A1 CM{{{X91</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A2 CM{{{X92</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A3 CM{{{X93</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A4 CM{{{X94</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A5 CM{{{X95</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A6 CM{{{X96</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A7 CM{{{X97</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A8 CM{{{X98</td>
</tr>
<tr>
<td>0006 CLPAPP</td>
<td>B1A9 CM{{{X99</td>
</tr>
<tr>
<td>0008 CM{{{X</td>
<td>B1A0 CM{{{X100}</td>
</tr>
<tr>
<td>Symbol</td>
<td>Function</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td>B28A</td>
<td>FRMD01</td>
</tr>
<tr>
<td>B2C1</td>
<td>FRMD04</td>
</tr>
<tr>
<td>001C</td>
<td>FS</td>
</tr>
<tr>
<td>E009</td>
<td>GETCHR</td>
</tr>
<tr>
<td>002A</td>
<td>GETFR0</td>
</tr>
<tr>
<td>B868</td>
<td>GETN02</td>
</tr>
<tr>
<td>0028</td>
<td>GETN0T</td>
</tr>
<tr>
<td>B4EB</td>
<td>GETR0D</td>
</tr>
<tr>
<td>008B</td>
<td>GRPND</td>
</tr>
<tr>
<td>0005</td>
<td>GRTYP</td>
</tr>
<tr>
<td>BACD</td>
<td>GTRD01</td>
</tr>
<tr>
<td>BADA</td>
<td>GTRD04</td>
</tr>
<tr>
<td>BAF8</td>
<td>GTRD07</td>
</tr>
<tr>
<td>B448</td>
<td>GTRN03</td>
</tr>
<tr>
<td>B55B</td>
<td>GTRN06</td>
</tr>
<tr>
<td>0012</td>
<td>HDRLEN</td>
</tr>
<tr>
<td>002E</td>
<td>HIL0</td>
</tr>
<tr>
<td>BF2B</td>
<td>HKAS01</td>
</tr>
<tr>
<td>B240</td>
<td>INITPM</td>
</tr>
<tr>
<td>BCED</td>
<td>INITTG</td>
</tr>
<tr>
<td>BOD8</td>
<td>INLS02</td>
</tr>
<tr>
<td>BCE8</td>
<td>INTG02</td>
</tr>
<tr>
<td>BEE9</td>
<td>KBINPT</td>
</tr>
<tr>
<td>E018</td>
<td>KEYBRD</td>
</tr>
<tr>
<td>0087</td>
<td>KEYMOD</td>
</tr>
<tr>
<td>0EA5</td>
<td>KEYNT3</td>
</tr>
<tr>
<td>AFE9</td>
<td>LDUS01</td>
</tr>
<tr>
<td>B766</td>
<td>LINS01</td>
</tr>
<tr>
<td>BC83</td>
<td>LISTEN</td>
</tr>
<tr>
<td>017E</td>
<td>LOAD01</td>
</tr>
<tr>
<td>AF4F</td>
<td>LOGN02</td>
</tr>
<tr>
<td>01C4X</td>
<td>LOGTBL</td>
</tr>
<tr>
<td>B355</td>
<td>LSTU01</td>
</tr>
<tr>
<td>B3BC</td>
<td>LSTU04</td>
</tr>
<tr>
<td>B343</td>
<td>LSTU07</td>
</tr>
<tr>
<td>B4BE</td>
<td>MDT01</td>
</tr>
<tr>
<td>B4A3</td>
<td>MDTUN</td>
</tr>
<tr>
<td>BF8E</td>
<td>MLTPLY</td>
</tr>
<tr>
<td>E14D</td>
<td>MONIMP</td>
</tr>
<tr>
<td>BE11</td>
<td>MONPRT</td>
</tr>
<tr>
<td>BF8B</td>
<td>MOUNB2</td>
</tr>
<tr>
<td>AD0C</td>
<td>MSGAKS</td>
</tr>
<tr>
<td>A388</td>
<td>MSGFNN</td>
</tr>
<tr>
<td>A37B</td>
<td>MSGITR</td>
</tr>
<tr>
<td>A0F7</td>
<td>MSGLUF</td>
</tr>
<tr>
<td>A968</td>
<td>MSGMUC</td>
</tr>
<tr>
<td>A7C5</td>
<td>MSGNTN</td>
</tr>
<tr>
<td>A7E2</td>
<td>MSGPAK</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>A17E</td>
<td>MSGPPB</td>
</tr>
<tr>
<td>A124</td>
<td>MSGTP</td>
</tr>
<tr>
<td>A36A</td>
<td>MSGSSI</td>
</tr>
<tr>
<td>A55A</td>
<td>MSGSTT</td>
</tr>
<tr>
<td>A7DC</td>
<td>MSGTRI</td>
</tr>
<tr>
<td>ACC1</td>
<td>MSQNY</td>
</tr>
<tr>
<td>A762</td>
<td>MSGATA</td>
</tr>
<tr>
<td>BF95</td>
<td>MULT01</td>
</tr>
<tr>
<td>080F</td>
<td>NENTUN</td>
</tr>
<tr>
<td>006C</td>
<td>NOTBSY</td>
</tr>
<tr>
<td>0080</td>
<td>NUL</td>
</tr>
<tr>
<td>0E08X</td>
<td>NUMPLA</td>
</tr>
<tr>
<td>B4A7</td>
<td>NNT081</td>
</tr>
<tr>
<td>B46F</td>
<td>NXT081</td>
</tr>
<tr>
<td>B43F</td>
<td>NXTP083</td>
</tr>
<tr>
<td>0816</td>
<td>OQANG</td>
</tr>
<tr>
<td>B4F1</td>
<td>PARNOT</td>
</tr>
<tr>
<td>A0B8</td>
<td>PART081</td>
</tr>
<tr>
<td>A023</td>
<td>PART085</td>
</tr>
<tr>
<td>BBE3</td>
<td>PERREC</td>
</tr>
<tr>
<td>B024</td>
<td>PLAREL</td>
</tr>
<tr>
<td>B3A4</td>
<td>PRELUD</td>
</tr>
<tr>
<td>B47B</td>
<td>PRET081</td>
</tr>
<tr>
<td>B447</td>
<td>PRETUN</td>
</tr>
<tr>
<td>B302</td>
<td>PRLD081</td>
</tr>
<tr>
<td>BAB2</td>
<td>PTHDR</td>
</tr>
<tr>
<td>0811</td>
<td>PRTSCR</td>
</tr>
<tr>
<td>0814</td>
<td>PTCNST</td>
</tr>
<tr>
<td>B8F7</td>
<td>QRYKEY</td>
</tr>
<tr>
<td>BA55</td>
<td>QRYTON</td>
</tr>
<tr>
<td>B2B2</td>
<td>QYFR083</td>
</tr>
<tr>
<td>B9FA</td>
<td>QYK801</td>
</tr>
<tr>
<td>B972</td>
<td>QYK804</td>
</tr>
<tr>
<td>BAD8</td>
<td>QYP081</td>
</tr>
<tr>
<td>B88A</td>
<td>QYN081</td>
</tr>
<tr>
<td>B808</td>
<td>QYN084</td>
</tr>
<tr>
<td>BA93</td>
<td>QT082</td>
</tr>
<tr>
<td>BAB7</td>
<td>QT085</td>
</tr>
<tr>
<td>B82B</td>
<td>RANDOM</td>
</tr>
<tr>
<td>B788</td>
<td>RATS01</td>
</tr>
<tr>
<td>0808</td>
<td>RATSCH</td>
</tr>
<tr>
<td>B9A3</td>
<td>RDKY082</td>
</tr>
<tr>
<td>B9B4</td>
<td>RDKY085</td>
</tr>
<tr>
<td>B9CA</td>
<td>RDKY087</td>
</tr>
<tr>
<td>B987</td>
<td>REKEY</td>
</tr>
<tr>
<td>BE1C</td>
<td>RESPON</td>
</tr>
<tr>
<td>A05F</td>
<td>RND0135</td>
</tr>
<tr>
<td>B7C8</td>
<td>RND082</td>
</tr>
</tbody>
</table>

**Note:** The table above contains the symbol table for the SIGHTSINGER’S MIRROR. Each entry includes the symbol, description, hexadecimal value, binary representation, and octal representation. The symbols are used in the context of the documentation or software associated with the SIGHTSINGER’S MIRROR system.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Symbol</th>
<th>Meaning</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B7F2</td>
<td>RNDS05</td>
<td>B7F6</td>
<td>RNDS06</td>
<td>0010</td>
<td>RNDSCH</td>
</tr>
<tr>
<td>B7A8</td>
<td>RNDSEL</td>
<td>000A</td>
<td>RGNTRN</td>
<td>001E</td>
<td>RS</td>
</tr>
<tr>
<td>BE2D</td>
<td>RSPN01</td>
<td>BE4C</td>
<td>RSPN02</td>
<td>BE4F</td>
<td>RSPN03</td>
</tr>
<tr>
<td>C80C</td>
<td>RSTKEY</td>
<td>BE5X</td>
<td>SAVCHMD</td>
<td>000C</td>
<td>SAVETO</td>
</tr>
<tr>
<td>0087</td>
<td>SAVFRM</td>
<td>B804</td>
<td>SCAN01</td>
<td>B80B</td>
<td>SCAN02</td>
</tr>
<tr>
<td>B817</td>
<td>SCAN03</td>
<td>B7FA</td>
<td>SCANTG</td>
<td>0080</td>
<td>SCLET</td>
</tr>
<tr>
<td>0FF5X</td>
<td>SCORE</td>
<td>F080</td>
<td>SCREEN</td>
<td>0A37X</td>
<td>SEED1</td>
</tr>
<tr>
<td>0A3C</td>
<td>SEED2</td>
<td>0006</td>
<td>SELCH</td>
<td>B6E5</td>
<td>SELTUN</td>
</tr>
<tr>
<td>0030</td>
<td>SETCP</td>
<td>0030</td>
<td>SETCP</td>
<td>002E</td>
<td>SETG01</td>
</tr>
<tr>
<td>B104</td>
<td>SETG02</td>
<td>B129</td>
<td>SETG03</td>
<td>B139</td>
<td>SETG04</td>
</tr>
<tr>
<td>B13D</td>
<td>SETG05</td>
<td>B154</td>
<td>SETG06</td>
<td>B158</td>
<td>SETG07</td>
</tr>
<tr>
<td>E631</td>
<td>SETINP</td>
<td>0089</td>
<td>SETKEY</td>
<td>B9A7</td>
<td>SETN01</td>
</tr>
<tr>
<td>0E02</td>
<td>SETNUM</td>
<td>0E26</td>
<td>SETNFS</td>
<td>0E15</td>
<td>SETOUT</td>
</tr>
<tr>
<td>B54C</td>
<td>SETPAR</td>
<td>B80F</td>
<td>SETTG</td>
<td>B870</td>
<td>SETUNE</td>
</tr>
<tr>
<td>080E</td>
<td>SETXPO</td>
<td>B3E0</td>
<td>SGIT01</td>
<td>B3E6</td>
<td>SGIT02</td>
</tr>
<tr>
<td>B3F5</td>
<td>SGIT03</td>
<td>B403</td>
<td>SGIT04</td>
<td>B41C</td>
<td>SGIT05</td>
</tr>
<tr>
<td>B420</td>
<td>SGIT06</td>
<td>B442</td>
<td>SGIT07</td>
<td>B443</td>
<td>SGIT08</td>
</tr>
<tr>
<td>B59D</td>
<td>SHOTUN</td>
<td>B5C5</td>
<td>SHOT01</td>
<td>B5CE</td>
<td>SHOT02</td>
</tr>
<tr>
<td>00F2</td>
<td>SI</td>
<td>B306</td>
<td>SINGIT</td>
<td>B6E8</td>
<td>SLT01</td>
</tr>
<tr>
<td>B6F7</td>
<td>SLTN02</td>
<td>B793</td>
<td>SLTN03</td>
<td>B789</td>
<td>SLTN04</td>
</tr>
<tr>
<td>B73F</td>
<td>SLTN05</td>
<td>B750</td>
<td>SLTN06</td>
<td>00FE</td>
<td>SMLFLT</td>
</tr>
<tr>
<td>00FE</td>
<td>SNGBAR</td>
<td>0025</td>
<td>SNGTUN</td>
<td>000E</td>
<td>SO</td>
</tr>
<tr>
<td>0001</td>
<td>SOH</td>
<td>0020</td>
<td>SPACE</td>
<td>AF61</td>
<td>SPCFL</td>
</tr>
<tr>
<td>A82D</td>
<td>SPCHAR</td>
<td>AF65</td>
<td>SPFI01</td>
<td>AF6D</td>
<td>SPFI02</td>
</tr>
<tr>
<td>AF6F</td>
<td>SPF013</td>
<td>AF73</td>
<td>SPF04</td>
<td>AFA1</td>
<td>SSM</td>
</tr>
<tr>
<td>AEB6</td>
<td>SSM01</td>
<td>AEC1</td>
<td>SSM02</td>
<td>AEDB</td>
<td>SSM03</td>
</tr>
<tr>
<td>AEE8</td>
<td>SSM04</td>
<td>0005</td>
<td>STAFF</td>
<td>B096</td>
<td>STATM</td>
</tr>
<tr>
<td>BE70</td>
<td>STCR01</td>
<td>0004</td>
<td>STFCFL</td>
<td>B093</td>
<td>STM01</td>
</tr>
<tr>
<td>0002</td>
<td>STX</td>
<td>001A</td>
<td>SUB</td>
<td>0016</td>
<td>SYN</td>
</tr>
<tr>
<td>E1A2</td>
<td>SYNC</td>
<td>E7B0</td>
<td>TAPL0D</td>
<td>E383</td>
<td>TAPSAM</td>
</tr>
<tr>
<td>112X</td>
<td>TEMPO1</td>
<td>110D</td>
<td>TEMPO2</td>
<td>0772X</td>
<td>TEMPC</td>
</tr>
<tr>
<td>077X</td>
<td>TEMPS0</td>
<td>0020</td>
<td>TGHLEN</td>
<td>00BD</td>
<td>TIMSIG</td>
</tr>
<tr>
<td>10FCX</td>
<td>TMRMSG</td>
<td>BCA8</td>
<td>TONOTE</td>
<td>001X</td>
<td>TOPRAM</td>
</tr>
<tr>
<td>B5D3</td>
<td>TRNST</td>
<td>001B</td>
<td>TNPOL</td>
<td>B5E4</td>
<td>TRN01</td>
</tr>
<tr>
<td>B6E3</td>
<td>TRNT02</td>
<td>B61B</td>
<td>TRNT03</td>
<td>B620</td>
<td>TRNT04</td>
</tr>
<tr>
<td>B621</td>
<td>TRNT05</td>
<td>0004</td>
<td>TUNCLF</td>
<td>0B2X</td>
<td>TUNCT</td>
</tr>
<tr>
<td>0002</td>
<td>TUNED2</td>
<td>0DBF</td>
<td>TUNGRP</td>
<td>0A9X</td>
<td>TUNHI</td>
</tr>
<tr>
<td>0000</td>
<td>TUNLEN</td>
<td>0A3X</td>
<td>TUNLON</td>
<td>BBCB</td>
<td>TUNREC</td>
</tr>
<tr>
<td>0A0X</td>
<td>TUNRNG</td>
<td>0093</td>
<td>TUNRTY</td>
<td>000B</td>
<td>TYPGRP</td>
</tr>
<tr>
<td>000B</td>
<td>TYPLET</td>
<td>0020</td>
<td>TYPPTER</td>
<td>04A0</td>
<td>TYPSC</td>
</tr>
<tr>
<td>081B</td>
<td>TYPUNT</td>
<td>0000</td>
<td>TYPUSR</td>
<td>0811</td>
<td>UNITNO</td>
</tr>
<tr>
<td>001F</td>
<td>US</td>
<td>0020</td>
<td>USEDBT</td>
<td>BB86</td>
<td>USEREC</td>
</tr>
<tr>
<td>0D9AX</td>
<td>USRANG</td>
<td>BBAC</td>
<td>USRANG</td>
<td>022X</td>
<td>USRHT</td>
</tr>
<tr>
<td>0AEX</td>
<td>USRLow</td>
<td>0E34X</td>
<td>USRMAD</td>
<td>0E4X</td>
<td>USRAR</td>
</tr>
<tr>
<td>0D1B</td>
<td>VIDEO</td>
<td>000B</td>
<td>VT</td>
<td>BEA3</td>
<td>WALT</td>
</tr>
<tr>
<td>AE0D</td>
<td>WARM</td>
<td>0001</td>
<td>WHTNOT</td>
<td>0CFX</td>
<td>WHTSVC</td>
</tr>
<tr>
<td>BC57</td>
<td>WRT01</td>
<td>BC78</td>
<td>WRT02</td>
<td>BC19</td>
<td>WRTB0F</td>
</tr>
<tr>
<td>0018</td>
<td>XLATIN</td>
<td>0008</td>
<td>XPOSFL</td>
<td>000B</td>
<td>XRTB0F</td>
</tr>
</tbody>
</table>

No Fatal error(s)
APPENDIX C

MISCELLANEOUS FORMS
To: L.S.U. Committee on the Use of Humans and Animals as Research Subjects, Dr. W. Sheldon Biven School of Veterinary Medicine Chairman

From: Randall M. Kolb, Ph.D. candidate, School of Music

Title of Research Project: A REAL-TIME MICROCOMPUTER-ASSISTED SYSTEM FOR TRANSLATING AURAL, MONOPHONIC TONES INTO MUSIC NOTATION AS AN AID IN SIGHTSING

This experiment will take place in the Fall semester of 1983, beginning at mid-term and continuing for a period of eight weeks to the end of the semester. The subjects for this experiment will be randomly selected from each of the first semester music theory classes (Music 1701) being taught. Each of the students selected will then be given an opportunity to withdraw. The total number of students in the experimental group will be approximately thirty-five. The control group will consist of all remaining students in the theory classes.

The members of the experimental group will be given access to a computer for sightsinging practice in the ear training laboratory during the eight weeks of the experiment. These subjects will be asked to practice a minimum of thirty minutes twice a week. Additional practice will be encouraged. No demands beyond the normal requirements of the music theory course will be imposed on the members of the control group.

In order to evaluate the effect of the training on the members of the experimental group, it will be necessary that the grades on the mid-term and final sightsinging examinations be made available to this investigator for analysis. All such information will be held in strict confidence, and the anonymity of each subject will be protected.

Investigator's Name
NAME: Randall M. Kolb  DATE: November 15, 1983

DEPT: Music

SUBJECT: Request for research approval

TITLE OF RESEARCH PROJECT: A Real-Time Microcomputer-Assisted System for Translating Aural, Monophonic Tones into Music Notation as an Aid in Sightsinging

The investigator gives assurances to the Committee on the Use of Humans and Animals as Research Subjects for each of the following:

Yes    No
1. The human subjects are volunteers.       X

2. Subjects have the freedom to withdraw at any time. X

3. That the data collected will not be used for any purpose not approved by the subjects. X

4. The subjects are guaranteed anonymity. X

5. The subjects will be informed beforehand as to the nature of their activity. X

6. The nature of the activity will not cause any physical or psychological harm to the subjects. X

7. Individual performances will not be disclosed to persons other than those involved in the research, those authorized by the subject. X

8. If minors are to participate in this experiment, valid consent has been obtained from the parents or guardian. X

9. That all questions have been answered to the subject's satisfaction. X

10. All volunteers will consent by signature. X

Any exceptions or qualifications to the above assurances are explained below:

__________________________________________

Investigator's Name
TENTATIVE PROPOSAL REVIEW:

FROM: Committee on the Use of Humans and Animals as Research Subjects

TO:

RE: Proposal of Randall M. Kolb _______received 11-16-83____

Entitled ________A Real-Time Microcomputer-Assisted System for__________

Translating Aural, Monophonic Tones into Music Notation as an__________

Aid in Sightsinging______________________________

This is to certify that the Chairman of the Committee on the Use of Humans and Animals as Research Subjects has reviewed the above proposal. The Chairman evaluated the procedures of the proposal with appropriate guidelines established for activities supported by federal funds involving as subjects humans and/or animals.

Recommendation of Chairman _______Approved__________

Comments: This proposal was originally approved on 11-16-83 and the form was lost. This form only confirms the previous action.

A review of this proposal by the Committee will be accomplished at the next monthly meeting and you will be notified of the committee's recommendation.

Date 4-9-84 _______ /s/ W. S. Bivin

W. S. Bivin, Chairman
**TO BE RETAINED BY THE INVESTIGATOR:**

**EXPERIMENT SIGN-UP FORM**

My signature, on this sheet, by which I volunteer to participate in the experiment on **The Efficacy of a Computer-Assisted Instructional System for Sightsinging Students** conducted by

Randall M. Kolb
Experimenter

indicates that I understand that all subjects in the project are volunteers, that I can withdraw at any time from the experiment, that I have been or will be informed as to the nature of the experiment, that the data I provide will be anonymous and my identity will not be revealed without my permission, and that my performance in this experiment may be used for additional approved projects. Finally, I shall be given an opportunity to ask questions prior to the start of the experiment and after my participation is complete.

Subject's signature
VITA

Randall Martin Kolb was born in Batesville, Mississippi on November 26, 1953 to Joanna and Paul W. Kolb. He graduated from Louisiana State University in Baton Rouge in 1975 and 1977 with a Bachelor of Music and Master of Music degree in piano performance.

In 1977 he began his Doctor of Philosophy degree in music theory with a minor in computer science while serving as a graduate teaching assistant in undergraduate music theory. After four years of full-time work toward the degree in this capacity he was employed in a full-time position as a computer programmer for the Louisiana State Department of Transportation and Development. He has continued to hold this position to 1984 while working to finish the doctorate.
Candidate:  Randall Kolb

Major Field:  Music Education

Title of Thesis:  A Real-Time Microcomputer-Assisted System for Translating Aural Monophonic Tones to Music Notation as an Aid for Sightsinging

Approved:

[Signatures]

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

[Signatures]

Date of Examination:

April 5, 1984