





JUL 5 '77 #172

M 331

JUN 188

FEB 15 1987

9

JUN 18 1987

ABSTRACT

THE DEVELOPMENT OF THE METHODS FOR ANALYSIS OF MUSICAL COMPOSITIONS AND FOR THE FORMATION OF A SYMMETRICAL TWELVE-TONE ROW USING THE ELECTRONIC DIGITAL COMPUTER

by Gilbert Harvey Roller

This research was conducted in order to develop the methods for the analysis of musical compositions and for the formation of a symmetrical twelve-tone row, using the electronic digital computer. The analysis method used is applicable for music of all historical periods. The use of the computer affords the advantages of speed and of printed results in a clear, tabulated form.

The analyses were obtained by coding the musical notes into numbers and forming a set of instructions (a program) for the electronic digital computer, which manipulated the numbers to produce the desired information. Analyses were made of musical selections of from four to twenty-two voices, and of composers from Mozart to Stravinsky. The results of the analyses contain a printed table which gives for each chord analyzed its measure number and its metrical position within the measure, presents a list of the intervals as measured from the lowest sounding note, and prints the letter name of the lowest sounding note, the octave placement of each chord tone, and the instrumentation of each chord. The analyses include a

catalog of all repeated chords and repeated chord progressions. The root of each chord, as determined by the Hindemith method, is also listed. The frequency of repetition of the melodic intervals in each voice was found, along with the frequency of intervals between successive roots. These are listed, with indications as to the upward and downward movements.

On file with the librarian at the Computer Center of Michigan State University are the programs for analyzing four-part music, together with accompanying card decks and instructions for the use of the programs.

The analyses were preceded by a short history of musical analysis and a brief description of the electronic digital computer.

A Control Data Corporation 160A Computer was used as the vehicle for solution of the problems in this study, but all the programs developed herein are adaptable to the large 3600 Computer.

An application of computer logic to a creative problem in music is described in the solution of a symmetrical twelve-tone row problem. Using the methods developed in the programs of this study all 55,440 possible symmetrical twelve-tone rows could be located. One row was processed in this research to show the development of the method.

In a search for the most satisfactory method of musical analysis there are two qualities which are obviously desirable: maximal speed and universal applicability. An analysis of

intervals with Arabic numerals may be applied to the vertical sonorities of music of all periods. Since the outstanding characteristic of the electronic, digital computer is its ability to compute Arabic numerals at a high rate of speed and with near-absolute accuracy, the major conclusion of this study is that musical analysis may be most easily, quickly and accurately done on a high-speed electronic digital computer. In addition, the electronic computer codifies the results in a readable and meaningful form.

The high speed of the digital computer enables it to solve with ease problems in selection. For this reason it is to be further concluded that creative problems in music which involve many choices and rejections afford an opportunity for the use of the computer.

There have been relatively few applications of the computer to music up to the present time, but with the rapid development of simpler programming techniques and the present accessibility of these machines, there will doubtless be an increased use of computers in the music field.

THE DEVELOPMENT OF THE METHODS
FOR ANALYSIS OF MUSICAL COMPOSITIONS
AND FOR THE FORMATION OF A SYMMETRICAL TWELVE-TONE ROW
USING THE ELECTRONIC DIGITAL COMPUTER

By

Gilbert Harvey Roller

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Music

1963

ACKNOWLEDGMENTS

Appreciation is expressed to the administrators in the Department of Music and to those of the Computer Center for their cooperation and assistance in this research.

The guidance and counsel of Dr. William R. Sur, Chairman of the doctoral committee, has been invaluable. Appreciation is expressed to Dr. Robert Sidnell and Dr. Merrell Sherburn for their interest and suggestions in preparing this paper. Dr. Paul Harder provided the impetus and concern for the twelve-tone row problem.

Further gratitude is expressed to Dr. Edgar Kirk, the dissertation adviser, whose warm counsel and time were given unselfishly. His encouragement and enthusiasm for the study contributed largely to its final completion.

I am grateful to Dr. David L. McKenna, President of Spring Arbor (Michigan) College, for the wisdom he shared at several strategic points.

Finally, this study is dedicated to Jeri, my wife, whose typing skills are commensurate with her efficiency as the mother of our four children, whose loyalty and love made times of stress and strain surmountable, and whose unswerving devotion gave me the courage to undertake and to complete the work.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	ii
LIST OF FIGURES.....	v
LIST OF CHARTS.....	vii
 Chapter	 Page
I. INTRODUCTION.....	1
II. DEVELOPMENT OF COMPUTER PROGRAMS FOR MUSICAL ANALYSIS.....	15
Coding Music into Numbers.....	25
III. RESULTS OF ANALYSIS ARRIVED AT THROUGH APPLICATION OF THE PROGRAMS TO MUSIC OF VARIOUS STYLES.....	35
Beethoven: <u>Quartet, Opus 131</u>	36
Five Frequency Tables from Program Ten Results.....	47
Hindemith: <u>Quintet, Opus 24, No. 2</u>	48
Program Five Results.....	49
Webern: <u>Quartet, Opus 9</u>	50
Program Seven Results.....	52
Mozart: <u>Symphony No. 40</u>	52
Root Movement Analyses.....	53
Stravinsky: <u>Symphony in Three Movements</u>	54
IV. DATA PREPARATION AND USE OF THE ANALYSIS PROGRAMS....	56
V. AN APPLICATION OF COMPUTER LOGIC TO A CREATIVE PROBLEM IN MUSIC.....	69
VI. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS.....	73
VII. COMPUTER PROGRAMS WITH THEIR RESULTS.....	77
Beethoven-- <u>String Quartet in C# Minor, Opus 131</u>	77
Hindemith-- <u>Kleine Kammermusik, Opus 24, No. 2</u> (Quintet).....	175
Mozart-- <u>Symphony in G Minor</u>	236

TABLE OF CONTENTS (Continued)

Chapter	Page
Webern-- <u>Six Bagatellen for String Quartet, Opus 9..</u>	295
Schumann-- <u>Papillons, Opus 2.....</u>	342
Stravinsky-- <u>Symphony in Three Movements.....</u>	376
Symmetrical Twelve-Tone Row.....	440
 BIBLIOGRAPHY.....	 472

LIST OF FIGURES

Figure	Page
1. Block Diagram of a Modern Digital Computer.....	4
2. Storage of a Two-Dimensional Array.....	24
3. Conversion of Notes to Numbers.....	24
4. Chart for Indicating First Digit of Coded Note.....	25
5. Coding Example.....	25
6. Coding Table for Interval Values.....	28
7. Information from Program Five.....	32
8. Hindemith's Table for Choice of Interval Value and Chord Root.....	34
9. Interval Interpretation.....	40
10. Table for De-coding Numbers into Notes.....	45
11. Coding System for INFO.....	57
12. Classification of Instruments (Quartet).....	59
13. Form for Coding INFO and Instruments.....	59
14. Master Plan for Data Movement.....	60
15. Classification of Instruments (Quintet).....	60
16. Form for Coding INFO and Instruments.....	61
17. Master Plan for Data Movement.....	61
18. Classification of Instruments (Webern).....	62
19. Form for Coding an Irregular INFO and Instruments.....	62
20. Master Plan for Data Movement.....	63
21. Classification of Instruments (Symphony).....	63

LIST OF FIGURES (Continued)

Figure	Page
22. Form for Coding INFO.....	64
23. Form for Coding Instruments.....	64
24. Master Plan for Data Movement.....	64
25. Classification of Instruments (Stravinsky).....	65
26. Form for Coding INFO.....	66
27. Form for Coding Instruments.....	66
28. Master Plan for Data Movement.....	66
29. Classification of Instruments (Schumann).....	67
30. Form for Coding INFO.....	67
31. Form for Coding the Piano Part.....	68
32. Master Plan for Data Movement.....	68
33. Order of Numbers.....	71
34. Substitution of Numbers for Letters.....	72

LIST OF CHARTS

Chart	Page
1. Harmonic Intervals of Program Five.....	39
2. Results Shown in Program Six.....	41
3. Results of Program Seven.....	44
4. Frequency of Root Occurrence.....	45
5. Comparison of Hindemith Roots, Lowest Sounding Notes, and Traditional Roots.....	46
6. Frequency of Melodic Intervals.....	47
7. Harmonic Intervals of Program Five.....	49
8. Results of Program Four.....	51
9. Results of Program OMITT.....	51
10. De-coding of Program Seven Results.....	52
11. Graph Showing Root Movement.....	53
12. Analyzed Chords from Program Five Results.....	55

CHAPTER I

INTRODUCTION

Musicians have always recognized the importance of continual research in theory and analysis as a necessary basis for intelligent interpretation in performance and for the development of effective new trends in composition. The music educator is further interested in a by-product of analysis--form--for it is one of the useful tools in teaching music literature. Music analysis is a slow, laborious task, and many systems have been developed in an effort to make the methods more meaningful and less time consuming.

At the beginning of the seventeenth century Peri, Caccini and Cavalieri were the earliest proponents of the use of figures as a method of indicating harmony. Jean Philippe Rameau (1683-1764), who "was both the founder of the theory of harmony in the modern sense of the term, and the most important French composer of the 18th century,"¹ was another early contributor to harmonic analysis. He was the first to recognize the inversion of chords. "This epoch-making doctrine [the doctrine of fundamental chords]" established the fact that all

¹Oliver Strunk, Source Readings in Music History (New York: W. W. Norton and Company, 1950), p. 564.

possible harmonies can be reduced to a limited number of fundamental forms (accords fondamentaux) called "tonic," "dominant," and "subdominant..."¹

After the time of Rameau, theorists began to analyze chords with Roman numerals. Since 1600 many variations of analysis, using symbols, signs, Arabic and Roman numerals, have been developed. Other important musicians who have contributed to analysis methods are Gioseffo Zarlino, Hugo Riemann, Heinrich Schenker, Joseph Schillinger, and Paul Hindemith.

The method of harmonic analysis most used today combines Roman and Arabic numerals. However, neither this method, nor any of the older systems, is adequate for the analysis of contemporary music. It is also true that there is no universally accepted means for analyzing any non-tertian harmony of the modern period. In a search for a method of analysis that would combine the advantage of universal applicability to music of all periods with the added benefit of speed, the electronic, digital computer was investigated as a possible instrument of analysis. Since the outstanding characteristic of the computer is its ability to compute Arabic numerals at a high rate of speed with near-absolute accuracy, and since an analysis of any given piece of music may be made with Arabic numerals, it would seem that the computer would, indeed, be the ideal, logical, and obvious choice of a method for musical analysis. The first problem of this study is to develop such a method of musical analysis using the electronic computer.

¹Paul Henry Lang, Music in Western Civilization (New York: W. W. Norton and Company, 1941), p. 546.

The second problem of this research is to provide a basis for designing a symmetrical twelve-tone row using the electronic computer. This related problem, with its method of solution, is presented in Chapter V.

A cursory knowledge of the computer and its peripheral machines is necessary for the understanding of both the problems discussed in this paper. A computer can do only what it is instructed to do--nothing more. To "set up a program" to be run through a computer means to provide a body of data and to devise a set of instructions for its use. The instructions must be very explicit, the logic of the problem correct, and the program entirely free from errors in its mechanics. After a given program has been developed other sets of data may be substituted and processed. In other words, the program is basic; the data, interchangeable.

While it is true that modern computers exhibit a variety of different logical designs, yet they do have a common general structure, as can be seen in the following description.

The operations of the individual parts of a computer are very closely intermeshed, and it is difficult to draw a clear line between them. It may nevertheless be helpful to outline a functional block diagram of a modern computer. Such a diagram is shown below. It should be understood that the diagram does not indicate the layout of actual physical building blocks....The function of the various units can then be stated as follows:

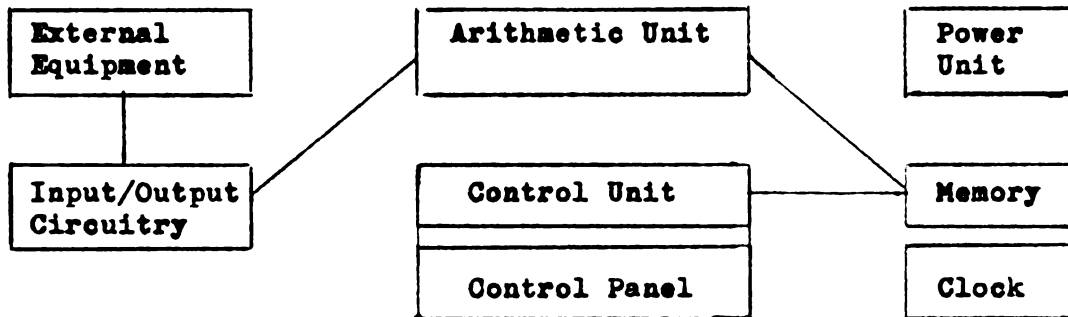


Figure 1. "Block diagram of a modern digital computer. The connecting lines are the main paths of information flow."

Arithmetic Unit: Execution of all machine instructions, except a few which are directly concerning external equipment. The unit has a few special small storage devices, called registers.... Information transferred from one place to another usually passes through the arithmetic unit.

Control Unit: The control unit is the nerve center of the computer. It controls all events for proper sequence. There are special registers for the necessary logical information. Each instruction is analyzed in the control unit, and the necessary operations are initiated accordingly. A special function of the control unit is the automatic checking of computer malfunctions and of certain coding errors. Alarm signals are given and sometimes the machine is stopped if such conditions are found.

Control Panel: All switches and other manual control elements necessary to operate the computer are assembled on the control panel. The registers of the arithmetic unit and of the control unit are connected to indicators on the panel, so that their contents may be visually inspected. The state of the various external devices, like ready, not ready, busy, etc., may be displayed. Certain lights serve to show alarm conditions...

Clock: The clock is a device which generates a continuous sequence of pulses as soon as the start button is depressed. It governs the basic speed of all operations...

Memory: The memory is the device which can store large quantities of information. It takes over the functions of two different devices of

earlier computers. First, it stores intermediate results for short periods of time....Secondly, it stores the program and the data....The memory is divided into a number of cells, or registers, each with an identifying number, called the address.... A distinction is sometimes made between internal and external memory. An external memory permits the physical removal of the storage medium from the computer, e.g., magnetic tapes.

Input-Output Circuitry: The timing of the external devices is determined by mechanical properties, such as the speed of a motor or the response time of a relay. Since these factors cannot be controlled to sufficiently close tolerances, the external devices cannot be synchronized with the clock pulse rate of the computer. The input-output circuitry serves as information buffer.

Power Unit: The power unit contains all the circuitry needed to generate and¹regulate all the voltages needed in the computer.

External Equipment: One of the peripheral machines (external equipment) is the key-punch machine. It prepares the IBM cards, punching only one line of numbers, letters, symbols, or combinations of these, on each card. It has a duplicating key for the convenience of the operator. A person desiring to use the computer types his program (set of instructions) on the key-punch machine, which has a keyboard somewhat similar to that of a typewriter. A card reader prints out on paper the information read from the card in as many lines as there are cards. This machine is of great convenience for use in correcting or altering the program. Other peripheral machines are a printer, two magnetic tape units, and a plotter.

¹Paul Von Handel (ed.), Electronic Computers (Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1961), p. 19.

In regard to the memory unit, described on pages four and five, it should be noted that programs are necessarily limited by the number of spaces in the memory unit. The 160-A, which was used for the programs in this study, has about 4,000 memory spaces. The new 3600 Computer, which has been installed at Michigan State University, has a memory storage of almost 32,000 items. The programs in this study were developed before the installation of the 3600 Computer.

What has been said thus far in this study, and particularly in the pages immediately preceding, has referred to the digital computer. Another widely used type of electronic computer is called the analog computer.

The difference between these two kinds of machine may best be explained by a reference to their names. The word digital refers to a system of digits (a single number-symbol), a definite quantity. Numbers such as 1, 2, 3, etc., are digits. The addition of $4 + 2 = 6$ is a digital computation. The digital computer uses numbers as definite quantities. For instance, any operation that is performed on an adding machine is a digital one. The abacus is another familiar example of a digital computer.

Analog (sometimes analogue) computers indicate quantities or numbers by reference to something else, inches along a scale--such as the slide rule--angle of rotation, or some other set standard. Fahnestock uses the illustration of a gallon of gasoline. When one speaks of this, he refers to a quantity of volume equal to that of a standard gallon as maintained by the

National Bureau of Standards in Washington, D. C.¹ The kind of quantity identification for which quantities (or numbers) are given with reference to a known standard is called analog. The numbers are analogous to what they represent; hence, the name. The analog computer often produces its results in the form of graphs. Of the two types of automatic computers, the digital is the more widely used, and for the purposes of this paper, the more appropriate.

Many authorities have made evaluations of the computer and its abilities. James Fahnestock, in Computers and How They Work, expresses the consensus of these evaluations:

It works at fantastic speeds, with virtually unlimited accuracy, and with almost infallible reliability.²

Today's large-scale computers help to make airplanes faster and safer. They help build automobiles. Guided missiles and space vehicles owe their very existence to electronic computers. Gigantic payrolls are prepared by machines, computers forecast weather, design ships, control industrial processes, run machine tools, and solve tremendously complex business problems for private and government organizations.³

Since this machine is so valuable in the world of industry, business, and government, the question naturally arises: What significance, if any, does it have with respect to the arts? In a June, 1961, article in Computers and Automation over 500 areas of computer applications were listed,

¹James Fahnestock, Computers and How They Work (New York: Ziff-Davis Publishing Company, 1959), p. 24.

²Ibid., p. 7.

³Ibid., p. 2.

none of which was closely connected with the humanities except for linguistics. There were four fringe applications: two to education and two concerned with retrieval of information in libraries.¹ This preponderance of use of computers in the industrial world as compared with their use in the arts is typical, at least up to the present time; however, computers have been employed somewhat in the arts, as the remainder of this chapter will demonstrate.

In the field of linguistics there is considerable research being carried on in the translation of languages. Franz L. Alt in his book, Advances in Computers, discusses two alternatives in this area. He feels that fully automatic, high quality translation (FAHQT), is not a reasonable aim. Either partly mechanized, high quality translation or fully mechanized, low quality translation is a more realistic goal.²

In the article, "Human Translation and Translation by Machine," the thinking of Glasefield, Perschke, and Samet is summarized as follows: To obtain FAHQT, machine translation operational dictionaries would be required for every language to cover words of both single and composite meaning; but the machine still would not be able to follow the train of thought or consider implications unless there is incorporated into it all

¹Neil MacDonald, "Over 500 Areas of Application of Computers," Computers and Automation, X (June, 1961), pp. 133-37.

²Franz L. Alt, Advances in Computers (New York: Academic Press, 1960), pp. 94ff.

generally known and therefore not explicitly formulated elements by means of a network of classifications.¹

In spite of the difficulties involved, some groups are still pursuing FAHQT. Others are making considerable progress following the other aforementioned alternatives.

Franz L. Alt concludes:

An existing program at the Harvard Computation Laboratory can produce word-by-word Russian to English translations at a sustained rate of about 17 words per minute on a UNIVAC I, and about 25 words per minute on a UNIVAC II. This is four to six times more than an expert human translator can produce, but since UNIVAC II time is 100 times more expensive than a human translator's time, commercial machine translation is out of the question at present.²

A computer program which is in progress in the field of literature involves counting word frequency. It is claimed that disputed authorships may be determined by this method.

Braille translations have been produced by a computer. It can translate a 300-page book into Braille in an hour. This would take six days for a skilled translator. The English texts are punched out on cards. The Braille characters emerge from the computer in coded symbols on punched cards which in turn are fed into a printer that prints Braille characters and English texts.

¹E. Von Glasefield, S. Perschke, and E. Samet, "Human Translation and Translation by Machine," Engineer, CCXII (September, 1961), pp. 442-443.

²Alt, p. 112.

It is edited, corrected, and the cards are then fed to an embossing machine which creates plates for the printing machine.¹

It is interesting to note that R. W. Revans in his Preface to D. N. Chorafas' book, Programming Systems for Electronic Computers, makes the following comparison:

When the Authorized Version of the English Bible was translated from its original texts the task took fifty-four men nearly three years. The men were among the greatest scholars of the age; one of them for example, knew fifteen different languages. A couple of years ago a computer translated the same book into Braille in about four hours.²

An art work was created by Ebram Ariazi with an electronic beam as the brush, an oscilloscope as the canvas, and an electronic computer as the painter. This was done at Massachusetts Institute of Technology in a course, Art for Engineers. "He [Ebram Ariazi] hopes to have the computer prepare a tape to run an automatic milling machine so that the computer can make a three-dimensional shape as an impression of what he has seen."³ He also intends to produce art works in color.

It is quite common for the computer to be used as a musical instrument. This involves the rendition of familiar airs in a single-note fashion.

¹"Braille Translations," Computers and Automation, October, 1959, p. 200.

²R. W. Revans, Preface to Programming Systems for Electronic Computers, by D. N. Chorafas (London: Butterworths, 1962), p. ix.

³"Art for Computers," Computers and Automation, January, 1963, p. 8.

Usually producing an oboe-like sound, the output of the computer is easily identified as to the musical number it is rendering. In a similar manner computers have been programmed to generate the sounds which constitute one part of a trio, the other two parts of which are rendered by human musicians or in some cases by the computer itself.¹

Phyllis Higgins says the notes which the computer is capable of producing are B₀, C, D, E, F₇, G, A, B_b, B, C', C#', D', E', and F'.²

Neil MacDonald suggests that orchestral instruments be given automatic control; that is, that the sounds produced by each instrument be controlled by an electronic computer. The timing of all the musical notes and rests could be governed, down to milliseconds, or perhaps even finer, if this is desirable. To offset the criticism of mechanization, he includes in the program the skill of a great conductor who can vary the meter and the relative loudness and softness of the different instruments. He believes this would be "marvelously beautiful" symphonic music.³

One of the most recent experimental approaches to music has been the use of an automatic high-speed digital computer in actual composition. In contrast with electronic music and musique concrète, where new sound media are created, the computer composes pieces

¹John A. Postley, Computers and People (New York: McGraw-Hill Book Company, 1960), p. 224.

²Phyllis Higgins, "Three-Part Music with a Computer as One Part," Computers and Automation, VII (March, 1958), p. 8.

³Neil MacDonald, "Music by Automatic Computers," Computers and Automation, VII (March, 1958), p. 8.

that can be played on conventional instruments; a string quartet called the Illiac Suite¹, the result of such a process, has been published.

A book by Lejaren Hiller, a musician, and Leonard M. Isaacson, a mathematician, Composition with an Electronic Computer, describes the steps and experiments that led to the composing of the Illiac Suite.

In Experiment I, single line melodies, and two- and four-part writing were done by the machine and controlled through a selection of first species counterpoint rules. After a beginning note had been given, the computer selected the remaining notes according to the rules stored in its memory.

All the rules for first species counterpoint were added in Experiment II, which produced four-part music of fair quality except for some monotony in rhythm.

Rhythm, dynamics and chromatic writing were dealt with in Experiment III. With 4/8 time as the meter and the eighth notes as the smallest rhythmic unit, all the possible rhythmic combinations were coded and used in producing music which provided more interest in this area. Entirely random chromatic melodies were composed by the computer in this experiment, to contrast with those on which the strict sixteenth century counterpoint rules were imposed. This included tone rows, which were printed with their inversion, retrograde, and retrograde inversion.

¹Peter S. Hansen, An Introduction to Twentieth Century Music (Boston: Allyn and Bacon, Inc. 1961), p. 354.

In Experiment IV the "objective was the synthesis of music from purely mathematical rules."¹ The melodic intervals for this experiment were selected from a table of probabilities. The computer was programmed to select intervals between parts according to the previous interval. The sound of the music produced was similar to some of the atonal music of contemporary composers. These four experiments prepared the way for the composing of the Illiac Suite.

Peter Hansen, Head of the Department of Music at Tulane University, explains how a computer can write music in the style of earlier composers:

A computing machine cannot "think," but it can be instructed to make selections and rejections. In one sense, the process of composition is the making of selections and rejections, whether consciously or unconsciously. The composer chooses key, meter, timbre, and type of harmony (not necessarily in that order) and once started, chooses from many possibilities what should follow. The number of possibilities are not limitless, but are determined by the general style in which he is writing. In the Bach style, for instance, the exact probability of a tonic chord following a dominant has been determined, and McHose can write, "In their music the composers of the 18th century use the normal progression about seventy-six percent of the time." Bach of course knew nothing about this; he simply wrote music. Nevertheless on the basis of his music the "Bach style" can be defined accurately. The same is true of any other style.

Coded instructions can be given to a computer so that it will choose a "possible," i.e. not incorrect, series of musical material. It can even be instructed to reject "impossible" (in terms of a given style) progressions. Therefore, if a computer is fed instructions based on probabilities

¹Lejaren Hiller, "Computer Music," Scientific American, CCI (December, 1959), p. 119.

determined by a given style of music, it can turn out endless "compositions" in the form of a code which can be easily transcribed into musical notation and performed. At this point human sensitivity enters. Through operation of the laws of chance, some of these "compositions" might be "better" than others. The critical judgment of the trained musician can then accept some of the results and reject others, just as a composer continually accepts or rejects his own ideas when he composes.

In summary, then, the applications of the computer to the areas of the fine arts have been limited, with language translation holding the most prominent place. The one application of the computer to painting is a mechanical one. To date, the most common application of the computer to music is to use the computer as a musical instrument. The only serious applications of the computer to music are those of Hiller, who uses an automatic high speed digital computer for actual composition, and Hansen, who suggests how a computer can write music imitative of another composer.

Within the range of this study, it is my hope to show how an automatic, digital computer can solve the problem of analyzing existing works of any period at a much-increased rate of speed and accuracy over present methods of analysis.

¹Hansen, pp. 354-355.

CHAPTER II

DEVELOPMENT OF COMPUTER PROGRAMS FOR MUSICAL ANALYSIS

The machine for which the programs included in this research were designed is a Control Data 160-A Computer. A stored-program digital computer, it uses a basic, language-type set of instructions written in the Fortran programming language. This means that a problem delineated in prose can be translated into mathematical symbols and words which the machine can understand. The Fortran language is composed of symbols, numbers, and specified words. In automatic programming the computer does a further job of translating by changing the Fortran into binary numbers (0's and 1's), and after that it makes the computations. Before the days of automatic programming the programmer himself had to furnish all data and instructions in binary numbers. It can be seen, then, that programming actually involves two steps: formulating the problem, and translating it into a Fortran language.

A great advancement, which has made it possible for the computer to be used by those with a very limited knowledge of its intricate workings, is the development of this formula-translating language, with the trade name, Fortran.

The Control Data Fortran System provides for the automatic coding of programs for the 160-A computer. Fortran derived from FORMula TRANslation is a language that greatly resembles the language of ordinary mathematics. Computers need elaborate and complex sequences

of instructions, that is, machine language programs to specify every step of their operation. By means of the Fortran system, the programmer is not required to prepare such machine language programs. Instead, he prepares a source program in the more familiar Fortran language. A compiler program then translates the source program into an object program ready to be run.

Fortran permits the programmer to communicate with the computer in a language more concise and familiar than machine language. This yields a saving in the time to train a programmer and in the time taken to write programs. In addition the correcting of errors is accomplished more rapidly.¹

Simple Fortran words, such as IF, when used in a program, are translated through a stored library of instructions, and a whole series of operations are accomplished. The other formula words used in this research are DIMENSION, FORMAT, READ, DO, GO TO, CONTINUE, PRINT, and PUNCH.

A DIMENSION statement serves to allocate storage for arrays.² The word DIMENSION is followed by directions as to how much of the memory space is to be utilized, and under what names the data is to be stored.

FORMAT precedes the description of the data's position on the punched cards, or determines in what manner the alphanumeric (letters and numbers) characters in the results are to be positioned on the printed page or punched cards.

READ causes data cards to be read into the computer's memory in the manner and in the quantity indicated.

¹Control Data Corporation, Fortran System (Minneapolis: Control Data Corporation, 1961), p. iv.

²For an explanation of arrays, see page twenty-three.

"A DO statement is a command to repeat a sequence of Fortran statements altering an index variable each time, until a threshold¹ is reached."² DO instructs the machine to do over and over a certain operation(s) a stated number of times.

An IF statement provides for conditional branching to other parts of the program. It is a two- or three-way decision point. The program may develop in any one of two or three directions, depending upon the results obtained from this expression.

GO TO alters the normal order of the program. Control normally moves from one statement to the statement immediately following. The GO TO provides a transfer to another section of the program.

CONTINUE is a dummy word which acts as the last statement of a DO loop. It transfers control back to the beginning if the sequence is not finished, but instructs the computer to proceed with the subsequent statements if the loop has completed the required number of sequences.

PRINT and PUNCH statements call for the answers to be typewritten or punched on IBM cards. The punch statement is useful in a series of programs, for the results of one program can be used without further preparation as data in the next program.

Fortran words simplify the whole programming process and save the programmer an enormous amount of time and effort.

¹The point at which a given number is reached.

²Control Data Corporation, p. 12.

For illustrative purposes, a sample program is given below. Each of the foregoing terms is explained in the context of the program.

STATE-
MENT

NUMBER PROGRAM INSTRUCTIONS

EXPLANATION

* Program--Illustration

The title, chosen by the programmer, is for identification purposes only. The * signifies that the statement is a comment, not an instruction. It is always on the first card in the first space of the card. Statement numbers may not be repeated, they need not be in consecutive order, and are used only when future reference is to be made to that statement.

DIMENSION MUSIC (12)

In this case MUSIC is the name of a location in the memory section. It will store twelve items of information. MUSIC is the name by which reference is made to this information whenever it is needed. The programmer coins a great many words to be used as vehicles

STATE-
MENTNUMBER PROGRAM INSTRUCTIONSEXPLANATION

for computation, storage, shorthand, and for reference aids, such as MUSIC, INFO, KORT.

1 **FORMAT (12(I6))**

The instructions, always enclosed in parentheses following **FORMAT**, explain in this case that there will be twelve numbers, and each will be included within six spaces (field width) on an IBM card. If the numbers used are integers (no decimal points involved), the letter "I" precedes the field width.

READ 1, (MUSIC (I), I = 1, 12) The machine when following this instruction will record the twelve numbers given. The reference, "READ 1," tells the machine exactly how it will find the numbers arranged on the punched data card. A place was reserved in the **DIMENSION** statement for **MUSIC**.

DO 15 I = 1, 12

The statements following DO are to be executed in order, (unless a GO TO calls for a transfer to another part of the program), to and including statement number 15. Repeat this process twelve times, each time taking a successive value for (I) from 1 to 12. Each of the twelve repetitions of this sequence is referred to as a loop. For the first loop I = 1, and in the seventh loop I = 7.

IF (MUSIC (I)) 8, 10, 10

This statement calls for a decision. If the value of the quantity within the outer parenthesis is less than 0, the next statement to be executed is statement number 8. If the value of the quantity within the outer parenthesis is equal to 0, control will transfer to statement number 10, or whatever the middle number may be. If the quantity is more than 0,

control will transfer to the third number. All three numbers may be different, or any two may be the same.

8 GO TO 18

If the program has stopped here following the IF command, the next statement to be processed is number 18.

10 MUSIC (I) = MUSIC (I) + 1

Arithmetic symbols that may be used in Fortran are: + (addition), - (subtraction), * (multiplication), and / (division). The = sign means the expression on the left side is to be replaced by the expression on the right side.

15 CONTINUE

The statement CONTINUE indicates the return to the DO statement to begin the subsequent loop. At the last loop, the machine will not return to the DO statement from here, but will go to the next statement in line.

18 PRINT 20, (MUSIC (I), I = 1, 12) The values of MUSIC

will be printed according to the form indicated in FORMAT statement number 20. The numbers recorded in the twelve places under MUSIC will be printed.

20 FORMAT (1H0, 12 (I6))

1H0 indicates that the machine should double space before printing the answers. Then, it should proceed to print twelve answers across the page, each within six spaces.

STOP

END

(blank card)

The two words, STOP, END, and the blank card terminate the program.

40742 40942 40742 40542 41042 40942 40142 31232 40542 40542 40542 40642

Twelve five-digit numbers, each typed within six spaces, supply the data with which the program will work.

The spacing of the words and numbers on the cards is very important. Information about spacing is readily available in computer manuals.

The electronic computer is basically a mathematical, problem-solving instrument designed to manipulate numbers. Only, therefore, as musical problems can be expressed in mathematical terms, can these terms be organized into programs and solutions be found. This means that in preparing a piece of music for analysis by the computer, the various parts of the music must be translated into numbers. For purposes of this paper, this process is called "coding."¹ The numbers thus derived are placed in a computer array, which is a memory storage unit. A one-dimensional array contains a list of numbers in either columns or rows. A two-dimensional array is composed of numbers arranged in both columns and rows. These may be referred to by the number of the column, and of the row.

The array INFO will store measure numbers and indications of the position of chords within a measure. For instance, 00111 and 02822 both give three kinds of information. The first three spaces, 001 and 028, are the measure numbers; the fourth digit in each case indicates the beat within the measure, beat 1, beat 2; while the last figure in both of the numbers stands for the fraction of the beat: 1 is the first half of the beat, 2 is the second half of the beat.

The cards on which this data is stored may be used for many compositions. For instance, any composition in 4/4 meter which calls for an analysis of the chords on each half beat could use the same INFO array. The complete first measure of a composition in 4/4 meter would be notated thus: 00111, 00112,

¹This is not to be confused with the automatic coding of programs of the Fortran system.

00121, 00122, 00131, 00132, 00141, 00142. This could be broken down further into as many divisions as desired. If every $\frac{1}{4}$ of the beat is to be analyzed, it would be shown like this: 00111, 00112, 00113, 00114, 00121, 00122, 00123, 00124, 00131, 00132, 00133, 00134, 00141, 00142, 00143, 00144.

In $\frac{3}{4}$ meter, one chord to a count, the fifteenth measure would be written: 01511, 01521, 01531. The storage of this information as a two-dimensional array could be visualized as in the following table. On the punched IBM data card the information for a 142nd and 143rd measure (one chord recorded for each beat) would appear in a line:

1421114221142311424114311143211433114341, and in storage like this:

	Column 1	2	3	4	5	6	7	8
Row 1	142	142	142	142	143	143	143	143
Row 2	11	21	31	41	11	21	31	41

Figure 2. Storage of a two-dimensional array.

Since normal analysis procedures would indicate successive measures and successive beats or fractions of beats, this $\frac{4}{4}$ meter INFO array may be used for any composition in $\frac{4}{4}$ meter, and these cards would not have to be re-punched each time. There is no reason why a special INFO arrangement cannot be made. Contemporary music may demand it. For the music of Hindemith and Webern analyzed in this research, irregular INFO patterns were used.

Coding Music into Numbers

The following table shows the conversion of the notes of the scale to numbers:

C-----01	F# or G _b --07
C# or D _b --02	G-----08
D-----03	G# or A _b --09
D# or E _b --04	A-----10
E-----05	A# or B _b --11
F-----06	B-----12

Figure 3. Conversion of notes to numbers.

This is necessarily arbitrary. Decisions will need to be made by the programmer when the music involves double sharps and double flats or other notes not listed in the table.

The range of the piano keyboard is divided into seven and one-quarter octaves. In the figure on the next page, number 1 represents the Contra octave; number 2, the Great; number 3, Small; number 4, One-line; number 5, Two-line; number 6, Three-line; and number 7, Four-line. The octave number is used as the first of the three numbers which represent the note itself.

A five-digit number will more completely illustrate each note of the composition, as 41043. The first three digits represent the note proper. This note is in the fourth octave, the tenth note. The fourth figure is the octave number repeated, the fifth place represents the instrument which is sounding the note. The instruments involved in any composition

will have to be arbitrarily assigned before coding. Two spaces may have to be allowed for the instrument, and any composition having more than nine instruments would need to use both spaces for their representation. To illustrate, let the third horn be represented by 12 and its concert pitch note, middle C. This would be coded as 401412.

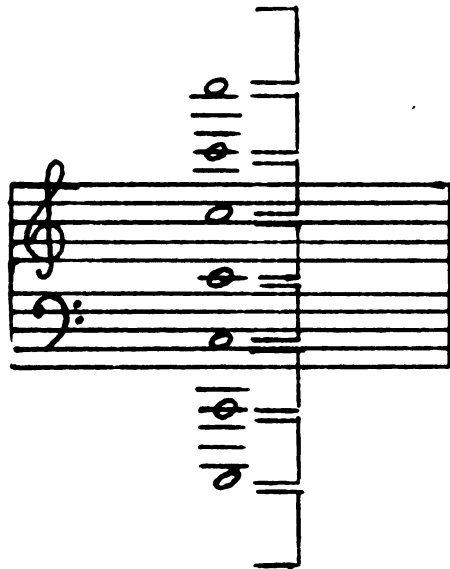


Figure 4. Chart for indicating first digit of coded note.

One more example of coding follows:

Measure	1	2	3	
				I Violin - 1
				II Violin - 2
				Viola - 3
				Cello - 4

Figure 5. Coding example.

The programmer names the various arrays. Since this study will not be using fractional or decimal numbers, only words beginning with the letters I, J, K, L, M, N, may be used. The words may not exceed six letters. The four voices of the example given here have been assigned the names, MUS, NOTE, KLANG, LONG, from top to bottom, respectively. This example is coding every half beat of the music.

MUS

41041, 41041, 50151, 50151, 50151, 41141, 41041, 40841, 40641,
40641, 40641, 40641

NOTE

40642, 40642, 41042, 41042, 40842, 40842, 40342, 40342, 40142,
40142, 40142, 40142

KLANG

40143, 40143, 40443, 40443, 40343, 40343, 31133, 31133, 31033,
31033, 31033, 31033

LONG

30634, 30634, 30734, 30734, 30834, 30834, 30134, 30134, 30634,
30634, 30634, 30634

To review, each note of a composition will be coded into one number of five digits. The first three figures depict the exact location of the note, i.e., the particular octave, and its placement within the octave. The fourth digit is a repeat of the octave number. The last figure in the first line is one identifying all these notes as belonging to the first violin. The other lines end with 2, 3, and 4, identifying their instrument.

The kinds of intervals within the octave are reduced to twelve, i.e., the enharmonic notes are counted only once. The following table is the coding table which is to be consulted for interpretation of interval value.

P1	m2(A1)	M2(d3)	m3(A2)	M3(d4)	P4(A3)		
00	01	02	03	04	05		
	89	90	91	92	93		
A4(d5)	P5(d6)	m6(A5)	M6(d7)	m7(A6)	M7	P8	
06	07	08	09	10	11	12	
94	95	96	97	98	99	100	
P---perfect		m---minor		A---augmented			
M---major		d---diminished					

Figure 6. Coding table for interval values.

For example, the interval between the lowest and highest notes in the first chord of Figure 5 is found by subtracting the two: 306 from 410. This results in the number, 104. To reduce any number above 12, use the following plan: The numbers between 101-112, 201-212, 301-312, 401-412, 501-512, 601-612, 701-712 are reduced simply by retaining the last two digits of the numbers. The numbers between 89-100, 189-200, 289-300, 389-400, 489-500, 589-600, 689-700, and 789-800 are reduced by dropping the first digit and subtracting 88. Therefore, 104 is reduced by retaining the 04; in 394 the 3 is dropped and 88 is subtracted from 94, resulting in 06.

Programs for Musical Analysis

Of the various phases of a musical analysis, the following have been processed through a computer for use in this study:

- 1) Harmonic Intervals, with rhythmic, octave, and instrument locations
- 2) Frequency of chord repetition
- 3) Repeated Chord Progressions
- 4) Hindemith Roots
- 5) Melodic Intervals (also for Hindemith roots)
- 6) Frequency of Melodic Intervals (also for Hindemith roots)

The distance between notes (intervals) is found by counting up or down. For instance, from C to E is three notes, and this interval is called a third. After coding, intervals are found by subtracting the number representing one note from the number representing the other note. For example, notes C and E are represented by the numbers, 401 and 405. The difference is 4, which in Figure 6 is shown to be a major third.

The first five programs in this research are concerned with finding the harmonic intervals in chord structures, with their rhythmic, octave and instrument locations. These harmonic intervals are found by subtracting the lowest-sounding note from every other note present in the chord.

Program One uses the coded notes for input and prints out on paper and punches out on IBM cards the intervals. Selected for developing this program is a string quartet. The intervals are found by subtracting the cello part from the first violin, the cello from second violin, and cello from viola. This is comparatively simple, unless one of the voices is absent, or unless a voice crosses below the cello

part. Program revisions are continually necessary for meeting these and other unforeseen problems.

The simple, three-measure, four-part musical example in Figure 5 is used to illustrate the basic steps described in Programs One through Five. (This is not to be confused with the programming of the Beethoven String Quartet in the next chapter which also uses Programs One through Five.)

The four voices, from top to bottom, were given the array names, MUS, NOTE, KLANG, and LONG. The instructions provided for LONG to be subtracted from MUS, NOTE, and KLANG. The results were:

<u>FIRST CHORD</u>			<u>SECOND CHORD</u>			<u>THIRD CHORD</u>			
410	406	401	501	410	404	501	408	403	
<u>306</u>	<u>306</u>	<u>306</u>	<u>307</u>	<u>307</u>	<u>307</u>	<u>308</u>	<u>308</u>	<u>308</u>	
Difference:	104	100	95	194	103	97	193	100	95
<u>FOURTH CHORD</u>			<u>FIFTH CHORD</u>			<u>SIXTH CHORD</u>			
411	408	403	410	403	311	408	403	311	
<u>308</u>	<u>308</u>	<u>308</u>	<u>301</u>	<u>301</u>	<u>301</u>	<u>301</u>	<u>301</u>	<u>301</u>	
Difference:	103	100	95	109	102	10	107	102	10
			<u>SEVENTH CHORD</u>						
			406	401	310				
			<u>306</u>	<u>306</u>	<u>306</u>				
Difference:			100	95	4				

The Print Out lists the differences (intervals), three for each chord, thus:

104,	100,	95
194,	103,	97
193,	100,	95
103,	100,	95
109,	102,	10
107,	102,	10
100,	95,	4

Since subsequent programs need to use these intervals in a different order, Program Two rearranges the numbers in the vertical columns to be read horizontally, thus:

104,	194,	193,	103,	109,	107,	100
100,	103,	100,	100,	102,	102,	95
95,	97,	95,	95,	10,	10,	4

Just as intervals are often reduced to within the range of an octave, so the intervals listed above may all be reduced to figures of 12 or below (see explanation, page twenty-eight). This reduction is made in Program Three. The table then reads:

4	6	5	3	9	7	12
12	3	12	12	2	2	7
7	9	7	7	10	10	4

Program Four illustrates that the numbers would be even more readable if each vertical column is rearranged in a numerically descending order:

12	9	12	12	10	10	12
7	6	7	7	9	7	7
4	3	5	3	2	2	4

This step of arranging the intervals in numerically descending order is important to the process of comparing the chords. Otherwise, it would be difficult, if not impossible, for the machine to make the comparisons.

Program Five compiles all the information resulting from Programs One through Four and prints them, after conversion, in table form:

<u>Measure No.</u>	<u>Rhythmic Position</u>	<u>Harmonic Intervals</u>	<u>Octave</u>	<u>Instrument</u>
001	11	8.	4	1
		5.	4	2
		3.	4	3
		F	3	4

Figure 7. Information from Program Five.

Included in the column under Harmonic Intervals is the name of the lowest sounding note, F. The point (.) designates a major or perfect interval; the dash (-) indicates a minor interval, and the addition sign (+), an augmented interval.

Program Six, Frequency of Chord Repetition, uses the output from Program Four for its basis. In the music under investigation any chord structure that is repeated is printed out.

The same procedure is followed for repeated chord progressions in Program Seven.

The contemporary composer and theorist, Paul Hindemith, has developed a method for finding the roots of chords that is usable for even the complex chords of twentieth-century music. Since it will work for any type of chord, this method is valuable for analyzing the music of all periods. Not only the intervals from the lowest note, but all other inner intervals, are found in Program Eight. In the first chord of Figure 5, for instance, there would be six intervals rather than three. They would be:

	410	406	401	410	406	410
	<u>306</u>	<u>306</u>	<u>306</u>	<u>401</u>	<u>401</u>	<u>406</u>
Difference:	104	100	95	09	05	04

These are reduced, as in Program Three, to: 4, 12, 7, 9, 5, 4.

Hindemith has arranged all the intervals in what he considers the order of their importance. The results obtained above, 4, 12, 7, 9, 5, 4, are checked against a list developed by Hindemith, which is given after "A" in the table below. In the Hindemith list, the most important interval is 7 (perfect fifth); if there is a 7 in the list from the above example, it is considered the most important interval. In the intervals listed after the array, B, the lower note is the root; for those after C, the upper note is the root, and for the tritone (6), there is no root. In the above example, then, the 7 (the difference between 401 and 306), is the most important interval. Since it appears in the array marked B, the lower note is the root. Therefore, 306, an F, is the root of this chord. In case the most important interval appears more than once, the

lower (or lowest) note takes precedence. This process of finding the root is repeated for each chord.

A	7, 5, 4, 8, 3, 9, 2, 10, 1, 11, 12, 0, 6
B	0, 3, 4, 7, 10, 11, 12
C	1, 2, 5, 8, 9

Figure 8. Hindemith's table for choice of interval value and chord root.

Program Nine deals with Melodic Intervals. In finding melodic intervals, the subtractions take place horizontally rather than vertically:

	501	501	411	410	408	406
	<u>410</u>	<u>501</u>	<u>501</u>	<u>411</u>	<u>410</u>	<u>408</u>
Difference:	+91	0	-90	-01	-02	-02

The positive numbers indicate an upward movement, and the negative numbers a downward progression. Reduced and interpreted, these are shown below:

+03	0	-02	-01	-02	-02
-----	---	-----	-----	-----	-----

or minor third, repetition, major second, minor second, major second, and major second. This program also finds the melodic intervals between the Hindemith roots.

Frequency of melodic intervals for each instrument and the Hindemith roots are tabulated in Program Ten.

CHAPTER III

RESULTS OF ANALYSIS ARRIVED AT THROUGH APPLICATION OF THE PROGRAMS TO MUSIC OF VARIOUS STYLES

The Beethoven String Quartet in C# Minor, Opus 131, First Movement, is used as the major vehicle for developing the Analytical Programs One through Ten. To further illustrate the applicability of these programs to various styles and to a varying number of parts, analyses of sections from the following compositions are also made: Hindemith's Kleine Kammermusik, Opus 24, No. 2, Webern's Six Bagatellen for String Quartet, Opus 9, Schumann's Papillons, Opus 2, Mozart's Symphony in G Minor, and Stravinsky's Symphony in Three Movements.

In all of these sets of programs there is an insufficient amount of music coded and analyzed to make any type of conclusive analytical statement. The primary purpose of this research is to provide a methodology which may be utilized for research rather than to produce meaningful analyses. The results are indicative, though, of the kind and amount of information which can be gained from a computer analysis.

All the computer results are tabulated in Chapter Seven, but for illustrative purposes several representative results are presented in this chapter. The significant results from the programs developed in this study are found in Programs Five,

Six, Seven, Eight, and Ten. Programs One through Five could have been processed as one program if the memory storage of the 160-A Computer had had enough units. As it is, the results from Program One are used as data for Program Two, the results from Program Two as the basic information for Program Three, etc., up to Program Five. Likewise, Program Nine is not in itself significant but furnishes data for Program Ten. For a clear understanding of the findings of this study, each of the ten programs is outlined.

Beethoven: Quartet, Opus 131¹

Program One surveys the notes of each chord to be analyzed, and finds the smallest number, which represents the lowest note of the chord. The machine uses this note as the one from which all the other notes of the chord are measured, even though it may not be the bass note. This measurement results in three intervals for each chord, which are the figures listed in the three columns under the heading INTERVALS on page ninety-seven. Each horizontal row of numbers represents a chord. The figures in each row may be interpreted by referring to page twenty-eight; however, interpretation of these intervals is included in the results of Program Five. Using the process described on page fifty-eight, five multiples of eighty chords (400) were used for Program One only. This program is found on page seventy-seven and its results on page eighty-two.

Program Two rearranges the numbers in the vertical columns to be read horizontally, as illustrated on page

¹The music used for this analysis is found on the next page.

This page of the musical score for Quartet No. 14, Opus 131 by L. van Beethoven, contains measures 5 through 20. The score is written for four staves, each with a key signature of three sharps (F#, C#, G#) and a common time signature (C). The notation includes various musical symbols such as notes, rests, and slurs. Measure numbers 5, 10, 15, and 20 are clearly marked above the staves. The music features complex rhythmic patterns and melodic lines across all four parts.

thirty-one. The actual data is arranged in eight columns and thirty rows. The eighty chords analyzed from the Beethoven String Quartet are represented by thirty lines of the output.

The output of Program Three, page 106, is in the same tabular form as Program Two, but all the numbers have been reduced to twelve or below. The notes are still in coded form.

The results of Program Four, pages 109 through 111, are tabulated in three columns, each horizontal line representing a chord. The numbers in each row are arranged from left to right in a numerically descending order.

Program Five's findings are tabulated on pages 123 through 134. For further interpretation of these figures, the reader is referred to page thirty-two. Every vertical sonority that occurs on the metrical beats shown in the second column of Program Five is analyzed. The table obtained presents a harmonic outline of the musical composition analyzed.

The harmonic intervals from the results of Program Five are given below:

[illegible]

3-	5.	3-	3-	5.	4+	4+	6-	7-	5.	6-	6-	7-	5.	3.	7-
3-	3.	4.	5.	3-	3-	3.	3-	3.	3.	3-	3-	3.	3-	5.	3.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C+
4+	6.	6.	3.	6-	6.	6.	8.	3-	6.	8.	3-	5.	4.	3-	2-
6.	4+	3.	5.	8.	6.	6.	6.	8.	4+	6-	8.	3.	2.	8.	7-
3-	3-	3.	8.	3.	4.	3.	3-	3-	3-	3-	3-	3.	3-	3-	4.
F	F	F+	F+	D	D	D	C+	B	D	C+	B	A	B	C+	C+
2-	6.	8.	3-	5.	4.	3.	3-	6-	3.	8.	6.	6-	5.	5.	2.
6.	4+	6.	8.	3-	6.	6.	6.	6.	6.	3-	3-	8.	7-	8.	6.
4.	3-	3.	3-	3-	3-	3.	3-	3-	8.	6.	4+	4.	3.	3.	4+
C+	E	D	C+	B	C+	D	E-	E-	E	F+	G	A-	A	A-	F+

Chart 1. Harmonic intervals of Program Five.

The section of music analyzed is a fugue. It is not until the fourth measure that a second voice enters and a harmonic interval is shown on the table. Between the second half of the second beat of the fourth measure and the second half of the second beat of the eighth measure when a third voice enters, there are five minor thirds, two major thirds, one perfect fourth, two augmented fourths, three perfect fifths, two minor sixths, and one major sixth.

The three-voice section which extends to the second half of the second beat of the twelfth measure contains four major chords indicated by 5.; four minor chords, 5. and 3-; three 3. 3- 3- major chords in first inversion, 6-; a major-minor seventh 3- chord, 7-; a diminished chord, 4+; a second inversion diminished- 3.

minor seventh chord, 4+; and one second inversion of a major-
3.

minor seventh chord, 4.
3-.

The remainder of the section has four voices and contains the following chords:

Six major chords, four in root position, one in first inversion and one in second inversion.

Ten minor chords. Of the five in root position, four are incomplete chords lacking the fifth. Four minor chords are in first inversion and one in second inversion.

Three diminished triads in first inversion are present, as well as five diminished-diminished seventh chords.

Four major-minor seventh chords, two in root position, one in second inversion, and one in third inversion.

One augmented chord.

Four non-classified chords, which are probably the result of non-harmonic tones.

The results of Program Six on page 142 are given in terms of the positive and negative numbers of the following table:

1 -- no interval	5 -- perfect fifth
-2 -- minor second	-6 -- minor sixth
2 -- major second	6 -- major sixth
-3 -- minor third	-7 -- minor seventh
3 -- major third	7 -- major seventh
4 -- perfect fourth	8 -- perfect octave
-5 -- diminished fifth	

Figure 9. Interval interpretation.

The chord is formed by reading the intervals horizontally.

PROGRAM 6

FREQUENCY OF CHORD REPETITION

<u>Number of Entry</u>	<u>Chord</u>		
1.	5	1	1
2.	5	1	1
3.	-3	1	1
4.	-3	1	1
5.	-3	1	1
6.	-3	1	1
7.	-5	1	1
8.	3	1	1
9.	-6	1	1
10.	5	3	1
11.	5	3	1
12.	5	3	1
13.	5	-3	1
14.	5	-3	1
15.	-6	-3	1
16.	-6	-3	1
17.	-7	3	1
18.	-7	5	3
19.	6	-5	-3
20.	6	-5	-3
21.	6	-5	-3
22.	5	3	3
23.	8	5	3
24.	8	6	-3
25.	8	-3	-3
26.	8	-3	-3
27.	8	-3	-3
28.	6	4	-3
29.	8	6	3

Chart 2. Results shown in Program Six.

The results of Program Six show (third to sixth entry) five minor chords, -3 1 1. Entry 13 and 14 account for two more minor chords, 5 -3 1. Also, 8 -3 -3, which appears three times beginning on the twenty-fifth line, outlines four

minor chords. The last entry, 8 6 3, identifies two first inversion minor chords. This is an analysis of the minor chords that appeared beginning with the two-voice texture of the fugue.

If a chord is repeated one or more times, it is listed in the results of Program Six. To determine the number of times a particular chord appears, count the number of times it is listed and add one.

The results of Program Six are not as complete as those of Program Five, as Program Six is not able to distinguish similar chords which have omitted chord members. For instance, these three chords are all first inversion major chords:

-6	8	8
-6	-6	-6
-3	-3	-6

Two minor sixths and a minor third, or an octave, minor sixth, and minor third, or an octave and two minor sixths all basically outline a major chord in first inversion; but the machine, which is looking for an exact repetition of numbers, does not recognize these as being the same basic chord.

The results of Program Seven on pages 151 through 153 list groups of chords separated by the word END. If all the chords within one group have the same intervals, for instance,

5	3	0
5	3	0
5	3	0

0, these chords are to be ignored. They are chords which the computer was investigating, but which were found not to be a part of a progression repetition.

PROGRAM 7

CHORD PROGRESSION

Number
of group

1.	5	1	1
	-3	1	1
	END		
2.	-3	1	1
	-3	1	1
	-3	1	1
	END		
3.	5	1	1
	-3	1	1
	END		
4.	-3	1	1
	END		
5.	5	3	1
	5	3	1
	5	3	1
	END		
6.	5	3	1
	5	3	1
	END		
7.	5	-3	1
	END		
8.	-6	-3	1
	-6	-3	1
	-7	3	1
	END		
9.	-7	5	3
	END		
10.	6	-5	-3
	6	-5	-3
	END		
11.	6	-5	-3
	6	-5	-3
	END		
12.	5	3	3
	END		
13.	8	5	3
	END		
14.	8	6	-3
	END		
15.	8	-3	-3
	8	-3	-3
	8	-3	-3
	END		
16.	6	-5	-3
	END		

PROGRAM 7 Continued

Number
of group

17.	8	-3	-3
	8	-3	-3
	END		
18.	8	-3	-3
	END		
19.	6	4	-3
	END		
20.	8	6	3
	END		

Chart 3. Results of Program Seven.

If the chords within any one group vary, as in the first, third, and eighth groups, a chord progression repetition is indicated. To find the frequency of any progression's repetition, count the number of groups in which the particular progression occurs and add one. The chord progression, 5 0 0, followed by -3 0 0, (listed twice), appears three times, and the chord progression, -6 -3 0, followed by -7 3 0, (listed once), occurs twice. All the numbers listed in Program Seven can be interpreted by referring to Figure 9, page forty.

The first entry signifies the repetition of the intervals of a perfect fifth followed by a minor third. This progression appears four times, since it is denoted twice in the table.

The first real chord progression is revealed in the numbers, -6 -3 1, -6 -3 1, -7 3 1. The progression of a minor sixth, minor third (first inversion major chord)

followed by a major-minor seventh chord, -7 3 1, is repeated twice. There are no other chord repetitions indicated.

The Hindemith roots of the chords are given in Program Eight, on page 163, but they are still in code and need to be translated. This is done by dropping the first digit (which signifies in which octave it is located) of each number and referring to the table below for the interpretation of the last two digits.

C-----01 or 89	F# or G $\flat$ --07 or 95
C# or D $\flat$ --02 or 90	G-----08 or 96
D-----03 or 91	G# or A $\flat$ --09 or 97
D# or E $\flat$ --04 or 92	A-----10 or 98
E-----05 or 93	A# or B $\flat$ --11 or 99
F-----06 or 94	B-----12 or 100

Figure 10. Table for de-coding numbers into notes.

On the next page a comparison is made of the Hindemith roots from Program Eight, the lowest sounding notes, as given in Program Five, and the roots as determined by traditional analysis. The Hindemith roots are listed on the top line, the lowest sounding notes on the second line, and the traditional roots on line three in each case.

The frequency of roots appeared in the following order:

<u>Hindemith roots</u>				<u>Traditional roots</u>			
C#,	10	G#,	4	C#,	10	G#,	6
F#,	9	G,	4	F#,	10	G,	1
B,	9	E# (F)	3	B,	7	E#,	6
A,	8	C,	2	A,	5	D#,	4
D,	7	D#,	1	D,	5	A#,	2
E,	6			E,	0	F ^x ,	1
						B#,	2

Chart 4. Frequency of root occurrence.

The modulations inherent in any fugual structure would distort the normal root occurrence that one would expect to find in tonal writing.

Measure 1	2	3	4
			C# - C#
Measure 5	6	7	8
F(E#)* B F#	D D D A	G D C# *	F# B C# G#
- - - -	- - - -	- - - -	- - - -
C# C# F# F#	D D D F#	E# E# E# E#	F# B C# G#
Measure 9	10	11	12
C(B#)C E C#	A A A E	F# A E D	E F# G# C#
- - - -	- - - -	- - - -	- - - -
G# G# C# C#	B# B# D# E	F# F# E D	E F# G# C#
Measure 13	14	15	16
F F F# F#	D G B C#	B D A B	A E C# F#
F F F# F#	D D D C#	B D C# B	A B C# C#
E# E# F# F#	D G B A#	B G# A B	A A A B
Measure 17	18	19	20
F# E B C#	B F# B D#	B C# F# G	C# A G# G#
C# E D C#	B C# D D#	D# E F# G	G# A G# F#
F# A# B C#	B F# B D#	D# C# D# F ^x	C# A G# G#

*Tritone has no root.

Chart 5. Comparison of Hindemith roots, lowest sounding notes, and traditional roots.

The melodic intervals of each instrument of the String Quartet are given in Program Nine under the appropriate headings: I Violin, II Violin, Viola, and Cello. Also listed in the results of Program Nine are the melodic intervals between the Hindemith roots. The melodic intervals are given as either positive or negative numbers. The positive numbers indicate downward melodic movement, negative numbers upward melodic movement. If the

number lies between 0-100, the melodic interval is within the octave; if it is between 101 and 200, the melodic interval is within two octaves; if between 201 and 300, three octaves.

The melodic intervals listed as results of Program Nine are reduced and printed under appropriate headings in Program Ten. Each type of melodic interval is given with the frequency of its occurrence.

Five Frequency Tables from Program Ten Results

P1	m2	M2	m3	M3	P4	A4	P5	m6	M6	m7	M7	P8
----	----	----	----	----	----	----	----	----	----	----	----	----

I Violin

30	11	10	3	2	0	0	0	0	1	0	0	0
-30	-8	-8	-2	-1	-2	0	0	0	-1	0	0	-1

II Violin

27	9	8	4	2	0	0	1	0	0	0	0	0
-27	-13	-6	-4	-2	-1	-1	-1	-1	0	0	0	-1

Viola

44	8	10	0	2	0	0	0	0	0	0	1	0
-44	-3	-7	-2	-1	0	0	-1	0	0	0	0	-2

Cello

56	4	6	0	1	0	0	0	0	0	0	0	0
-56	-7	-4	-2	-1	0	0	0	0	0	0	0	0

Hindemith Roots

28	2	7	2	4	3	0	4	1	0	6	0	0
-28	-2	-3	-2	-5	-2	-1	-7	-1	0	0	0	-1

Chart 6. Frequency of melodic intervals.

Results of Program Ten begin with a listing of all the melodic intervals in the first violin, i.e., melodic intervals between coded notes. These melodic intervals are codified under

the headings of perfect prime, minor second, major second, minor third, etc., in the frequency table. It will be noted that there are few large skips: one octave, one major sixth, and two perfect fourths. The movement is very chromatic, as is shown by the large number of minor seconds.

The last frequency table, denoting melodic intervals between the Hindemith roots, reveals two progressions upward (those indicated as negative numbers) and two downward of a minor second. Seven progressions of a major second are downward, and three are upward. Two progressions are up, and two are down a minor third. Four are up a major third and five are down. This reveals that the roots are progressing primarily in seconds and thirds, as there are twenty-seven of these as against seven progressions of a fourth or a fifth. Of the eight remaining progressions, one is up a minor sixth, one down a minor sixth, and six are up a minor seventh.

The results of the other compositions (Hindemith's Kleine Kammermusic, Opus 24, No. 2, Webern's Six Bagatellen for String Quartet, Opus 9, Mozart's Symphony in G Minor, Stravinsky's Symphony in Three Movements, and Schumann's Papillons, Opus 2) are to be interpreted similarly to the basic set just described, with the exception that Programs Two and Three are usually combined into one program, called Program Three.

Hindemith Quintet, Opus 24, No. 2

A selection from the Fifth Movement of Hindemith's

Quintet, Opus 24, No. 2 was processed to test the analysis programs for use with a five-voice, contemporary composition. The page numbers locating the significant results are shown below:

Program One, Harmonic Intervals, page 178

Program Five, Harmonic Analysis, page 195

Program Six, Repeated Chords, page 207

Program Seven, Repeated Chord Progressions, page 217

Program Eight, Roots of each Chord, page 224

Program Ten, Frequency of Melodic Intervals, page 233

Frequency of Upward and Downward Root Movements, page 235.

Program Five Results

The harmonic intervals as found in Program Five are listed below:

8.	8.	8.	6.	3.	6.	7.	6.	4.	6.	8.	8.	6.	3.	6.	8.
8.	8.	8.	6.	3.	6.	7.	6.	4.	6.	8.	8.	6.	3.	6.	8.
3.	3.	3.	2-	6-	2-	3.	2-	6.	2-	3.	3.	2-	6-	2-	3.
6.	6.	6.	6.	6.	6.	6.	6.	7-	6.	6.	6.	6.	6.	6.	6.
E	E	E	D	D	D	E	D	C+	D	E	E	D	D	D	E

Chart 7. Harmonic intervals of Program Five.

The sonorities listed in the table for Program Five show only five different chords. The first is composed of an octave, a major sixth, and a major third; it appears six times. Composed of a major sixth and a minor second, the second chord is repeated six times. The third chord has a major sixth, a minor sixth, and a major third and appears twice. The last two sonorities occur only once, the first being constructed of a major seventh, a major sixth and a major third; the second of a minor seventh, a major sixth and a perfect fourth.

Webern: Quartet, Opus 9

The Webern String Quartet had to be coded in eight voices because of the many double stops employed. If double, triple, or quadruple stops are used in the strings, more than one array will have to be reserved for each instrument. The results are found on the pages listed below:

Program One, Harmonic Intervals, page 298

Program Five, Harmonic Analysis, page 313

Program Six, Repeated Chords, page 324

Program Seven, Repeated Chord Progressions, page 326

Program Eight, Roots of each Chord, page 331

Program Ten, Frequency of Melodic Intervals, page 340

Frequency of Upward and Downward Root Movements, page 341.

In Program Four the numbers are reduced and arranged in numerically descending order; in Program OMITT all interval repetitions are eliminated and the number of intervals is reduced to four. These results are given in order to show that in the reduction from seven to four intervals no significance was lost.

PROGRAM FOUR

8	3	0	0	0	0	0
11	11	9	3	2	0	0
12	12	8	7	7	0	0
10	9	9	4	4	0	0
8	6	0	0	0	0	0
11	11	8	8	3	3	1
11	11	10	5	5	3	3
11	11	10	9	9	5	5
7	7	7	7	4	3	3
9	8	1	1	0	0	0

PROGRAM FOUR (Cont.)

7	1	1	0	0	0	0
10	4	2	2	0	0	0
11	7	3	1	1	0	0
12	10	4	0	0	0	0

Chart 8. Results of Program Four.

PROGRAM OMITT

8	3	0	0
11	9	3	2
12	8	7	0
10	9	4	0
8	6	0	0
11	8	3	1
11	10	5	3
11	10	9	5
7	4	3	0
9	8	1	0
7	1	0	0
10	4	2	0
11	7	3	1
12	10	4	0

Chart 9. Results of Program OMITT.

Schumann: Papillons, Opus 2

This piano composition was attempted in order to demonstrate a different technique in coding. It was coded horizontally rather than vertically. This was necessary since most piano compositions do not have a clearly delineated number of voices present throughout. The specific procedure is outlined in Chapter IV. The results are found on the pages listed below:

Program One, Harmonic Intervals, page 344

Program Five, Harmonic Analysis, page 363

Program Six, Repeated Chords, page 373

Program Seven, Repeated Chord Progressions, page 375.

Program Seven Results

The numbers in the results of Program Seven must be de-coded before they can be interpreted. The first column below shows the numbers as given in the results; the second column reduces them to musical intervals.

<u>Results</u>			<u>Interpretation</u>		
12	7	4	8	5	3
12	8	3	8	6-	3-
12	9	5	8	6	4
END					
12	7	4	8	5	3
12	8	3	8	6-	3-
END					
12	7	4	8	5	3
12	8	3	8	6-	3-
END					

Chart 10. De-coding of Program Seven results.

This shows that the progression of a major chord in root position, 8 5 3, first inversion, 8 6- 3-, second inversion, 8 6 4, occurs twice. The progression of a major chord root position moving to first inversion occurs four times.

Mozart: Symphony No. 40

A symphonic selection from the Classical Era was analyzed. The third movement from Mozart's Fortieth Symphony contains thirteen parts. The results are shown on the pages listed below. It may be interesting to note that in Program Eight the most important interval (in deriving the Hindemith root) for each chord had to be selected from the seventy-eight intervals present within every chord. In the Stravinsky

selection there were 231 intervals for each chord. The high speed of the digital computer made this a relatively simple task.

Program One, Harmonic Intervals, page 239

Program Five, Harmonic Analysis, page 259

Program Six, Repeated Chords, page 271

Program Seven, Repeated Chord Progressions, page 279

Program Eight, Roots of each Chord, page 285

Program Ten, Frequency of Melodic Intervals, page 293

Frequency of Upward and Downward Root Movements, page 294.

Root Movement Analyses

The graph below outlines the root movement that is shown in the results of Program Ten under the headings, Hindemith Roots and Melodic Intervals:

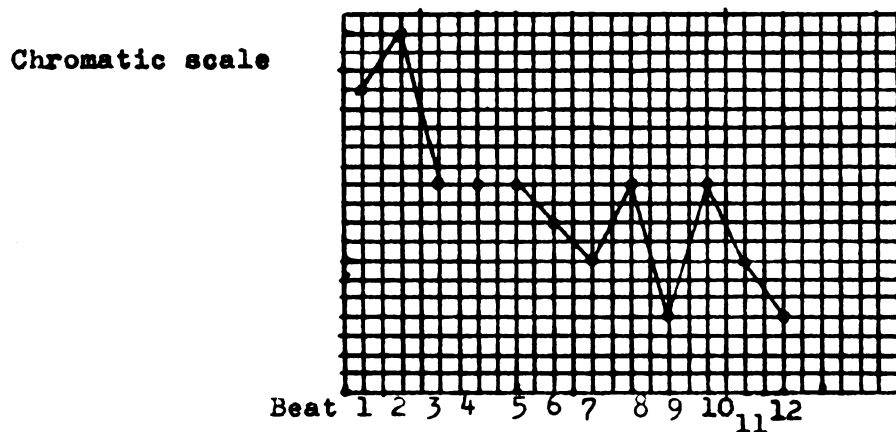


Chart 11. Graph showing root movement. The lines in the table represent distances up or down in half steps.

The root movements here, in contrast to the contemporary compositions, have no minor seconds. There are two movements of a major second, four of a minor third, two of a major third, one of a perfect fourth up, and one each up and down of a perfect fifth.

Stravinsky: Symphony in Three Movements

A contemporary composition of large proportions is used to test the versatility of this basic analysis method. Chosen for this was a small portion of Stravinsky's Symphony in Three Movements, with twenty-one voices. The results are shown on the pages listed below:

Program One, Harmonic Intervals, page 381

Program Five, Harmonic Analysis, page 409

Program Six, Repeated Chords, page 419

Program Seven, Repeated Chord Progressions, page 421

Program Eight, Roots of each Chord, page 429

Program Ten, Frequency of Melodic Intervals, page 439

Frequency of Upward and Downward Root Movements, page 439.

A chart of the analyzed chords made from the results of Program Five is provided on the next page to show the information in another form.

Three major chords are listed in the results of Program Five (8 3, 8 3, 8 6 4). Two major chords with an added major second (8 5 3 2) and two chords composed of a perfect fifth and major seventh are also evident. One chord has only a minor sixth; one a major sixth, minor third and major second; and one a perfect fourth and major second.

Chord	1	2	3	4	5	6	7	8	9	10	11	12
Octave	X	X	X	X	X	X	X	X	XX	X	X	X
Seventh							X	X			X	
Sixth									X	X		
Fifth			X	X			X	X			X	
Fourth										X		X
Third	X	X	X	X								
Second			X	X					X			X

Chart 12. Analyzed chords from Program Five results.

CHAPTER IV

DATA PREPARATION AND USE OF THE ANALYSIS PROGRAMS

The programs developed in this study may be used again and again to analyze pieces of music similar to those illustrated herein. The Quartet programs and accompanying card decks will be on file, with instructions for their use, with the librarian at the Computer Center.

In addition to the program it is necessary to have the data furnished from a given piece of music. The charts and tables developed in this chapter are to aid in setting up the data sheet from which the Fortran Statement cards are punched. Although it is possible to punch the cards directly from the music, it may be done more easily and quickly from a data sheet. This sheet may be prepared for each instrument and for the array INFO. INFO records the metrical position of the chords to be analyzed, but is used only in Program Five.

The tables in this chapter are based on the eighty-column Fortran Statement cards. This is the type of IBM card used at Michigan State University on which all data to be used in conjunction with these analysis programs is to be punched. These cards may be punched in the data preparation room of the Computer Center.

The instructions of the program are terminated by a card punched with the word **END**. A blank card is placed

between this card and the beginning of the data, which is also punched on IBM cards.

For use in making up the data sheets there is in each unit of the tables a five- or six-digit number, depending on the program. Notes coded with six digits use the following form:

<u>1st digit</u>	<u>2nd digit</u>	<u>3rd digit</u>	<u>4th digit</u>	<u>5th digit</u>	<u>6th digit</u>
Octave number	Chromatic scale degree		Octave number	Instrument number	

Notes coded with five digits use this form:

<u>1st digit</u>	<u>2nd digit</u>	<u>3rd digit</u>	<u>4th digit</u>	<u>5th digit</u>
Octave number	Chromatic scale degree		Octave number	Instrument number

The array INFO also uses five digits, according to Figure 11:

<u>1st digit</u>	<u>2nd digit</u>	<u>3rd digit</u>	<u>4th digit</u>	<u>5th digit</u>
Measure number requires three digits			Position of the note within the measure	

Figure 11. Coding system for INFO.

The array INFO is regular if the same number of chords are analyzed in each measure and if the chords analyzed are uniformly spaced. An irregular INFO has a varying number of chords per measure, or the analysis may be made of irregularly spaced chords, or both. In this case only the positions of the chords analyzed are recorded in INFO.

Double bars in the charts indicate the end of a measure.

For the chamber works and the symphonic compositions each part or instrument is coded separately. Piano music, which does not use a pre-determined number of voices, requires a vertical reading of each chord. In the analysis of piano music, eight places are reserved for each chord. Even if there are more than this number of notes present in a chord, it is assumed that there are some repeated notes, and these would be omitted.

Additional arrays needed for Program Eight are named MOE, JOE, KOE, and LOE. This is true for Program Eight in all the pieces of music. Beginning with column one on four Fortran Statement cards, the following numbers are to be punched. (A - indicates a blank space.)

MOE (first card)	--7--5--4--8--3--9--2-10--1
JOE (second card)	--0--3--4--7-10-11-12
KOE (third card)	--1--2--5--8--9
LOE (fourth card)	--6

Program One was processed twice, first with a set number (eighty) of chords to be analyzed and secondly, with the possibility of analyzing a varying number of chords. The first program in Chapter VII is the variable one. Following it is the set program of eighty chords. To increase or decrease the amount of data in the variable program, refer to statement number 104 in the variable Program One. This statement controls the amount of data that can be used. (LRETET - 5) will allow for 400 chords. The digit five controls the multiples of eighty chords that

will be processed. If the five is changed to four, for instance, the program will process only 320 chords.

Charts and Tables for Data Preparation

Beethoven's String Quartet in C# Minor, Opus 131, First Movement
(Four voices) Pages 77 to 174.

<u>Array names</u>	<u>Instrument</u>	<u>Instrument's number</u>
MUS	I violin	1
NOTE	II violin	2
KLANG	Viola	3
LONG	Cello	4

Figure 12. Classification of instruments.

Below is the INFO for the first eighty chords (twenty measures). Two lines of the table equal the information for one IBM card.

INFO

00111	00112	00121	00122	00211	00212	00221	00222
00311	00312	00321	00322	00411	00412	00421	00422
00511	00512	00521	00522	00611	00612	00621	00622
00711	00712	00721	00722	00811	00812	00821	00822
00911	00912	00921	00922	01011	01012	01021	01022
01111	01112	01121	01122	01211	01212	01221	01222
01311	01312	01321	01322	01411	01412	01421	01422
01511	01512	01521	01522	01611	01612	01621	01622
01711	01712	01721	01722	01811	01812	01821	01822
01911	01912	01921	01922	02011	02012	02021	02022

Figure 13. Form for coding INFO and instruments.

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
	Order of data	PRINT	PUNCHED CARDS TO:	
I	MUS, NOTE, KLANG, LONG	Yes	II; Remove last 5 for Va.	1
II	Output from I	No	III	2
III	Output from II	Yes	IV and V	3
IV	Output from III	Yes	VI and VII	4
Va	Last 5 cards from Output I	No	V	2
V	INFO, MUS, NOTE, KLANG Output from Va and III	Yes	None	4
VI	Output from IV	Yes	None	5
VII	Output from IV	Yes	None	5
VIII	One card each from MUS, NOTE, KLANG, LONG. Repeat until all cards are used. After the first LONG in- sert MOE, JOE, KOE, and LOE.	Yes	IX	1
IX	MUS, NOTE, KLANG, LONG Output from VIII	Yes	X	2
X	Output from IX	Yes	None	3

Figure 14. Master plan for data movement.

Hindemith's Quintet, Opus 24, No. 2, Fifth Movement

(Five voices) Pages 175 to 235.

<u>Array names</u>	<u>Instrument</u>	<u>Instrument's number</u>
MUS	Flute	1
NOTE	Oboe	2
KLANG	Clarinet	3
KORN	Horn	4
LONG	Bassoon	5

Figure 15. Classification of instruments.

Below is the INFO for sixty-four chords (sixteen measures).

Two lines of the table equal the information for one IBM card.

INFO

00111	00141	00211	00241	00251	00261	00311	00341
00351	00361	00411	00441	00471	00481	00491	00511
00541	00611	00631	00641	00661	00711	00731	00741
00761	00811	00841	00861	00911	00931	00941	00961
00971	00991	01011	01041	01111	01141	01151	01161
01211	01241	01251	01261	01311	01341	01371	01381
01391	01411	01421	01431	01441	01451	01461	01511
01521	01531	01541	01551	01561	01611	01621	01631

Figure 16. Form for coding an irregular INFO and instruments.

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
		PRINT	PUNCHED CARDS TO:	
I	MUS, NOTE, KLANG, KORN, and LONG	Yes	III; Remove last 4 for Va	1
III	Output from I	Yes	IV and V	2
IV	Output from III	Yes	VI and VII	3
Va	Last 4 cards from Output I	No	V	2
V	INFO, MUS, NOTE, KLANG, KORN Output from Va and III	Yes	None	3
VI	Output from IV	Yes	None	4
VII	Output from IV	Yes	None	4
VIII	One card each from MUS, NOTE, KLANG, KORN, and LONG. Repeat until all cards are used. After the first LONG, insert MOE, JOE, KOE, and LOE.	Yes	IX	1
IX	MUS, NOTE, KLANG, KORN, and LONG. Output from VIII	Yes	X	2
X	Output from IX	Yes	None	3

Figure 17. Master plan for data movement.

Webern's String Quartet, Opus 9, Third Movement (Eight voices)
Eight, rather than four, parts are reserved because of the many
double stops in each instrument. Refer to pages 295 to 340.

<u>Array names</u>	<u>Instrument</u>	<u>Instrument's number</u>
MU	I violin	1
MUS	I violin	2
NO	II violin	3
NOT	II violin	4
KL	Viola	5
KLA	Viola	6
LO	Cello	7
LON	Cello	8

Figure 18. Classification of instruments.

Below is the INFO for thirty-three chords (eight measures).
One line of the table equals the information for one IBM card.

INFO

00121	00122	00213	00223	00311	00313	00314	00321	00323	00411	00412
00421	00422	00423	00513	00514	00521	00523	00611	00613	00622	00711
00714	00722	00723	00724	00811	00812	00814	00821	00822	00824	00911

Figure 19. Form for coding an irregular INFO and instruments.

PROGRAM NUMBER	INPUT Order of data	OUTPUT		ORDER OF EXE- CUTION
		PRINT	PUNCHED CARDS TO:	
I	MU, MUS, NO, NOT, KL, KLA, LO, and LONG	Yes	III; Remove last 3 cards for Va.	1
III	Output from I	Yes	IV	2
IV	Output from III	Yes	OMITT	3

PROGRAM NUMBER	INPUT Order of data	OUTPUT		ORDER OF EXE- CUTION
		PRINT	PUNCHED CARDS TO:	
Va	Last card from Output I	None	V	2
OMITT	Output from IV	No	IIIa	4
IIIa	Output from OMITT	No	V	5
V VI VII	The information from OMITT and IIIa is processed through Programs, V, VI, and VII of the Hindemith <u>Quintet</u> .			
VIII	One card each from MU, MUS Yes NO, NOT, KL, KLA, LO, and LON. After the first LON insert MOE, JOE, KOE, and LOE.	IX		1
IX X	Any four voices may be chosen for melodic analysis along with the Hindemith roots and processed through the <u>String Quartet</u> programs IX and X.			

Figure 20. Master plan for data movement.

Mozart's Symphony in G Minor, No. 40

(Thirteen voices) Pages 236 to 294.

<u>Array names</u>	<u>Instrument</u>	<u>Instrument's number</u>
MU	Flute	1
MUS	I oboe	2
MUSI	II oboe	3
MUSIC	I clarinet	4
NO	II clarinet	5
NOT	I bassoon	6
NOTE	II bassoon	7
KL	I horn	8
KLA	II horn	9
LO	I violin	10
LON	II violin	11
LONG	Viola	12
LONGA	Bass	13

Figure 21. Classification of instruments.

Below is the INFO for twelve chords (four measures).

Two lines of the table equal the information for one IBM card.

INFO

00111	00121	00131	00211	00221	00231
00311	00321	00331	00411	00421	00431

Figure 22. Form for coding INFO.

INFO has five digits, and the instruments require six digits.

Figure 23. Form for coding instruments.

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
	Order of data	PRINT	PUNCHED CARDS TO:	
I	MU, MUS, MUSI, MUSIC, NO, NOT, NOTE, KL, KLA, LO, LON, LONG, and LONGA. All of each array together.	Yes	III; Remove last card for Va.	1
III	Output from I	Yes	IV and V	2
IV	Output from III	Yes	VI and VII	3
Va	Last card from Output I	None	V	2
OMITT	Output from IV	No	IIIa	4
IIIa	Output from OMITT	No	V	5
V VI VII	The information from OMITT and IIIa is processed through Programs V, VI, and VII of the Hindemith <u>Quintet</u> .			
VIII	MU, MUS, MUSI, MUSIC, NO, NOT, NOTE, KL, KLA, LO, LON, LONG, and LONGA. After the first LONGA insert MOE, JOE, KOE, LOE.	Yes	IX	1
IX X	Any four voices may be chosen for melodic analysis along with the Hindemith roots and processed through the <u>String Quartet</u> programs IX and X.			

Figure 24. Master plan for data movement.

Stravinsky's Symphony in Three Movements

(Twenty-two voices) Pages 376 to 439.

<u>Array names</u>	<u>Instrument</u>	<u>Instrument's number</u>
MUSA	I flute	1
MUSB	II flute	2
MUSC	I oboe	3
MUSD	II oboe	4
MUSE	I clarinet	5
MUSG	II clarinet	6
MUSH	I bassoon	7
MUSI	II bassoon	8
KORA	I horn	9
KORB	II horn	10
KORC	III horn	11
KORE	IV horn	12
NOTA	Piano	13
NOTB	Piano	14
NOTC	Piano	15
NOTR	Harp	16
NOTP	Harp	17
LONA	I Violin	18
LONB	II Violin	19
LONC	Viola	20
LOND	Cello	21
LONE	Bass	22

Figure 25. Classification of Instruments.

On the next page is the INFO for sixteen chords (two measures). Two lines of the table equal the information for one IBM card.

INFO

00111	00112	00113	00114	00121	00122	00123	00124
00211	00212	00213	00214	00221	00222	00223	00224

Figure 26. Form for coding INFO.

INFO has five digits, and the instruments require six digits followed by four spaces.

----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----

Figure 27. Form for coding instruments.

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
	Order of data	PRINT	PUNCHED CARDS TO:	
I	MUSA, MUSB, MUSC, MUSD, MUSE, MUSG, MUSH, MUSI, KORA, KORB, KORC, KORE, NOTA, NOTB, NOTC, NOTR, NOTP, LONA, LONB, LONC, LOND, and LONE. All of each array together	Yes	III; Remove last 2 cards for Va	1
III	Output from I	Yes	IV and V	2
IV	Output from III	Yes	VI and VII	3
Va	Last two cards from Output I	None	V	2
OMITT	Output from IV	No	IIIa	4
IIIa	Output from OMITT	No	V	5
V VI VII	The information from OMITT and IIIa is processed through Programs V, VI, and VII of the Hindemith Quintet.			

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
	Order of data	PRINT	PUNCHED CARDS TO:	
VIII	MUSA, MUSB, MUSC, MUSD, MUSE, MUSG, MUSH, MUSI, KORA, KORB, KORC, KORE, NOTA, NOTB, NOTC, NOTR, NOTP, LONA, LONB, LONC, LOND, and LONE. MOE, JOE, KOE, LOE.	Yes	IX	1
IX X	Any four voices may be chosen for melodic analysis along with the Hindemith roots and processed through the <u>String Quartet</u> programs IX and X.			

Figure 28. Master plan for data movement.

Schumann's piano composition, Papillons, Opus 2

(Eight voices) Pages 342 to 375.

<u>Array name</u>	<u>Instrument</u>
LUM	Piano

Figure 29. Classification of instruments.

Below is the INFO for fifteen chords (five measures).

Two lines of the table equal the information for one IBM card.

INFO

00111	00121	00131	00211	00221	00231	00311	00321
00331	00411	00421	00431	00511	00521	00531	-----

Figure 30. Form for coding INFO

Each chord is coded by starting with the top note and reading down. If spaces remain in the table, fill them each with four zeros and the number of the column. For instance, if a chord contains only four notes, the line might read:

50751 41042 40343 30334 00004 00006 00007 00008.

Fifteen chords, five measures. One line of the table equals the information for one IBM card.

[illegible]

Figure 31. Form for coding the piano part.

PROGRAM NUMBER	INPUT	OUTPUT		ORDER OF EXE- CUTION
	Order of data	PRINT	PUNCHED CARDS TO:	
I	LUM	Yes	I; Remove last card for Va.	1
II	LUM	Yes	V and VIII and IX	2
III	Output from I	Yes	IV	3
IV	Output from III	Yes	OMITT	4
Va	Last card from Output I	None	V	4
OMITT	Output from IV	No	IIIa	4
IIIa	Output from OMITT	No	V	5
V VI VII	The information from OMITT and IIIa is processed through Programs V, VI, and VII of the Hindemith <u>Quintet</u> .			
IX X	Any four voices may be chosen for melodic analysis and processed through the <u>String Quartet</u> programs IX, X.			

Figure 32. Master plan for data movement.

CHAPTER V

AN APPLICATION OF COMPUTER LOGIC TO A CREATIVE PROBLEM IN MUSIC

The first and longer problem of this research has been to develop a method of musical analysis using the electronic computer. The other problem is to employ the electronic computer to provide a basis for finding all the possible symmetrical twelve-tone rows through the development of one particular symmetrical twelve-tone row.

A symmetrical twelve-tone row is formed when the twelve different notes present in a chromatic scale are arranged so that the intervals between each of the last six notes are in reverse order to that of the intervals between each of the first six notes.

The serial composer is interested in tone rows with maximum permutation possibilities. The patterned rows, such as a symmetrical twelve-tone row, seem to possess inherently this quality of permutability. The reason this is true is that twelve-tone music must be logically conceived, and an accent on patterns in the row makes the logic more perceivable to the listener.

In this chapter there are four programs written to explore all the possible twelve-tone row combinations and to

process one such combination. Program 56T is designed to find all the possible sets of notes that may be present in the first six notes of a tone row. Only the rows beginning with the note C (represented by the number 1) are investigated. All other possibilities are the eleven transpositions of each of these rows. The results of Program 56T show 462 basic sets on pages 441 to 455. There is no more use for this program.

None of the 462 basic sets of numbers resulting from Program 56T contains exactly the same set of numbers. Even when C is kept as the first note of all the rows, there are yet 120 different combinations or arrangements of each of the 462 basic sets. This means that there are 55,440 different possible sequences for the first six notes of a tone-row. It is not possible within the scope of this research paper to investigate all 55,440 sets of notes to find how many of these have one or more symmetrical tone-row possibilities; however, the four programs of this chapter are designed for use in a research problem such as this.

Program Retro-A is set up to find 120 arrangements of any of the 462 basic sets. From arrangements of the second half of the row, Program Retro-Kompar discovers and prints out all the possible solutions for any one set. The printed output of the program is given in sets of six numbers. These six numbers are reversed and matched to the set of six notes that is being investigated to form the symmetrical tone-row. To discover if there is an arrangement of the second half of

a row which will, with the first half, form a symmetrical twelve-tone row, Program Retro-Kompar is used in conjunction with Retro-A.

Program Retro-H prints out the remaining six numbers that complete any twelve-tone row. This program is practical only if a large quantity of rows are processed.

An example will help to clarify the procedures used in developing one symmetrical twelve-tone row. The first six numbers (notes) of a row are given. For example, let the numbers be 1, 4, 6, 2, 5, and 9. The remaining numbers of the row are 3, 7, 8, 10, 11, and 12.

Program Retro-A is processed six times. Each time the first figure of the second half of the row is a different member of the group 3, 7, 8, 10, 11, and 12. It is immaterial in what order the other numbers follow. The arrangement may be as follows:

```

3,7,8,10,11,12
7,8,10,11,12,3
8,10,11,12,3,7
10,11,12,3,7,8
11,12,3,7,8,10
12,3,7,8,10,11.

```

To prepare the data each time for Retro-A, three Fortran Statement cards are necessary with one particular set of numbers arranged as shown in the following table:

Card 1	A	B	C	D	E	F		Original order
Card 2	B	C	D	E	F			Rearrangement
Card 3	C	D	E	F	B	C	E	C Rearrangement

Figure 33. Order of numbers.

In the table below the first set of numbers are for Card 1. On cards 2 and 3 substitution of numbers for letters is made by referring to the relationships established on Card 1.

Card 1	A	B	C	D	E	F		
	3	7	8	10	11	12		
Card 2	B	C	D	E	F			
	7	8	10	11	12			
Card 3	C	D	E	F	B	C	D	E
	8	10	11	12	7	8	10	11

Figure 34. Substitution of numbers for letters.

In punching data for Retro-A, one-digit numbers are preceded by two spaces, two-digit numbers by one space. The combined six sets of results obtained from Retro-A comprise 720 arrangements of 3, 7, 8, 10, 11, and 12.

In Retro-Kompar intervals between members of the row are found by subtracting successive numbers. The intervals present between 1, 4, 6, 2, 5, and 9 are 3, 2, 4, 3, and 4. In this program it is determined whether this same set of intervals is present in any one of the 720 arrangements of 3, 7, 8, 10, 11, and 12. Program Retro-Kompar is run six times, also. Each time, 120 cards from Retro-A are followed by a card punched with the desired intervals. The spacing for the intervals on a Fortran Statement card is:

---3---2---4---3---4.

This card is used each of the six times following the sets of 120 cards. As in the analysis programs, a blank card is included between the END card of the instructions and the beginning data card.

There is only one arrangement of this particular tone row that will give the desired intervals. It is found in the fifth set of numbers; therefore, only the fifth trial has a print-out. The results are 7, 10, 8, 12, 3, and 11. A symmetrical twelve-tone row of 1, 4, 6, 2, 5, 9, 11, 3, 12, 8, 10, and 7 is formed by reversing the order of the results of the program and attaching it to the first half of the row.

Program 56T is found on pages 440 to 455, Program Retro-H on pages 456 and 457, Program Retro-A on pages 458 to 464, and Program Retro-Kompar on pages 465 to 471.

There are many other interesting twelve-tone row combinations that could be investigated with the aid of the electronic computer. For instance, tone-rows in which all eleven intervals are prescribed, or in which each interval becomes progressively larger or smaller, could be found. It is entirely possible that a study of tone-row permutations could be made. Thus it can be seen that the computer can be of immense aid to the serial composer by providing tone-rows built to his specifications.

CHAPTER VI

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The purpose of this study has been to show how an electronic computer may be used to analyze music, and to develop a symmetrical twelve-tone row for the use of the serial composer. To introduce the problem a very brief historical summation of the history of analysis was made. After pointing out the need for a speedier method of analysis and one that could be used for the music of any period, a method using the electronic computer was proposed.

The chapter on methodology included instructions as to programming and coding and an example to show how both are done. Of the various phases of a musical analysis the following were processed through a computer:

- 1) Harmonic Intervals, with rhythmic, octave, and instrument locations
- 2) Frequency of Chord Repetition
- 3) Repeated Chord Progressions
- 4) Hindemith Roots
- 5) Melodic Intervals (also for Hindemith roots)
- 6) Frequency of Melodic Intervals (also for Hindemith roots).

Each of these was handled in one or more specific programs listed as Programs One, Two, Three, etc. What each program accomplishes was outlined in detail. To show the applicability of computer analysis to various styles and a varying number of voices (from four to twenty-two), several compositions from different historical periods were processed, utilizing the ten basic programs.

The card decks with instructions for their use are on file with the librarian of the Computer Center at Michigan State University. Chapter Four explains the method of preparation of the music and how to use the data within the various programs.

An application of computer logic to a creative problem in music was described. To provide the serial composer with a desired kind of patterned row, it was necessary to sort out the correct solution from among the many, many possibilities. The electronic computer was the obvious choice for this involved, mathematical computation. Using this machine a symmetrical twelve-tone row was built: 1, 4, 6, 2, 5, 9, 11, 3, 12, 8, 10, 7. This experiment also prepares the way for building other patterned rows using the electronic computer.

In a search for the most satisfactory method of analysis there are two qualities which are obviously desirable: maximal speed and universal applicability. Roman numeral analysis is not compatible with contemporary compositional practices, but an analysis of intervals with Arabic

numerals may be applied to the vertical sonorities of music of all periods. Since the outstanding characteristic of the electronic, digital computer is its ability to compute Arabic numerals at a high rate of speed and with a high degree of accuracy, the major conclusion of this study is that musical analysis may be most easily, quickly, and accurately done on a high-speed electronic digital computer. In addition the electronic computer codifies the results in a readable and meaningful form.

It is quite possible that a trained analyst could take a movement from a string quartet and write a harmonic analysis below the chords as fast as one could code the material needed for computer analysis. It would not be possible for him, however, in the same amount of time to put the material into tables showing the harmonic structure, the lowest-sounding note, the chord's location, instrumentation and octave registration, to make a listing of the roots as found by the Hindemith method, to locate all repeated chords and chord progressions, and to present the frequency of melodic intervals used in any voice as well as a record of upward and downward root movement.

The high speed of the digital computer enables it to solve with ease problems in selection. For this reason it is to be concluded that creative problems in music which involve many choices and rejections afford an opportunity for the use of the computer.

Many other musical experiments are possible with the aid of the electronic computer. After completion of the present study, it now seems possible that a harmonic analysis could be accomplished which would give results in traditional Roman numerals. It is also probable that an extensive program could be devised that would locate and identify non-harmonic tones in a particular style of composition. Both of these programs, Roman numeral analysis and an analysis of non-harmonic tones, can be built upon the foundations laid in this research.

Since some work is being done in the field of literature concerning authorship identification, a similar procedure could probably be worked out in music.

One of the tasks in which the computer excels because of its speed is classification of data. It would seem, therefore, that a codification of contemporary techniques is now possible. This would be invaluable to both musicologists and composers.

A comparative study of the styles of different composers is easily within the realm of possibility.

On an analog computer it may be possible to show line plots of melodies for classification and comparison.

It is difficult to keep pace with a field which is progressing and expanding at such a rapid rate, but the serious musician will want to be continually alert to the many possible applications of the electronic computer to music.

CHAPTER VII COMPUTER PROGRAMS WITH THEIR RESULTS

```

*   GILBERT ROLLER - QUARTET - NO. 1 - INTERVALS
    DIMENSION      MUS(3,80),NOTE(3,80),KLANG(3,80),
      1LONG(3,80),INT(80),INTE(80),INTER(80),LUM(8),KORD(80)
    95 FORMAT (1H0,8(I4))
    96 FORMAT (8(I4))
    300 FORMAT (9HINTERVALS)
      PRINT 300
      LRETET = 1
    2 FORMAT (16(I3,11,I1))
    1 READ 2,((MUS(I,N),I=1,3),N=1,80)
      READ 2,((NOTE(I,N),I=1,3),N=1,80)
      READ 2,((KLANG(I,N),I=1,3),N=1,80)
      READ 2,((LONG(I,N),I=1,3),N=1,80)
      M=0
      J=1
    10 M=M+1
      N=1
      I=1
      LUM(N)=      MUS(I,M)
      N=N+1
      LUM(N)=      NOTE(I,M)
      N=N+1
      LUM(N)=      KLANG(I,M)
      N=N+1
      LUM(N)=      LONG(I,M)
      L=N-3
    3 IF (LUM(N)-0) 12,4,12
    4 N=N-1
      IF (N-L) 12,12,11
    11 GO TO 3
    12 L=N-L
    5 IF (LUM(N)-LUM(I)) 7,7,6
    6 KE=LUM(N)
      KA= LUM(I)
      LUM(N)=KA
      LUM(I)=KE
    7 I=I+1

```



```

L=L-1
IF (L-0) 9,9,8
8 GO TO 5
9 I=1
91 IF (LUM(I)-0) 93,93,92
92 KORD(I)=LUM(I)-LUM(N)
GO TO 94
93 KORD(I)=0
94 IF (I-3) 105,106,106
105 I=I+1
GO TO 91
106 PRINT 95, (KORD(I),I=1,3)
PUNCH 96, (KORD(I),I=1,3)
98 J=J+1
IF (J-81) 102,103,103
102 GO TO 10
103 GO TO 104
104 IF (LRETET-5) 110,110,111
110 LRETET = LRETET +1
GO TO 1
111 GO TO 112
112 STOP
END

50151501515025141041410414104141041409414084141041409414074140541407414094140941
409414124141041410414064140741410414084140641406415065150751507515055150451
5045150851505515055150451504515055150551505515035150351502515015141241
412414034140241411414141241412415025150351412415025150351505515055150351
5035150251502515055150751507515075151251509515075150551505515045150951
000000 0
40642406424074240742403424034240242312324032402423123231032312324024240142
5045230942309324094250152501525025241242411425025241242310323093231032404240642
4034231232311324024240342512323122231132311323052310323123240242402424231232
31132311323123240242403424114241242501515015150252410424114240942408424094250452
000000 0

```


103	103	0	103	103	0
199	92	0	199	92	0
196	93	0	196	93	0
103	7	0	103	7	0
195	191	0	195	191	0
194	191	0	194	191	0
194	192	0	194	192	0
196	103	0	196	103	0
196	104	0	196	104	0
195	192	0	195	192	0
196	103	0	196	103	0
196	3	0	196	3	0
198	4	0	198	4	0
195	3	0	195	3	0
192	95	0	192	95	0
110	104	7	110	104	7
106	97	3	106	97	3
97	6	103	97	6	103
95	4	104	95	4	104
104	95	100	104	95	100
108	100	104	108	100	104
109	9	5	109	9	5
109	9	4	109	9	4
200	9	3	200	9	3
291	100	91	291	100	91
109	6	3	109	6	3
200	8	3	200	8	3
291	100	91	291	100	91
295	192	92	295	192	92
293	190	91	293	190	91
203	100	3	203	100	3
201	10	5	201	10	5
201	9	5	201	9	5
197	6	3	197	6	3
200	9	4	200	9	4
203	100	3	203	100	3
295	191	91	295	191	91
205	109	3	205	109	3

QUARTET - INTERVALS - NO.1 - RESULTS

106	8	103	106	8	103
104	7	100	104	7	100
203	194	96	203	194	96
200	192	95	200	192	95
107	103	3	107	103	3
103	6	3	103	6	3
203	195	97	203	195	97
200	192	95	200	192	95
106	8	103	106	8	103
103	7	100	103	7	100
96	3	0	96	3	0
100	96	0	100	96	0
106	97	2	106	97	2
108	8	3	108	8	3
504	409	412	504	409	412
505	405	409	505	405	409
105	8	3	105	8	3
109	9	3	109	9	3
502	408	405	502	408	405
504	408	404	504	408	404
508	411	402	508	411	402
509	404	312	509	404	312
509	405	402	509	405	402
509	412	404	509	412	404
204	107	100	204	107	100
204	107	104	204	107	104
298	7	104	298	7	104
297	4	104	297	4	104
207	104	104	207	104	104
210	100	103	210	100	103
204	101	98	204	101	98
295	101	98	295	101	98
205	100	97	205	100	97
204	100	95	204	100	95

601 307 0
601 306 309
602 309 0
602 202 306
512 206 302
304 296 196
207 502 310
300 204 104
511 407 205
307 203 103
205 96 0
300 192 92
510 303 312
206 109 106
305 410 407
312 409 405
404 407 0
404 409 412
405 407 410
107 0 4
199 5 2
292 97 93
295 95 91
401 107 104
292 196 100
292 196 96
292 195 95
204 198 94
204 200 95
291 194 96
200 192 95

601 307 0
601 306 309
602 309 0
602 202 306
512 206 302
304 296 196
207 502 310
300 204 104
511 407 205
307 203 103
205 96 0
300 192 92
510 303 312
206 109 106
305 410 407
312 409 405
404 407 0
404 409 412
405 407 410
107 0 4
199 5 2
292 97 93
295 95 91
401 107 104
292 196 100
292 196 96
292 195 95
204 198 94
204 200 95
291 194 96
200 192 95

205	105	97
204	195	95
205	100	97
297	105	100
295	104	98
305	205	197
209	203	5
206	197	2
208	108	3
300	109	5
208	203	3
205	197	5
297	197	6
302	110	10
300	104	7
200	107	198
200	105	8
200	104	7
295	203	100
295	295	103
295	293	102
295	203	100
200	104	7
200	107	98
200	105	8
200	103	7
292	195	100
295	192	98
293	193	96
293	196	95
200	7	3
203	100	7
302	202	105

205	105	97
204	195	95
205	100	97
297	105	100
295	104	98
305	205	197
209	203	5
206	197	2
208	108	3
300	109	5
208	203	3
205	197	5
297	197	6
302	110	10
300	104	7
200	107	198
200	105	8
200	104	7
295	203	100
295	295	103
295	293	102
295	203	100
200	104	7
200	107	98
200	105	8
200	103	7
292	195	100
295	192	98
293	193	96
293	196	95
200	7	3
203	100	7
302	202	105

300 207 203
406 503 0
105 97 0
105 100 0
204 200 0
308 501 0
304 0 308
302 0 309
301 409 309
306 501 0
509 501 406
507 502 404
199 195 95
402 411 306
205 196 0
202 105 0
305 209 105
303 203 100
303 196 96
204 195 95
305 107 103
301 103 100
301 196 96
292 195 95
395 195 103
391 191 100
391 196 96
291 195 95
395 195 103
391 191 100
391 196 96
292 195 95
212 404 311

300 207 203
406 503 0
105 97 0
105 100 0
204 200 0
308 501 0
304 0 308
302 0 309
301 409 309
306 501 0
509 501 406
507 502 404
199 195 95
402 411 306
205 196 0
202 105 0
305 209 105
303 203 100
303 196 96
204 195 95
305 107 103
301 103 100
301 196 96
292 195 95
395 195 103
391 191 100
391 196 96
291 195 95
395 195 103
391 191 100
391 196 96
292 195 95
212 404 311

301	195	102
512	209	402
300	203	195
303	295	98
304	204	197
298	195	195
297	200	105
300	105	197
303	203	200
298	203	195
297	205	105
300	204	197
303	203	200
299	204	107
298	203	200
298	203	195
300	204	200
300	203	195
298	204	200
299	292	195
298	293	195
298	290	195
299	293	195
210	201	101
298	293	196
299	107	103
210	106	103
299	107	2
297	106	4
305	502	309
297	98	5
301	107	4
305	502	311
305	0	402
305	503	312

409	503	303	
410	412	303	
605	410	502	
609	410	402	
203	97	4	
207	103	7	
207	103	7	
200	96	3	
197	97	93	
197	94	91	
195	95	92	
194	97	91	
197	93	91	
196	96	3	
194	103	96	
192	100	95	
193	102	93	
195	100	92	
96	0	0	
98	0	0	
96	0	0	
94	0	0	
92	0	0	
2	0	0	
92	0	0	
95	0	0	
97	0	0	
98	0	0	
97	0	0	
93	0	0	
3	0	0	
3	0	0	
3	0	0	
93	0	0	
97	0	0	

409	503	303	
410	412	303	
605	410	502	
609	410	402	
203	97	4	
207	103	7	
207	103	7	
200	96	3	
197	97	93	
197	94	91	
195	95	92	
194	97	91	
197	93	91	
196	96	3	
194	103	96	
192	100	95	
193	102	93	
195	100	92	
96	0	0	
98	0	0	
96	0	0	
94	0	0	
92	0	0	
2	0	0	
92	0	0	
95	0	0	
97	0	0	
98	0	0	
97	0	0	
93	0	0	
3	0	0	
3	0	0	
3	0	0	
93	0	0	
97	0	0	

2	0	0	0
6	0	0	0
7	0	0	0
98	0	0	0
92	0	0	0
505	505	410	
100	96	3	
505	503	408	
504	503	407	
102	100	5	
104	100	7	
410	0	402	
410	0	312	
410	0	402	
408	0	403	
407	0	403	
405	0	403	
407	0	403	
408	0	402	
408	0	312	
408	0	311	
408	0	312	
407	0	402	
405	0	402	
405	0	402	
405	0	402	
407	0	402	
407	0	310	
312	0	310	
402	0	408	
403	0	408	
403	0	407	
403	510	407	

403 512 406
 403 602 405
 402 602 405
 402 507 403
 312 509 403
 312 510 402
 310 510 402
 505 310 402
 507 309 402
 509 309 401
 509 305 309
 197 100 4
 203 106 8
 203 106 9
 203 97 5
 203 100 7
 203 106 8
 203 103 7
 204 4 0
 203 97 3
 203 100 7
 205 197 103
 202 196 97
 204 197 100
 205 197 103
 204 197 104
 200 196 96
 202 197 194
 204 200 197
 392 298 295
 395 295 292
 393 296 293
 392 298 295
 300 203 107
 303 107 103

403 512 406
 403 602 405
 402 602 405
 402 507 403
 312 509 403
 312 510 402
 310 510 402
 505 310 402
 507 309 402
 509 309 401
 509 305 309
 197 100 4
 203 106 8
 203 106 9
 203 97 5
 203 100 7
 203 106 8
 203 103 7
 204 4 0
 203 97 3
 203 100 7
 205 197 103
 202 196 97
 204 197 100
 205 197 103
 204 197 104
 200 196 96
 202 197 194
 204 200 197
 392 298 295
 395 295 292
 393 296 293
 392 298 295
 300 203 107
 303 107 103

300 197 103
 297 197 104
 296 196 103
 203 197 6
 203 195 3
 197 194 3
 197 192 92
 196 194 3
 196 193 93
 296 198 103

96 0 0
 100 0 0
 95 0 0
 92 0 0
 195 98 0
 103 3 0
 108 6 0
 104 102 0
 107 104 0
 199 96 0
 200 96 0
 200 97 0
 293 98 0
 295 96 0
 202 93 0
 203 94 0
 295 196 0
 296 197 0
 203 196 0
 198 194 0
 198 193 0

300 197 103
 297 197 104
 296 196 103
 203 197 6
 203 195 3
 197 194 3
 197 192 92
 196 194 3
 196 193 93
 296 198 103
 0 0 0
 0 0 0
 0 0 0
 96 0 0
 100 0 0
 95 0 0
 92 0 0
 195 98 0
 103 3 0
 108 6 0
 104 102 0
 107 104 0
 199 96 0
 200 96 0
 200 97 0
 293 98 0
 295 96 0
 202 93 0
 203 94 0
 295 196 0
 296 197 0
 203 196 0
 198 194 0
 198 193 0

193	101	0
307	502	0
303	502	0
307	412	0
307	409	0
306	412	0
307	412	0
95	2	0
97	3	0
96	93	0
300	202	205
300	202	105
203	106	9
206	108	8
206	109	0
206	108	8

193	101	0
307	502	0
303	502	0
307	412	0
307	409	0
306	412	0
307	412	0
95	2	0
97	3	0
96	93	0
300	202	205
300	202	105
203	106	9
206	108	8
206	109	6
206	108	8

```

*   GILBERT ROLLER - QUARTET - NO. 1 - INTERVALS
    DIMENSION MUS(3.80),NOTE(3.80),KLANG(3.80),LONG(3.80),LUM(8),
    1KORD(80)
    95 FORMAT (1H0.8(I4))
    96 FORMAT (8(I4))
    300 FORMAT (1H0.9HINTERVALS)
    PRINT 300
    LETET =1
    20 FORMAT (8(I3.11.11))
    2 FORMAT (16(I3.11.11))
    1 READ 2,((MUS(I,N),I=1,3),N=1.80)
    READ 2,((NOTE(I,N),I=1,3),N=1.80)
    READ2,((KLANG(I,N),I=1,3),N=1.80)
    READ 2,((LONG(I,N),I=1,3),N=1.80)
    M=0
    J=1
    10 M=M+1
    N=1
    I=1
    LUM(N)=      MUS(I,M)
    N=N+1
    LUM(N)=      NOTE(I,M)
    N=N+1
    LUM(N)=      KLANG(I,M)
    N=N+1
    LUM(N)=      LONG(I,M)
    L=N-3
    3 IF (LUM(N)-0) 12.4.12
    4 N=N-1
    IF (N-L) 12.12.11
    11 GO TO 3
    12 L=N-L
    5 IF (LUM(N)-LUM(I)) 7.7.6

```



```

6 KE=LUM(N)
  KA= LUM(1)
  LUM(N)=KA
  LUM(1)=KE
7 I=I+1
  L=L-1
  IF (L-0) 9,9,8
8 GO TO 5
9 I=1
91 IF (LUM(1)-0) 93,93,92
92 KORD(1)=LUM(1)-LUM(N)
  LONG(1,M)=LUM(N)
  GO TO 94
93 KORD(1)=0
94 IF (I-3) 105,105,106
105 I=I+1
  GO TO 91
106 PRINT 95, (KORD(I),I=1,3)
  PUNCH 96, (KORD(I),I=1,3)
98 J=J+1
  IF (J-81) 102,103,103
102 GO TO 10
103 PUNCH 20,((LONG(I,N),I=1,3),N=1,80)
  STOP
  END
5015150151502514104141041410414084141041409414084141041409414074140541407414094140941
40941412414124141041410414064140741410414084140641406415065150751507515055150431
50451505515055150551504515045150551505515055150351502515015141241
412414034140241411414141241412415025150351412415025150351505515055150351
50351502515035150551507515075151251509515075150551505515045150951

```



```

000000
40642406424074240742403424034240342402423123240332402423123231032312324024240142
50452309423094240942501525015250252412424114250252412423103230932310324044240642
403423125231132402424034231232312323123230932310323123240242402424231232
31132311323123240242403424114241242501515015150252410424114240942408424094250452
000000 0
000000 0
40143401434024340243310333103331033309333073331033309333070030500307333093330933
30933409434114340743308333073330533303003063330533303330233303333053330733
307333083307333053330330533307334074340543404434024340243402434014340143
000000 0
000000 0
000000 0
30634306343073430734303343033430334302342122430334302342122421024212243023430234
3023430234302342122430234303343043430534307343083430934310343093430734

```


196	93	0
103	7	0
195	191	0
194	191	0
194	192	0
196	103	0
198	104	0
195	192	0
196	103	0
196	3	0
198	4	0
195	3	0
192	95	0
110	104	7
106	97	3
97	6	103
95	4	104
104	93	100
108	100	104
109	9	3
109	9	4
200	9	3
291	100	91
109	6	3
200	8	3
291	100	91
295	192	92
293	190	91
203	100	3
201	10	3
201	9	3
197	6	3

200	9	4
203	100	3
295	191	91
205	109	3
204	109	4
203	197	103
208	197	103
204	197	100
200	103	97
197	103	94
196	100	93
195	98	92
195	100	92
202	197	94
501005010050200502004100041000410004100040900		
4080041000409004070040500407004090040200		
4060040600407004070040300403004030040200		
3120040300402003120031000312004020030900		
4010030900309004020031000310003100030900		
3070031000309003070030500307003090030234		
3063430634307343073430334303343033430234		
2122430334302342122421024212243023430234		
30234303343023421224302343033430434		
3043430334307343083430934310343093430734		

110	104	7
106	97	3
97	6	103
95	4	104
104	95	100
108	100	104
109	9	5
109	9	4
200	9	3
291	100	91
109	6	3
200	8	3
291	100	91
295	192	92
293	190	91
203	100	3
201	10	5
201	9	5
197	6	3
200	9	4
203	100	3
295	191	91
205	109	3
204	109	4
203	197	103
208	197	103
204	197	100
200	103	97
197	103	94
196	100	93
195	98	92
195	100	92
202	197	94

```

*      GILBERT ROLLER - QUARTET - REDUCTION - NO. 3
      DIMENSION INT(80),INTE(80),INTER(80),KO(80),KOR(80),KORT(80)
      4 FORMAT (8(14))
      1 FORMAT (8(14))
      READ 1, (INT(I),I=1,80)
      READ 1, (INTE(I),I=1,80)
      READ 1, (INTER(I),I=1,80)
      M=1
      N=12
      50 DO 83 I=1,80
      L=1
      IF (M-1) 82,81,82
      81 IF (INT(I)) 90,91,91
      90 INT(I)=(-1)*INT(I)
      91 J=INT(I)
      IF (J-N) 60,80,87
      80 KO(I)=INT(I)
      GO TO 76
      82 IF (M-2) 84,83,84
      83 IF (INTE(I)) 92,93,93
      92 INTE(I)=(-1)* INTE(I)
      93 J=INTE(I)
      IF (J-N) 81,81,87
      81 KOR(I)=INTE(I)
      GO TO 76
      84 IF (M-3) 77,85,78
      85 IF (INTER(I)) 94,95,95
      94 INTER(I)=(-1)* INTER(I)
      95 J=INTER(I)
      IF (J-N) 82,82,87
      82 KORT(I)=INTER(I)
      GO TO 76
      87 K=0

```

```

47 N=N+88
   K=K+88
   IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 IF (M-1) 72,71,72
71 KO(I)=INT(I)-K
72 IF (M-2) 74,73,74
73 KOR(I)=INTE(I)-K
74 IF (M-3) 76,75,76
75 KORT(I)=INTER(I)-K
76 N=12
85 CONTINUE
   M=M+1
   N=12
   IF (M-4) 77,78,78
77 GO TO 50
78 PUNCH 4, (KO(I),I=1,80)
   PUNCH 4, (KOR(I),I=1,80)
   PUNCH 4, (KORT(I), I=1,80)
   STOP
   END

```

0	0	0	0	0	0		
0	0	0	0	0	0		7
3	6	5	3	7	3	4	8
96	3	4	194	197	195	103	195

103	199	196	103	195	194	194	196
193	195	196	196	198	195	192	110
106	97	95	104	108	109	109	200
291	109	200	291	295	293	203	201
201	197	200	203	295	205	204	203
203	204	200	197	196	195	195	202
0	0	0	0				
0	0	0	0	0			
0	0	0	0	0			
0	0	0	0	0			92
103	0	0	7	191	191	192	103
104	192	103	3	4	3	95	104
97	6	4	95	100	9	9	9
100	6	8	100	192	190	100	10
9	6	9	100	191	109	109	197
197	197	103	103	100	98	100	197
0	0	0	0	0			
0	0	0	0	0			
0	0	0	0	0			
0	0	0	0	0			
0	0	0	0	0			
0	0	0	0	0			7
3	103	104	100	104	5	4	3
91	3	3	91	92	91	3	5
5	3	4	3	91	3	4	103
103	100	97	94	93	92	92	94

* GILBERT ROLLER - QUARTET - REDUCTION - NO. 3 RESULTS

0 0 3 8 3 10 6 3 1 8 0 0 0 0
 0 0 6 3 7 9 9 9 9 4 0 0 0 0
 0 0 0 5 4 8 8 7 12 12 12 0 0 0
 0 0 0 3 6 3 8 4 3 3 9 0 0 0
 0 0 0 7 9 7 10 8 7 7 8 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7 8 7 8 10 12 1 3 2
 4 3 6 4 9 3 4 7
 3 7 6 7 9 5 5 7

0 0 4 9 12 9 9 0 0 0 0 0 0
 0 0 0 6 6 9 9 3 0 0 0 0 0 0
 0 0 3 4 3 12 12 3 0 0 0 0 0 0
 0 7 3 7 12 4 5 12 3 0 0 0 0 0
 0 3 4 12 9 2 9 10 12 12 9 9
 4 3 4 9 10 9 9
 7 3 5 3 6
 4 3 4 4
 5 3 3 4

```

*      GILBERT ROLLER - QUARTET - SORTING - NO.4
      DIMENSION INT(80), INTE(80), INTER(80)
      49 FORMAT (3(14))
      1  FORMAT (8(14))
      READ 1, (INT(I),I=1,80)
      READ 1, (INTE(I), I=1,80)
      READ 1, (INTER(I),I=1,80)
      DO 30 I=1,80
      M1=INT(I)
      M2=INTE(I)
      M3=INTER(I)
      DO 30 K=1,3
      KORTS=0
      IF(KORTS-M1) 12,13,13
      12 KORTS=M1
      13 IF(KORTS-M2) 14,15,15
      14 KORTS=M2
      15 IF(KORTS-M3) 16,17,17
      16 KORTS=M3
      17 IF(KORTS-M1) 19,18,19
      18 M1=0
      GO TO 23
      19 IF(KORTS-M2) 21,20,21
      20 M2=0
      GO TO 23
      21 IF(KORTS-M3) 23,22,23
      22 M3=0
      23 IF (K-1) 25,24,25
      24 KO=KORTS
      25 IF (K-2) 27,26,27
      26 KOR=KORTS
      27 IF (K-3) 30,28,30
      28 KORT=KORTS
      PUNCH 49, KO,KOR,KORT

```

30 CONTINUE
 STOP
 END

```

0 0 3 8 3 10 6 3 1 8 0 0 0 0 0 3 4 9 12 9 9 0 0 0 0 0 0 0 0 0 0 3 3 5 3
0 0 6 3 7 7 9 9 9 4 0 0 0 0 0 4 4 6 6 6 9 0 0 0 0 0 0 0 0 0 0 3 3 5 3
0 0 5 4 3 5 7 12 12 12 0 0 0 0 0 5 3 4 3 3 3 0 0 0 0 0 0 0 0 0 0 4 3 3 12
0 0 3 6 3 3 8 4 3 3 9 0 0 0 0 0 7 3 7 12 12 12 3 0 0 0 0 0 0 0 0 12 3 3 6
0 0 7 9 7 10 8 7 7 3 0 0 0 0 0 3 4 12 4 3 12 12 0 0 0 0 0 0 0 0 0 0 4 4 3 5
0 0 7 6 7 9 5 5 7 3 3 3 9 2 9 10 3 12 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
4 3 3 6 4 9 3 4 7 4 3 7 9 12 9 12 4 3 5 3 6
7 8 7 3 10 12 1 3 2 4 3 4 9 10 9 9 7 3 5 3 6

```

•

1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	3	4	4	4	4	5	4	3	3	
0	0	0	4	3	4	3	3	3	3	4	3	4	4	3	3	4	3	4	7	6	6	4	7	8	9	9	9	9	3						
9	7	3	7	3	7	5	7	7	6	6	8	10	7	8	8	10	7	7	10	9	9	7	12	12	9	9	12	12							

3	2	3	4	2	3	1	1	3	4	3	3	3	4	3	3	4	3	3	5	4	4	2	
6	9	3	4	3	3	5	5	6	9	3	3	5	4	3	8	9	9	6	8	7	7	6	
9	12	12	7	5	12	10	9	9	12	12	7	9	9	9	9	9	12	12	9	12	10	12	9

```

*   GILBERT ROLLER - QUARTET - REDUCTION OF LONG - NO. 5A
      DIMENSION LONG (3,80)
      2 FORMAT (16(13,11,11))
      READ 2, ((LONG (1,N),I=1,3),N=1,80)
      N=12
      50 DO 85 I=1,80
      L=1
      J=LONG(1,I)
      IF (J-N) 80,80,57
      80 LONG (1,I) = LONG (1,I)
      GO TO 76
      57 K=0
      47 N=N+88
      K=K+88
      IF (J-N) 58,58,59
      58 GO TO 70
      59 N=N+12
      K=K+12
      IF (J-N) 60,60,65
      60 GO TO 70
      65 IF (L-6) 66,70,70
      66 L=L+1
      GO TO 47
      70 LONG (1,I) = LONG (1,I) -K
      76 N=12
      85 CONTINUE
      PUNCH 2,((LONG (1,N),I=1,3),N=1,80)
      STOP
      END

```


000000
000000
000000 0
306343063430734303343033430334302342122430334302342122421024212243023430234
302343053430334302342122430234303343043430534307343083430934310343093430734

* GILBERT ROLLER - QUARTET - REDUCTION OF LONG - NO. 5A - RESULTS

000000
 000000
 000000 0
 006340063400734003340033400234012240033400234012240023400234
 00234005340033400234012240023400334004340043400534007340083400934010340093400734

```

* GILBERT ROLLER - QUARTET - PRINT OUT - NO.5
  DIMENSION INFO(2,80),MUS(3,80),NOTE(3,80),KLANG(3,80),LONG(3,80),
  IKORD(240)
  1 FORMAT (16(13,12))
  READ 1,((INFO(I,N),I=1,2),N=1,80)
  2 FORMAT (16(13,11,11))
  READ 2,((MUS(I,N),I=1,3),N=1,80)
  READ 2,((NOTE(I,N),I=1,3),N=1,80)
  READ 2,((KLANG(I,N),I=1,3),N=1,80)
  READ 2,((LONG(I,N),I=1,3),N=1,80)
  3 FORMAT (8(14))
  READ 3, (KORD(I),I=1,240)
  20 FORMAT (1H0,17,115,18X,2H0 ,113,110/)
  270 FORMAT (1H0,17,115,18X,2H2-,113,110/)
  271 FORMAT (1H0,17,115,18X,2H2.,113,110/)
  272 FORMAT (1H0,17,115,18X,2H3-,113,110/)
  273 FORMAT (1H0,17,115,18X,2H3.,113,110/)
  274 FORMAT (1H0,17,115,18X,2H4-,113,110/)
  275 FORMAT (1H0,17,115,18X,2H4+,113,110/)
  276 FORMAT (1H0,17,115,18X,2H5.,113,110/)
  277 FORMAT (1H0,17,115,18X,2H6-,113,110/)
  278 FORMAT (1H0,17,115,18X,2H6.,113,110/)
  279 FORMAT (1H0,17,115,18X,2H7-,113,110/)
  280 FORMAT (1H0,17,115,18X,2H7.,113,110/)
  281 FORMAT (1H0,17,115,18X,2H8.,113,110/)
  21 FORMAT (41X,2H0 ,113,110/)
  285 FORMAT (41X,2H2-,113,110/)
  286 FORMAT (41X,2H2.,113,110/)
  287 FORMAT (41X,2H3-,113,110/)
  288 FORMAT (41X,2H3.,113,110/)
  289 FORMAT (41X,2H4.,113,110/)
  290 FORMAT (41X,2H4+,113,110/)
  291 FORMAT (41X,2H5.,113,110/)
  292 FORMAT (41X,2H6-,113,110/)

```



```

293 FORMAT (41X,2H6,,113,110/)
294 FORMAT (41X,2H7-,113,110/)
295 FORMAT (41X,2H7.,113,110/)
296 FORMAT (41X,2H8.,113,110/)
22  FORMAT (41X,2H0 ,113,110//)
315 FORMAT (41X,2HC ,113,110//)
316 FORMAT (41X,2HC+ ,113,110//)
317 FORMAT (41X,2HD ,113,110//)
318 FORMAT (41X,2HE- ,113,110//)
319 FORMAT (41X,2HE ,113,110//)
320 FORMAT (41X,2HF ,113,110//)
321 FORMAT (41X,2HF+ ,113,110//)
322 FORMAT (41X,2HG ,113,110//)
323 FORMAT (41X,2HA- ,113,110//)
324 FORMAT (41X,2HA ,113,110//)
325 FORMAT (41X,2HB- ,113,110//)
326 FORMAT (41X,2HB ,113,110//)
350 FORMAT (1H0,4HCODE/4X,21H. IS MAJOR OR PERFECT/4X,10H- IS MINOR/
14X,14H+ IS AUGMENTED)
PRINT 350
42  FORMAT (1H0,13HMEASURE NO. ,19HRHYTHMIC POSITION ,
120HHARMONIC INTERVALS ,8HOCTAVE ,12HINSTRUMENT )
41  PRINT 42
J=160
DO 302 N=1,80
J=J-159
IF (KORD(J)-0) 4,5,4
5  PRINT 20,INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
4  IF (KORD(J)-2) 200,201,202
200 PRINT 270, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
201 PRINT 271, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217

```

```

202 IF (KORD(J)-4) 203,204,205
203 PRINT 272, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
204 PRINT 273, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
205 IF (KORD(J)-6) 206,207,208
206 PRINT 274, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
207 PRINT 275, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
208 IF (KORD(J)-8) 209,210,211
209 PRINT 276, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
210 PRINT 277, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
211 IF (KORD(J)-10) 212,213,214
212 PRINT 278, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
213 PRINT 279, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
214 IF (KORD(J)-12) 215,216,217
215 PRINT 280, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
216 PRINT 281, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
217 J=J+80
IF (KORD(J)-0) 6,7,6
7 PRINT 21, NOTE(2,N),NOTE(3,N)
GO TO 235
6 IF (KORD(J)-2) 218,219,220
218 PRINT 285, NOTE(2,N),NOTE(3,N)
GO TO 235
219 PRINT 286, NOTE(2,N),NOTE(3,N)
GO TO 235

```

```

220 IF (KORD(J)-4) 221,222,223
221 PRINT 287, NOTE(2,N),NOTE(3,N)
    GO TO 235
222 PRINT 288, NOTE(2,N),NOTE(3,N)
    GO TO 235
223 IF (KORD(J)-6) 224,225,226
224 PRINT 289, NOTE(2,N),NOTE(3,N)
    GO TO 235
225 PRINT 290, NOTE(2,N),NOTE(3,N)
    GO TO 235
226 IF (KORD(J)-8) 227,228,229
227 PRINT 291, NOTE(2,N),NOTE(3,N)
    GO TO 235
228 PRINT 292, NOTE(2,N),NOTE(3,N)
    GO TO 235
229 IF (KORD(J)-10) 230,231,232
230 PRINT 293, NOTE(2,N),NOTE(3,N)
    GO TO 235
231 PRINT 294, NOTE(2,N),NOTE(3,N)
    GO TO 235
232 IF (KORD(J)-12) 233,234,235
233 PRINT 295, NOTE(2,N),NOTE(3,N)
    GO TO 235
234 PRINT 296, NOTE(2,N),NOTE(3,N)
235 J=J+80
    IF (KORD(J)-0) 8,9,8
        9 PRINT 21, KLANG(2,N),KLANG(3,N)
            GO TO 253
        8 IF (KORD(J)-2) 236,237,238
236 PRINT 285, KLANG(2,N),KLANG(3,N)
    GO TO 253
237 PRINT 286, KLANG(2,N),KLANG(3,N)
    GO TO 253
238 IF (KORD(J)-4) 239,240,241

```



```

239 PRINT 287. KLANG(2,N).KLANG(3,N)
    GO TO 253
240 PRINT 288. KLANG(2,N).KLANG(3,N)
    GO TO 253
241 IF (KORD(J)-6) 242,243,244
242 PRINT 289. KLANG(2,N).KLANG(3,N)
    GO TO 253
243 PRINT 290. KLANG(2,N).KLANG(3,N)
    GO TO 253
244 IF (KORD(J)-8) 245,246,247
245 PRINT 291. KLANG(2,N).KLANG(3,N)
    GO TO 253
246 PRINT 292. KLANG(2,N).KLANG(3,N)
    GO TO 253
247 IF (KORD(J)-10) 248,249,250
248 PRINT 293. KLANG(2,N).KLANG(3,N)
    GO TO 253
249 PRINT 294. KLANG(2,N).KLANG(3,N)
    GO TO 253
250 IF (KORD(J)-12) 251,252,253
251 PRINT 295. KLANG(2,N).KLANG(3,N)
    GO TO 253
252 PRINT 296. KLANG(2,N).KLANG(3,N)
253 IF (LONG(1,N) -0) 10,11,10
    11 PRINT 22. LONG(2,N).LONG(3,N)
        GO TO 302
    10 IF (LONG(1,N)-2) 255,256,257
255 PRINT 315. LONG(2,N).LONG(3,N)
    GO TO 302
256 PRINT 316. LONG(2,N).LONG(3,N)
    GO TO 302
257 IF (LONG(1,N) -4) 258,259,260
258 PRINT 317. LONG(2,N).LONG(3,N)

```

```

GO TO 302
259 PRINT 318, LONG(2,N), LONG(3,N)
GO TO 302
260 IF (LONG(1,N) -6) 261,262,263
261 PRINT 319, LONG(2,N), LONG(3,N)
GO TO 302
262 PRINT 320, LONG(2,N), LONG(3,N)
GO TO 302
263 IF (LONG (1,N) -8) 264,265,266
264 PRINT 321, LONG(2,N), LONG(3,N)
GO TO 302
265 PRINT 322, LONG(2,N), LONG(3,N)
GO TO 302
266 IF (LONG(1,N) -10) 267,268,269
267 PRINT 323, LONG(2,N), LONG(3,N)
GO TO 302
268 PRINT 324, LONG(2,N), LONG(3,N)
GO TO 302
269 IF (LONG(1,N)-12) 327,328,302
327 PRINT 325, LONG (2,N), LONG(3,N)
GO TO 302
328 PRINT 326, LONG (2,N), LONG(3,N)
302 CONTINUE
STOP
END

```

```

00111001120012100122002110021200221002220031100312003210032200411004120042100422
00511005120052100522006110061200621006220071100712007210072200811008120082100822
0091100912009210092201011010120102101022011110111201120112201211012120122101222
01311013120132101322014110141201421014220151101512015210152201611016120162101622
01711017120172101722018110181201821018220191101912019210192202011020120202102022
501515015150251502514104141041410414104140541405414074140741409414094140941
40941412414124141041410414064140741410414084140641406415065150751507515055150451
504515085150551505515055150451504515045150551505515035150351502515015141241

```


* GILBERT R.- NO. 5 - PRINT OUT - RESULTS

CODE

. IS MAJOR OR PERFECT

- IS MINOR

/ IS AUGMENTED

MEASURE NO.	RHYTHMIC POSITION	HARMONIC INTERVALS	OCTAVE	INSTRUMENT
1	11	0	5	1
		0	0	0
		0	0	0
		0	0	0
1	12	0	5	1
		0	0	0
		0	0	0
		0	0	0
1	21	0	5	1
		0	0	0
		0	0	0
		0	0	0
1	22	0	5	1
		0	0	0
		0	0	0
		0	0	0

2	11	0 0 0 0	4 0 0 0	1 0 0 0
2	12	0 0 0 0	4 0 0 0	1 0 0 0
2	21	0 0 0 0	4 0 0 0	1 0 0 0
2	22	0 0 0 0	4 0 0 0	1 0 0 0
3	11	0 0 0 0	4 0 0 0	1 0 0 0
3	12	0 0 0 0	4 0 0 0	1 0 0 0
3	21	0 0 0 0	4 0 0 0	1 0 0 0
3	22	0 0 0 0	4 0 0 0	1 0 0 0

1 2 0 0 1 2 0 0 1 2 0 0 1 2 0 0 1 2 0 0 1 2 0 0 1 2 0 0

4 4 0 0 4 4 0 0 4 4 0 0 4 4 0 0 4 4 0 0 4 3 0 0 4 3 0 0

3-
0 0 0 0 5-
0 0 0 0 3-
0 0 0 0 3-
0 0 0 0 6-
0 0 0 0 6-
0 0 0 0 3-
0 0 0 0

22

11

12

21

22

11

12

5

6

6

6

6

7

7

7	21	3. 0 0 0	4 4 0 0	1 2 0 0
7	22	4. 0 0 0	5 3 0 0	1 2 0 0
8	11	6. 0 0 0	5 3 0 0	1 2 0 0
8	12	5. 0 0 0	5 3 0 0	1 2 0 0
8	21	3- 0 0 0	5 4 0 0	1 2 0 0
8	22	5. 3. 0 0	5 4 3 0	1 2 3 0
9	11	3- 3- 0 0	5 5 4 0	1 2 3 0

9	12	5. 3. 0 0	5 4 4 0	1 2 3 0
9	21	3- 4. 0 0	5 3 4 0	1 2 3 0
9	22	3- 5. 0 0	5 4 4 0	1 2 3 0
10	11	5. 3- 0 0	5 5 3 0	1 2 3 0
10	12	4- 3- 0 0	5 5 3 0	1 2 3 0
10	21	4- 3. 0 0	5 5 3 0	1 2 3 0
10	22	6- 3- 0 0	5 4 3 0	1 2 3 0

11	11	7-	5 4 3 0	1 2 3 0
11	12	5-	5 5 3 0	1 2 3 0
11	21	6-	5 4 3 0	1 2 3 0
11	22	6-	5 3 0 0	1 2 0 0
12	11	7-	5 3 0 0	1 2 0 0
12	12	5-	5 3 3 0	1 2 3 0
12	21	3-	5 4 3 0	1 2 3 0

12	22	7- 3. 5. 6/	4 4 3 3	1 2 3 4
13	11	4/ 6. 3- F	4 4 3 3	1 2 3 4
13	12	6. 4/ 3- F	4 3 4 3	1 2 3 4
13	21	5. 3. 3. F/	4 3 4 3	1 2 3 4
13	22	3. 5. 8. F/	4 4 4 3	1 2 3 4
14	11	6- 8. 3. D	4 4 4 3	1 2 3 4
14	12	6. 6. 4. D	4 3 3 3	1 2 3 4

14	21	6. 6. 3. D	4 3 3 3	1 2 3 4
14	22	8. 6. 3- C/	5 3 3 3	1 2 3 4
15	11	3- 8. 3- B	5 3 0 2	1 2 0 4
15	12	6. 4/ 3- D	4 3 3 3	1 2 3 4
15	21	8. 6- 3- C/	5 3 3 3	1 2 3 4
15	22	3- 8. 3- B	5 3 3 2	1 2 3 4
16	11	5. 3. 3. A	5 4 3 2	1 2 3 4

16	12	4. 2. 3- B	5 4 3 2	1 2 3 4
16	21	3- 8. 3- C/	5 4 3 3	1 2 3 4
16	22	2- 7- 4. C/	5 3 3 3	1 2 3 4
17	11	2- 6. 4. C/	5 3 3 3	1 2 3 4
17	12	6. 4/ 3- E	5 3 3 3	1 2 3 4
17	21	8. 6. 3. D	5 3 3 3	1 2 3 4
17	22	3- 8. 3- C/	5 4 3 3	1 2 3 4

18	11	5. 3- 3- B	5 4 3 2	1 2 3 4
18	12	4. 6. 3- C/	5 4 3 3	1 2 3 4
18	21	3. 6. 3- D	5 4 3 3	1 2 3 4
18	22	3- 6. 3- E-	5 5 4 3	1 1 3 4
19	11	6- 6. 3- E-	5 5 4 3	1 1 3 4
19	12	3. 6. 8- E	5 5 4 3	1 2 3 4
19	21	8. 3- 6- F/	5 4 4 3	1 2 3 4

19	22	6. 3- 4/ G	5 4 4 3	1 2 3 4
20	11	6- 8. 4. A-	5 4 4 3	1 2 3 4
20	12	5. 7- 3. A	5 4 4 3	1 2 3 4
20	21	5. 8. 3. A-	5 4 4 3	1 2 3 4
20	22	2. 6. 4/ F/	5 5 4 3	1 2 3 4

```

*   GILBERT ROLLER - QUARTET - CHORD REPETITION - NO.6
    DIMENSION KORD (3,80)
    47 FORMAT (1H0, 30H FREQUENCY OF CHORD REPETITION)
    PRINT 47
    48 FORMAT (1H0, 3(14))
    1  FORMAT (3(14))
    READ 1, ((KORD(I,N),I=1,3),N=1,80)
    J=1
    DO 40 N=1,80
      K1 = KORD (1,N)
      K2 = KORD (2,N)
      K3 = KORD (3,N)
      M1=K1
      M2=K2
      M3=K3
      J=J+1
      NEW =0
      DO 40 M = J,80
        L1 = KORD (1,M)
        L2= KORD (2,M)
        L3 = KORD (3,M)
        IF (M1-L1) 31,32,31
        31 GO TO 40
        32 IF (M2-L2) 31,34,31
        34 IF (M3-L3) 31,35,31
        35 IF (M1-0) 43,36,43
        36 IF (M2-0) 43,37,43
        37 IF (M3-0) 43,40,43
        43 IF (NEW) 44,44,156
        44 IF (K1-L2) 61,60,61
        60 K1=8
        GO TO 100
        61 IF (K1-11) 63,62,63

```


62 K1=7	
GO TO 100	
63 IF (K1-10)	65,64,65
64 K1=-7	
GO TO 100	
65 IF (K1-9)	67,66,67
66 K1=6	
GO TO 100	
67 IF (K1-8)	69,68,69
68 K1=-6	
GO TO 100	
69 IF (K1-7)	71,70,71
70 K1=5	
GO TO 100	
71 IF (K1-6)	73,72,73
72 K1=-5	
GO TO 100	
73 IF (K1-5)	75,74,75
74 K1=4	
GO TO 100	
75 IF (K1-4)	77,76,77
76 K1=3	
GO TO 100	
77 IF (K1-3)	79,78,79
78 K1=-3	
GO TO 100	
79 IF (K1-2)	81,80,81
80 K1=2	
GO TO 100	
81 IF (K1-1)	83,82,83
82 K1=-2	
GO TO 100	
83 IF (K1-0)	100,84,100
84 K1=1	

100 IF (K2-12)	102,101,102
101 K2=8	
GO TO 130	
102 IF (K2-11)	104,103,104
103 K2=7	
GO TO 130	
104 IF (K2-10)	106,105,106
105 K2=7	
GO TO 130	
106 IF (K2-9)	108,107,108
107 K2=6	
GO TO 130	
108 IF (K2-8)	110,109,110
109 K2=6	
GO TO 130	
110 IF (K2-7)	112,111,112
111 K2=5	
GO TO 130	
112 IF (K2-6)	114,113,114
113 K2=5	
GO TO 130	
114 IF (K2-5)	116,115,116
115 K2=4	
GO TO 130	
116 IF (K2-4)	118,117,118
117 K2=3	
GO TO 130	
118 IF (K2-3)	120,119,120
119 K2=3	
GO TO 130	
120 IF (K2-2)	122,121,122
121 K2=2	
GO TO 130	
122 IF (K2-1)	124,123,124

123 K2=-2	
GO TO 130	
124 IF (K2=0)	130,125,130
125 K2=1	
130 IF (K3=12)	132,131,132
131 K3=8	
GO TO 156	
132 IF (K3=11)	134,123,134
133 K3=7	
GO TO 156	
134 IF (K3=10)	136,135,136
135 K3=-7	
GO TO 156	
136 IF (K3=9)	138,137,138
137 K3=6	
GO TO 156	
138 IF (K3=8)	140,139,140
139 K3=-6	
GO TO 156	
140 IF (K3=7)	142,141,142
141 K3=5	
GO TO 156	
142 IF (K3=6)	144,143,144
143 K3=-5	
GO TO 156	
144 IF (K3=5)	146,145,146
145 K3=4	
GO TO 156	
146 IF (K3=4)	148,147,148
147 K3=3	
GO TO 156	
148 IF (K3=3)	150,149,150
149 K3=-3	

5 3 4 4 4 5 4 3 3 3 3 3 4 2 3 1 1 3 4 3 3 3 4 3 3 4 3 3 3 4 4 2
 6 6 4 7 8 9 9 9 9 9 9 9 4 3 3 5 5 5 9 3 3 5 4 3 3 9 5 6 3 7 7 6
 9 9 7 12 12 9 9 12 12 9 12 12 7 5 12 10 9 9 12 12 7 9 9 9 9 12 12 9 12 10 12 9

* GILBERT ROLLER - QUARTET - CHORD REPETITION - NO. 6

FREQUENCY OF CHORD REPETITION

5	1	1
5	1	1
-3	1	1
-3	1	1
-3	1	1
-3	1	1
-5	1	1
3	1	1
-6	1	1
5	1	1
5	1	1
5	1	1
5	1	1
5	1	1
5	1	1
-6	1	1
-6	1	1
-7	1	1
-7	1	1
6	3	3
6	-3	-3
6	-3	-3
5	-3	-3
8	3	3
8	3	3
8	-3	-3
8	-3	-3
8	-3	-3
6	-3	-3
8	3	3


```

*      GILBERT ROLLER - OP.7 NO.7
      DIMENSION KORD (3,80)
      4  FORMAT (1H0,4X,17MCHORD PROGRESSION)
      PRINT 4
      1  FORMAT (3(14))
      READ 1, ((KORD(1,N),1=1,3),N=1,80)
      41 FORMAT (1H0,3(14))
      5  FORMAT (1H0, 3HEND)
      L=1
      J=1
      K=1
      I=0
      M=0
      10 K1=KORD(1,J)
      K2=KORD(2,J)
      K3=KORD(3,J)
      L=L+1
      30 L=L+1
      L1= KORD (1,L)
      L2= KORD (2,L)
      L3=KORD(3,L)
      IF (K1-L1) 32,33,32
      32 IF (1-1) 37,20,39
      33 IF (K2-L2) 32,34,32
      34 IF (K3-L3) 32,35,32
      35 IF (K1-0) 53,46,53
      46 IF (K2-0) 53,47,53
      47 IF (K3-0) 53,37,53
      53 IF (K1-12) 61,60,61
      60 K1=8
      GO TO 100
      61 IF (K1-11) 63,62,63

```


62	K1=7	
	GO TO 100	
63	IF (K1-10)	65,64,65
64	K1=-7	
	GO TO 100	
65	IF (K1-9)	67,66,67
66	K1=6	
	GO TO 100	
67	IF (K1-8)	69,68,69
68	K1=-6	
	GO TO 100	
69	IF (K1-7)	71,70,71
70	K1=5	
	GO TO 100	
71	IF (K1-6)	73,72,73
72	K1=-5	
	GO TO 100	
73	IF (K1-5)	75,74,75
74	K1=4	
	GO TO 100	
75	IF (K1-4)	77,76,77
76	K1=3	
	GO TO 100	
77	IF (K1-3)	79,78,79
78	K1=-3	
	GO TO 100	
79	IF (K1-2)	81,80,81
80	K1=2	
	GO TO 100	
81	IF (K1-1)	83,82,83
82	K1=-2	
	GO TO 100	
83	IF (K1-0)	100,84,100

84 K1=1	
100 IF (K2-12)	102,101,102
101 K2=8	
GO TO 130	
102 IF (K2-11)	104,103,104
103 K2=7	
GO TO 130	
104 IF (K2-10)	106,105,106
105 K2=7	
GO TO 130	
106 IF (K2-9)	108,107,108
107 K2=6	
GO TO 130	
108 IF (K2-8)	110,109,110
109 K2=6	
GO TO 130	
110 IF (K2-7)	112,111,112
111 K2=5	
GO TO 130	
112 IF (K2-6)	114,113,114
113 K2=5	
GO TO 130	
114 IF (K2-5)	116,115,116
115 K2=4	
GO TO 130	
116 IF (K2-4)	118,117,118
117 K2=3	
GO TO 130	
118 IF (K2-3)	120,119,120
119 K2=3	
GO TO 130	
120 IF (K2-2)	122,121,122

121	K2=2	
	GO TO 130	
122	IF (K2-1)	124,123,124
123	K2=-2	
	GO TO 130	
124	IF (K2-0)	130,125,130
125	K2=1	
130	IF (K3-12)	132,131,132
131	K3=8	
	GO TO 156	
132	IF (K3-11)	134,133,134
133	K3=7	
	GO TO 156	
134	IF (K3-10)	136,135,136
135	K3=-7	
	GO TO 156	
136	IF (K3-9)	138,137,138
137	K3=6	
	GO TO 156	
138	IF (K3-8)	140,139,140
139	K3=-6	
	GO TO 156	
140	IF (K3-7)	142,141,142
141	K3=5	
	GO TO 156	
142	IF (K3-6)	144,143,144
143	K3=-5	
	GO TO 156	
144	IF (K3-5)	146,145,146
145	K3=4	
	GO TO 156	
146	IF (K3-4)	148,147,148
147	K3=3	
	GO TO 156	

```

148 IF (K3=3)      150.149.150
149 K3=-3
    GO TO 156
150 IF (K3=2)      152.151.152
151 K3=2
    GO TO 156
152 IF (K3=1)      154.153.154
153 K3=-2
    GO TO 156
154 IF (K3=0)      156.155.156
155 K3=1
156 PRINT 41, K1,K2,K3
    M=M+1
    GO TO 36
20 1=0
    JEJ-1
    L = L-1
    GO TO 21
36 JEJ+1
    I=I+1
21 K1=KURD(1,J)
    K2=KURD(2,J)
    K3=KURD(3,J)
37 IF (L=80) 38.39.39
38 GO TO 30
39 JEJ+1
    I = 0
    L=J
    IF (M=1) 7.0.0
    6 PRINT 5
    M=0
    7 IF (L=80) 40.42.42
    40 GO TO 10

```


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 1 3 4 4 4 5 4 3 3 3 3 3 4 2
 0 0 4 1 4 1 3 3 4 3 4 4 3 3 4 3 4 7 0 0 4 7 8 9 9 9 3 0 8 1 4 3
 7 3 7 3 7 3 7 7 0 0 8 10 7 8 8 10 7 7 10 9 9 7 12 12 9 9 12 12 9 12 12 7 5

* GILBERT ROLLER - CP. 7 NO. 7

CHORD PROGRESSION

5 1 1
-3 1 1

END

-3 1 1
-3 1 1
-3 1 1

END

5 1 1
-3 1 1

END

-3 1 1

END

5 3 1
5 3 1
5 3 1

END

5 3 1
5 3 1

```

END      5  -3  1
END
END      5  -3  1
END
-6  -3  1
-6  -3  1
-7  -3  1
END
-7  5  3
END
6  -5  -3
6  -5  -3
END
6  -5  -3
6  -5  -3
END
5  3  3
END

```

8 5 3

END

8 6 -3

END

8 -3 -3
8 -3 -3
8 -3 -3

END

6 -5 -3

END

8 -3 -3
8 -3 -3

END

8 -3 -3

END

6 4 -3

END

8 6 3

END


```

*   GILBERT ROLLER - QUARTET - HINDEMITH ROOTS - NO.8
    DIMENSION MUS(2,16),NOTE(2,16),KLANG(2,16),LONG(2,16),MOE(13),
     1JOE(7),KOE(5),LOE(1),INT1(16),INT2(16),INT3(16),INT4(16),INT5(16),
     2INT6(16),KORT1(16),KORT2(16),KORT3(16),KORT4(16),KORT5(16),
     3KORT6(16),MITH(16)
    131 FORMAT (1H0, 16(I4))
    132 FORMAT (16(I4))
     2 FORMAT (16(I3,I2))
      READ 2,((MUS(1,N),I=1,2),N=1,16)
      READ 2,((NOTE(1,N),I=1,2),N=1,16)
      READ 2,((KLANG(1,N),I=1,2),N=1,16)
      READ 2,((LONG(1,N),I=1,2),N=1,16)
     1 FORMAT (13(I3))
      READ 1, (MOE(I),I=1,13)
     4 FORMAT (7(I3))
      READ 4, (JOE(I),I=1,7)
     6 FORMAT (5(I3))
      READ 6, (KOE(I),I=1,5)
     8 FORMAT (13)
      READ 8, (LOE(I),I=1,1)
      NEW = 1
    110 DO 215 N=1,16
      IF (MUS(1,N)-0) 71,73,71
    71 IF (LONG(1,N)-0) 72,73,72
    72 INT1(N)=MUS(1,N)-LONG(1,N)
      GO TO 74
    73 INT1(N)=0
    74 IF (MUS(1,N)-0) 75,77,75
    75 IF (KLANG(1,N)-0) 76,77,76
    76 INT4(N)=MUS(1,N)-KLANG(1,N)
      GO TO 78
    77 INT4(N)=0
    78 IF (MUS(1,N)-0) 79,202,79
    79 IF (NOTE(1,N)-0) 200,202,200

```

```

200 INT6(N)=MUS(1,N)-NOTE(1,N)
    GO TO 203
202 INT6(N)=0
203 IF (NOTE(1,N)-0) 204,205,204
204 IF (LONG(1,N)-0) 205,206,205
205 INT2(N)=NOTE(1,N)-LONG(1,N)
    GO TO 207
206 INT2(N)=0
207 IF (NOTE(1,N)-0) 208,210,208
208 IF (KLANG(1,N)-0) 209,210,209
209 INT5(N)=NOTE(1,N)-KLANG(1,N)
    GO TO 211
210 INT5(N)=0
211 IF (KLANG(1,N)-0) 212,214,212
212 IF (LONG(1,N)-0) 213,214,213
213 INT3(N)=KLANG(1,N)-LONG(1,N)
    GO TO 215
214 INT3(N)=0
215 CONTINUE
    M=1
    N=12
150 DO 183 I=1,16
    L=1
    IF (M-1) 152,151,152
151 J=INT1(I)
    IF (J-N) 80,80,86
    80 KORT1(I) = INT1(I)
    GO TO 182
152 IF (M-2) 154,153,154
153 J=INT2(I)
    IF (J-N) 81,81,86
    81 KORT2(I) = INT2(I)
    GO TO 182
154 IF (M-3) 156,155,156

```

```

155 J=INT3(I)
    IF (J-N)      82,82,86
      82 KORT3(I) = INT3(I)
      GO TO 182
156 IF (M-4) 158,157,158
157 J=INT4(I)
    IF (J-N)      83,83,86
      83 KORT4(I) = INT4(I)
      GO TO 182
158 IF (M-5) 160,159,160
159 J=INT5(I)
    IF (J-N)      84,84,86
      84 KORT5(I) = INT5(I)
      GO TO 182
160 IF (M-6) 165,161,185
161 J=INT6(I)
    IF (J-N)      85,85,86
      85 KORT6(I) = INT6(I)
      GO TO 182
      86 K=0
163 N=N+88
      K=K+88
    IF (J-N) 164,164,165
164 GO TO 170
165 N=N+12
      K=K+12
    IF (J-N) 166,166,167
166 GO TO 170
167 IF (L-6) 168,170,170
168 L=L+1
      GO TO 163
170 IF (M-1) 172,171,172

```

```

171 KORT1(I)=INT1(I)-K
172 IF (M-2) 174,173,174
173 KORT2(I)=INT2(I)-K
174 IF (M-3) 176,175,176
175 KORT3(I)=INT3(I)-K
176 IF (M-4) 178,177,178
177 KORT4(I)=INT4(I)-K
178 IF (M-5) 180,179,180
179 KORT5(I)=INT5(I)-K
180 IF (M-6) 182,181,185
181 KORT6(I)=INT6(I)-K
182 N=12
183 CONTINUE
184 N=12
185 M=M+1
186 IF (M-7) 184,185,185
187 GO TO 150
188 N=1
189 DO 66 I=1,13
190 IF (KORT1(N)-MOE(I)) 21,20,21
191 GO TO 30
192 IF (KORT2(N)-MOE(I)) 23,22,23
193 GO TO 48
194 IF (KORT3(N)-MOE(I)) 25,24,25
195 GO TO 60
196 IF (KORT4(N)-MOE(I)) 27,26,27
197 GO TO 36
198 IF (KORT5(N)-MOE(I)) 29,28,29
199 GO TO 54
200 IF (KORT6(N)-MOE(I)) 66,19,66
201 GO TO 42
202 JROOT = KORT1(N)
203 DO 32 J=1,7
204 IF (JROOT-JOE(J)) 32,31,32
205 MITH(N)=LONG(1,N)
206 GO TO 67

```

```

32 CONTINUE
   DO 34 J=1,5
   IF (JROOT-KOE(J))          34,33,34
33 MITH(N)=HUS(1,N)
   GO TO 67
34 CONTINUE
   IF (JROOT-LOE(1))          140,35,140
35 MITH(N)=0
   GO TO 67
140 GO TO 66
36 JROOT = KORT4(N)
   DO 38 J=1,7
   IF (JROOT-JOE(J))          38,37,38
37 MITH(N)=KLANG(1,N)
   GO TO 67
38 CONTINUE
   DO 40 J=1,5
   IF (JROOT-KOE(J))          40,39,40
39 MITH(N)=MUS(1,N)
   GO TO 67
40 CONTINUE
   IF (JROOT-LOE(1))          141,41,141
41 MITH(N)=0
   GO TO 67
141 GO TO 66
42 JROOT = KORT6(N)
   DO 44 J=1,7
   IF (JROOT-JOE(J))          44,43,44
43 MITH(N)=NOTE(1,N)
   GO TO 67
44 CONTINUE
   DO 46 J=1,5
   IF (JROOT-KOE(J))          46,45,46

```

```

45 MITH(N)=MUS(1,N)
   GO TO 67
46 CONTINUE
   IF (JROOT-LOE(1)) 142,47,142
47 MITH(N)=0
   GO TO 67
142 GO TO 66
48 JROOT = KORT2(N)
   DO 50 J=1,7
   IF (JROOT-JOE(J)) 50,49,50
49 MITH(N)=LONG(1,N)
   GO TO 67
50 CONTINUE
   DO 52 J=1,5
   IF (JROOT-KOE(J)) 52,51,52
51 MITH(N)=NOTE(1,N)
   GO TO 67
52 CONTINUE
   IF (JROOT-LOE(1)) 143,53,143
53 MITH(N)=0
   GO TO 67
143 GO TO 66
54 JROOT= KORT5(N)
   DO 56 J=1,7
   IF (JROOT-JOE(J)) 56,55,56
55 MITH(N)=KLANG(1,N)
   GO TO 67
56 CONTINUE
   DO 58 J=1,5
   IF (JROOT-KOE(J)) 58,57,58
57 MITH(N)=NOTE(1,N)
   GO TO 67
58 CONTINUE
   IF (JROOT-LOE(1)) 144,59,144

```

```

59 MITH(N)=0
   GO TO 67
144 GO TO 66
60 JROOT = KORT3(N)
   DO 62 J=1,7
   IF (JROOT-JOE(J))      62,61,62
61 MITH(N)=LONG(1,N)
   GO TO 67
62 CONTINUE
   DO 64 J=1,5
   IF (JROOT-KOE(J))      64,63,64
63 MITH(N) =KLANG(1,N)
   GO TO 67
64 CONTINUE
   IF (JROOT-LOE(1))      66,65,66
65 MITH(N)=0
   GO TO 67
66 CONTINUE
67 IF (N-16) 69,68,68
69 N=N+1
   GO TO 186
68 PRINT 131, (MITH(I),I=1,16)
   PUNCH 132,(MITH(I),I=1,16)
   IF (NEW-2) 100,101,102
100 READ 2,((MUS(I,N),I=1,2),N=1,16)
   READ 2,((NOTE(I,N),I=1,2),N=1,16)
   READ 2,((KLANG(I,N),I=1,2),N=1,16)
   READ 2,((LONG(I,N),I=1,2),N=1,16)
   NEW = NEW +1
   GO TO 110
101 READ 2,((MUS(I,N),I=1,2),N=1,16)
   READ 2,((NOTE(I,N),I=1,2),N=1,16)
   READ 2,((KLANG(I,N),I=1,2),N=1,16)
   READ 2,((LONG(I,N),I=1,2),N=1,16)
   NEW = NEW +1

```



```

GO TO 110
102 IF (NEW - 4) 103,104,105
103 READ 2,((MUS(I,N),I=1,2),N=1,16)
   READ 2,((NOTE(I,N),I=1,2),N=1,16)
   READ 2,((KLANG(I,N),I=1,2),N=1,16)
   READ 2,((LONG(I,N),I=1,2),N=1,16)
   NEW = NEW +1
GO TO 110
104 READ 2,((MUS(I,N),I=1,2),N=1,16)
   READ 2,((NOTE(I,N),I=1,2),N=1,16)
   READ 2,((KLANG(I,N),I=1,2),N=1,16)
   READ 2,((LONG(I,N),I=1,2),N=1,16)
   NEW = NEW +1
GO TO 110
105 STOP
END

5015150151502514104141041410414104140941409414074140541407414094140941
000000 *
000000 0
000000 0
7 5 4 8 3 9 2 10 1 11 12 0 6
0 3 4 7 10 11 12
1 2 5 8 9
6
40941412414104141041410414064140741410414084140641406415065150751507515055150451
40642406424074240742403424034240342402423123240332402423123231032312324024240142
000000 0
000000 0
5045150951505515055150451504515055150551505515035150351502515015141241
50452309423093240942501525015250252412424114250252412423103230932310324044240642
4014340143402434024331035310333103330933079331033309333070030500307333093330933

```

000000 0
41241405414024141141414124141241502515035141241502515035150551505515055150334
40342312323113240242403423123231232311323123230932310323123240242402424024231232
30933409434114340743407433083330733305335030030633305333033023330333303330733
306343063430734307343033430334302342122430334302342122421024212243023430234
5035150251503515075150751507515125150951507515055150551503515045150951
311323113231232402424034241142414242501515015150252410424114240942408424094250452
3073330633073330553303333053307334074340743405434044340243402434014340143
30234309343033430234212243023430334304343043430534307343083430934310343093430734


```

* GILBERT ROLLER - QUARTET - REPEATED CHORDS - NO.9
  DIMENSION MUS(3,80),NOTE(3,80),KLANG(3,80),LONG(3,80),KAN(80) ,
    1 KUN(80),KEN(80),KIN(80),MITH(80)
  2 FORMAT (16(13,11,11))
    READ 2,((MUS(I,N),I=1,3),N=1,80)
    READ 2,((NOTE(I,N),I=1,3),N=1,80)
    READ 2,((KLANG(I,N),I=1,3),N=1,80)
    READ 2,((LONG(I,N),I=1,3),N=1,80)
  3 FORMAT (16(14))
    READ 3, (MITH(I),I=1,80)
  13 FORMAT (1H0,5X,17HMELODIC INTERVALS)
  14 FORMAT (1H0,5X,8H1 VIOLIN)
  15 FORMAT (1H0,16(14))
  16 FORMAT (1H0,5X,8H2 VIOLIN)
  17 FORMAT (1H0,5X, 5HVIOLA)
  18 FORMAT (1H0,5X,5HCELLO)
  19 FORMAT (1H0,5X,15HHINDEMI TH ROOTS)
  20 FORMAT (16(14))
    IAN=0
    MAN=0
    DO 8 N=1,80
      IAN=MAN
      JAN=MUS(1,N)
      IF (IAN) 30,20,30
      IF (JAN) 21,20,21
      20 KAN(N)=0
      MAN = JAN
      GO TO 8
      21 KAN(N)=IAN-JAN
      MAN=JAN
      8 CONTINUE
      IAN=0
      MAN=0
      DO 9 N=1,80

```

```

IAN=MAN
JAN=NOTE(1,N)
IF (IAN) 31,22,31
31 IF (JAN) 23,22,23
22 KON(N)=0
MAN=JAN
GO TO 9
23 KON(N)=IAN-JAN
MAN=JAN
9 CONTINUE
IAN=0
MAN=0
DO 10 N=1,80
IAN=MAN
JAN=KLANG(1,N)
IF (IAN) 32,24,32
32 IF (JAN) 25,24,25
24 KUN(N)=0
MAN=JAN
GO TO 10
25 KUN(N)=IAN-JAN
MAN=JAN
10 CONTINUE
IAN=0
MAN=0
DO 11 N=1,80
IAN=MAN
JAN=LONG(1,N)
IF (IAN) 33,26,33
33 IF (JAN) 27,26,27
26 KEN(N)=0
MAN=JAN
GO TO 11
27 KEN(N)=IAN-JAN

```

```

MAN=JAN
11 CONTINUE
  IAN=0
  MAN=0
  DO 12 N=1,80
    IAN=MAN
    JAN=MITH(N)      34,28,34
    IF (IAN)          29,28,29
  34 IF (JAN)
  28 KIN(N)=0
  MAN=JAN
  GO TO 12
  29 KIN(N)=IAN-JAN
  MAN = JAN
  12 CONTINUE
  PRINT 13
  PRINT 14
  PRINT 15, (KAN(I),I=1,80)
  PUNCH 20, (KAN(I),I=1,80)
  PRINT 16
  PRINT 15, (KON(I),I=1,80)
  PUNCH 20, (KON(I),I=1,80)
  PRINT 17
  PRINT 15, (KUN(I), I=1,80)
  PUNCH 20, (KUN(I), I=1,80)
  PRINT 18
  PRINT 15, (KEN(I),I=1,80)
  PUNCH 20, (KEN(I),I=1,80)
  PRINT 19
  PRINT 15, (KIN(I), I=1,80)
  PUNCH 20, (KIN(I), I=1,80)
  STOP
  END

```

000000 0
4064240642407424074240342403424024231232403324024231032312324024240142
5045230942309324094250152501525025241242411425025241242310323093231032404240642
4034231232311324024240342312323113231232309323103231232402424024231232
311323113231232402424034241142412425015015150252410424114240942408424094250452
000000 0
000000 0
4014340143402434024331033310333093307333103330933070030500307333093330933
309334094341143407434074330833307333053330300306333033302333033305330733
307333083330733305333033330533307334074340743404340243402434014340143
000000 0
000000 0
000000 0
000000 0
30634306343073430734303343033430324303243033430234212243033430234212243023430234
30234305343033430234212243023430334304343043430534307343063430934310343093430734
0 0 0 0
406 0 412 407 403 403 403 410 408 403 402 0 507 312 402 309
401 401 505 402 310 310 310 505 307 310 505 503 305 307 309 302
306 306 307 307 303 308 412 302 212 303 310 212 210 505 302 307
307 305 312 302 212 507 412 304 512 502 307 308 402 310 309 509


```

*      GILBERT ROLLER - QUARTET - CHORD PROGRESSION - NO.10
      DIMENSION KAN(400) , MELO(30)
      30 FORMAT (16(14))
      READ 30, (KAN(I),I=1,400)
      105 FORMAT (1H0,4H P1,9H MINOR 2,4H M2,9H MINOR 3,4H M3,4H P4,
      14H A4,4H P5,9H MINOR 6,4H M6,9H MINOR 7,4H M7,4H P8)
      106 FORMAT (1H0,1X,4(13,3X),3X,3(13,1X),13,3(3X,13),4X,13,2X,13)
      107 FORMAT (1H0,17HMELODIC INTERVALS)
      111 FORMAT (1H0,8H1 VIOLIN)
      112 FORMAT (1H0,9H11 VIOLIN)
      113 FORMAT (1H0,5HVIOLA)
      114 FORMAT (1H0,5HCELO)
      115 FORMAT (1H0,15HHINDENWITH ROOTS)
      PRINT 111
      M=1
      N=12
      NO = 1
      NOT = 80
      66 IT = 1
      31 DO 29 I= NO,NOT
      L=1
      51 IF (KAN(I)) 40,41,41
      40 GO TO 52
      41 J=KAN(I)
      IF (J-N) 50,50,57
      50 KAN(I)=KAN(I)
      GO TO 76
      57 K=0
      47 N=N+88
      K=K+88
      IF (J-N) 58,58,59
      58 GO TO 70
      59 N = N+12
      K=K+12
      IF (J-N) 60,60,65

```



```

60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 KAN(I) = KAN(I) - K
76 N=12
   GO TO 29
52 J=KAN(I)
   IF (J+N) 54,53,53
53 KAN(I) = KAN(I)
   GO TO 17
54 K=0
10 N=N+88
   K=K+88
   IF (J+N) 12,11,11
11 GO TO 16
12 N=N+12
   K=K+12
   IF (J+N) 14,13,13
13 GO TO 16
14 IF (L-6) 15,16,16
15 L=L+1
   GO TO 10
16 KAN(I)=KAN(I)+K
17 N=12
29 CONTINUE
   PRINT 107
   PRINT 106, (KAN(I),I=NO,NOT)
   DO 6 I=1,26
     MELO(I) = 0
     6 CONTINUE
100 MOT = 0

```

```

88 DO 83 I= 1,13
   IF (KAN(IT)-MOT) 93,82,93
82 MELO(I) = MELO(I)+1
93 MOT = MOT+1
83 CONTINUE
   MOT = 0
   DO 85 I= 14,26
   IF (KAN(IT) + MOT) 95,84,95
84 MELO(I) = MELO(I) - 1
95 MOT = MOT+1
85 CONTINUE
   IF (IT- NOT) 20,87,87
20 IT = IT +1
   GO TO 100
87 PRINT 105
   PRINT 106, (MELO(I),I=1,13)
   PRINT 106, (MELO(I),I=14,26)
   IF (M-5) 18,18,110
18 NO=NO+80
   NOT =NOT+80
   N=12
   M=M+1
   IF (M-3) 116,117,118
116 PRINT 112
   GO TO 125
117 PRINT 113
   GO TO 125
118 IF (M-5) 119,120,110
119 PRINT 114
   GO TO 125
120 PRINT 115
125 GO TO 31
110 GO TO 121
121 STOP
   END

```

0	0	-1	0	92	0	0	1	1	-2	1	2	2	-2	-2	0
0	-3	0	2	0	4	-1	-3	2	2	0	-100	-1	0	2	1
0	-4	3	0	0	1	0	-1	0	0	0	2	0	1	1	89
0	9	1	-9	0	-1	0	-90	-1	91	-90	-1	-2	0	0	2
0	1	-1	-2	-2	0	0	0	-5	3	2	2	0	0	1	-5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-4	0	-1	0	4	0	0	1	90	-91	1	90	2	-2	-90	1
-103	195	0	-100	-92	0	-1	90	1	-91	90	102	1	-1	-94	-2
3	91	1	-91	-1	91	0	1	-1	3	-1	-2	-90	0	0	90
1	0	-1	-90	-1	-8	-1	-89	0	-1	92	-1	2	1	-1	-95
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-52	0	-1	0	92	0	0	1	2	-3	1	2	2	-2	-2	0
0	-100	-2	4	0	99	1	2	2	-3	1	2	1	-1	-2	-2
0	-1	1	2	2	-2	-2	-2	100	0	2	1	2	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-4	0	-1	0	4	0	0	1	90	-91	1	90	2	-2	-90	0
0	-3	2	1	90	-90	-1	-1	0	-1	-2	-1	-1	-1	1	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-4	0	0	5	4	0	0	-7	2	5	1	0	0	195	-90	93
-92	0	-104	103	92	0	0	-195	198	-3	-195	2	198	-2	-2	7
-4	0	-1	0	4	-5	-104	110	90	-91	-7	93	2	-295	203	-5
0	2	-7	10	90	-295	95	108	-208	10	195	-1	-94	92	1	-200

* GILBERT ROLLER - NO.10 - QUARTET - RESULTS
I VIOLIN

MELODIC INTERVALS												
0	0	-1	0	4	0	0	1	1	-2	1	2	2
-2	-2	0	0	-3	0	2	0	4	-1	-3	2	2
0	-12	-1	0	2	1	0	-4	3	0	0	1	0
-1	0	0	0	2	0	1	1	1	0	9	1	-9
0	-1	0	-2	-1	3	-2	-1	-2	0	0	2	0
1	-1	-2	-2	0	0	0	-5	3	2	2	0	0
1	-5											
P1	MINOR 2	M2	MINOR 3	M3	P4	A4	P5	MINOR 6	M6	MINOR 7	M7	P8
30	11	10	3	2	0	0	0	0	1	0	0	0
-30	-8	-8	-2	-1	-2	0	0	0	-1	0	0	-1

II VIOLIN

MELODIC INTERVALS												
0	0	0	-4	0	-1	0	4	0	0	1	2	-3
0	0	0	-2	-2	1	-3	7	0	-12	-4	0	-1
1	2	2	2	2	1	-1	-6	-2	3	3	1	-3
2	1	-3	1	-1	3	-1	-2	-2	0	0	2	1
-1	3	0	1	-8	-1	-1	0	-1	4	-1	2	1
0	-1	-2	-1								2	1
-1	-7											
P1	MINOR 2	M2	MINOR 3	M3	P4	A4	P5	MINOR 6	M6	MINOR 7	M7	P8
27	9	8	4	2	0	0	1	0	0	0	0	0
-27	-13	-6	-4	-2	-1	-1	-1	-1	0	0	0	-1

VIOLA

MELODIC INTERVALS

4	-5	-4	10	2	-3	-7	10	2	-7	3	-5	0
2	-7	10	2		7	8	-8	10	7	-1	-6	4
1	-12											
P1	MINOR 2	M2	MINOR 3	M3	P4	A4	P5	MINOR 6	M6	MINOR 7	P8	
28	2	7	2	4	3	0	4	1	0	6	0	0
-28	-2	-3	-2	-5	-2	-1	-7	-1	0	0	0	-1

* GILBERT ROLLER - QUINTET - INTERVALS - NO.1

```

DIMENSION MUS(J,67),NOTE(J,67),KLANG(J,67),KORN(J,67),LONG(3,67),
1LUM(20),KORD(13)
95 FORMAT (4(14))
195 FORMAT (1H0,4(14))
20 FORMAT (3(13,11,11))
2 FORMAT (16(13,11,11))
200 FORMAT (8(13,11,11))
READ 2,((MUS(I,N),I=1,3),N=1,64)
READ20,((MUS(I,N),I=1,3),N=65,67)
READ 2,((NOTE(I,N),I=1,3),N=1,64)
READ20,((NOTE(I,N),I=1,3),N=65,67)
READ 2,((KLANG(I,N),I=1,3),N=1,64)
READ20,((KLANG(I,N),I=1,3),N=65,67)
READ 2,((KORN(I,N),I=1,3),N=1,64)
READ20,((KORN(I,N),I=1,3),N=65,67)
READ 2,((LONG(I,N),I=1,3),N=1,64)
READ20,((LONG(I,N),I=1,3),N=65,67)
M=0
J=1
10 M=M+1
N=1
I=1
LUM(N)= MUS(I,M)
N=N+1
LUM(N)= NOTE(I,M)
N=N+1
LUM(N)= KLANG(I,M)
N=N+1

```



```

LUM(N)=      KUMN(1,M)
N=N+1
LUM(N)=      LONG(1,M)
L=N-4
3 IF (LUM(N)-0) 12,4,12
4 N=N-1
  IF (N-L) 12,12,11
11 GO TO 3
12 L=N-L
13 IF (LUM(1)) 3,7,5
5 IF (LUM(N)-LUM(1)) 7,7,0
6 KE=LUM(N)
  KA= LUM(1)
  LUM(N)=KA
  LUM(1)=KE
7 I=I+1
  L=L-1
  IF (L-0) 9,9,8
8 GO TO 13
9 I=1
91 IF (LUM(1)-0) 93,93,92
92 KORD(1)=LUM(1)-LUM(N)
  LONG (1,M)=LUM(N)
  GO TO 94
93 KORD(1)=0
94 IF (I-4 ) 105,106,106
105 I=I+1
  GO TO 91
106 PRINT 95, (KORD(1),I=1,4)
  PUNCH 95, (KORD(1),I=1,4)
98 J=J+1
  IF (J-68) 102,103,103

```


* GILBERT MULLER - QUINTET - INTERVALS - NO.1 - RESULTS

300	200	104	197
300	200	104	197
300	200	104	197
309	209	201	109
304	204	108	109
309	209	201	109
299	200	104	197
309	209	201	109
305	205	109	110
309	209	201	109
300	200	104	197
300	200	104	197
309	209	201	109
304	204	108	109
309	209	201	109
300	200	104	197
104	100	8	0
104	100	8	0
107	103	7	0
198	195	95	0
202	199	96	0
293	201	95	0
202	199	96	0
198	195	95	0
107	103	7	0
104	100	8	0
104	100	8	0
195	191	95	0
198	195	95	0
202	199	96	0


```

*   GILBERT ROLLEN - QUINTET - REDUCTION
      DIMENSION KURU (268),KURT1(67),KURT2(67),KURT3(67),KURT4(67)
      1 FORMAT (16(14))
      3 FORMAT (4(14))
105 READ 3, (KURU(I),I=1,268)
      2 FORMAT (16(14))
      M=1
      N=12
      IT = 1
      KT = 265
      5 DO 98 I = M,KT, 4
      L=1
      J=KURU(I)
      11 IF (M-2) 11,13,15
      11 IF (J-N) 12,12,45
      12 KURT1(IT) = KURU(I)
      60 TO 96
      13 IF (J-N) 14,14,45
      14 KURT2(IT) = KURU(I)
      60 TO 96
      15 IF (M-4) 16,16,98
      16 IF (J-N) 17,17,45
      17 KURT3(IT) = KURU(I)
      60 TO 96
      18 IF (J-N) 19,19,45
      19 KURT4(IT) = KURU(I)
      60 TO 96
      45 K=0
      47 N=N+88
      K=K+88
      IF (J-N) 58,58,59
      58 60 TO 70
      59 N = N+12
      K=K+12

```

```

IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
GO TO 47
70 IF (M-2) 71,72,73
71 KORT1(IT) = KORD(I) -K
GO TO 96
72 KORT2(IT) = KORD(I) -K
73 IF (M-4) 74,75,96
74 KORT3(IT) = KORD(I) -K
GO TO 96
75 KORT4(IT) = KORD(I) -K
96 N=12
IT = IT+1
98 CONTINUE
IT = 1
M=M+1
KT=KT+1
N=12
IF (M-5) 99,100,100
99 GO TO 5
100 PRINT 2, (KORT1(I),I=1,67)
PRINT 2, (KORT2(I),I=1,67)
PRINT 2, (KORT3(I),I=1,67)
PRINT 2, (KORT4(I),I=1,67)
PUNCH 1, (KORT1(I),I=1,67)
PUNCH 1, (KORT2(I),I=1,67)
PUNCH 1, (KORT3(I),I=1,67)
PUNCH 1, (KORT4(I),I=1,67)
STOP
END

```

300 200 104 197
 300 200 104 197
 300 200 104 197
 309 209 201 109
 304 204 108 109
 309 209 201 109
 299 200 104 197
 309 209 201 109
 305 205 109 110
 309 209 201 109
 300 200 104 197
 300 200 104 197
 309 209 201 109
 304 204 108 109
 309 209 201 109
 300 200 104 197
 104 100 8 0
 104 100 8 0
 107 103 7 0
 198 195 95 0
 202 199 96 0
 293 201 95 0
 202 199 96 0
 198 195 95 0
 107 103 7 0
 104 100 8 0
 104 100 8 0
 195 191 95 0
 198 195 95 0
 202 199 96 0
 205 201 95 0
 202 199 96 0
 198 195 95 0
 195 191 95 0
 300 205 199 202
 300 205 199 202

300 205 199 202

309	209	204	206
304	207	202	206
309	209	204	206
300	205	199	202
309	209	204	206
304	207	202	206
309	209	204	206
300	205	199	202
300	205	199	202
309	209	204	206
304	207	202	206
309	209	204	206
0	0	0	0
0	200	104	0
0	200	102	0
0	202	0	0
0	204	106	0
0	204	104	0
0	206	0	0
0	207	108	0
0	207	106	0
0	296	0	0
0	295	197	0
0	295	195	0
0	297	0	0
0	295	199	0
0	295	197	0
0	293	0	0
0	291	197	0
0	291	195	0

```

* 749817 ROLLER J          3MIN IC OP
PROGRAM QUIN4
  DIMENSION INTA(64),INTB(64),INTC(64),INTD(64)
49 FORMAT (4(I4))
  1 FORMAT (16(I4))
  READ 1, (INTA(I),I=1,64)
  READ 1, (INTB(I),I=1,64)
  READ 1, (INTC(I),I=1,64)
  READ 1, (INTD(I),I=1,64)
  DO 30 I=1,64
    M1 = INTA(I)
    M2 = INTB(I)
    M3 = INTC(I)
    M4 = INTD(I)
    DO 30 K=1,4
      KORTS = 0
      IF (KORTS - M1) 12,13,13
12 KORTS = M1
13 IF (KORTS-M2) 14,15,15
14 KORTS = M2
15 IF (KORTS-M3) 16,17,17
16 KORTS = M3
17 IF (KORTS - M4) 40,41,41
40 KORTS =M4
41 IF (KORTS - M1) 19,18,19
18 M1=0
    GO TO 23
19 IF (KORTS - M2) 21,20,21
20 M2=0
    GO TO 23
21 IF (KORTS-M3) 42,22,42

```



```

*      GILBERT ROLLER - QUINTER 111A - RE-ARRANGE
      DIMENSION KORD (16,20),INTERV(320)
      1 FORMAT (10(14))
      2 FORMAT (16(14))
      READ 2, ((KORD (I,N),I=1,16),N=1,20)
      J=1
      I=1
      4 DO 5 N=1,20
        INTERV(J)=KORD(I,N)
        J=J+1
      5 CONTINUE
      IF (I-12) 6,7,7
      6 I=I+1
      GO TO 4
      7 PUNCH 1, (INTERV(I),I=1,320)
      STOP
      END

```

QUINTET - NO. Va - REDUCTION OF LONG
USE QUARTET PROGRAM Va

```

* 74981PROGRAM EBU1R5          1MIN IC OP
DIMENSION INFO(2,16),MUS(3,16),NOTE(3,48),LONG(3,16),KORD(64)
1 FORMAT (16(13,12))
2 FORMAT (16(13,11,11))
  READ 1, ((INFO(I,N),I=1,2),N=1,16)
  READ 2, ((MUS(I,N),I=1,3),N=1,16)
  READ 2, ((NOTE(I,N),I=1,3),N=1,48)
  READ 2, ((LONG(I,N),I=1,3),N=1,16)
3 FORMAT (16(14))
  READ 3, (KORD(I),I=1,64)
20 FORMAT (1H0,17,115,16X,2H0,113,110/)
270 FORMAT (1H0,17,115,18X,2H2-,113,110/)
271 FORMAT (1H0,17,115,18X,2H2-,113,110/)
272 FORMAT (1H0,17,115,18X,2H3-,113,110/)
273 FORMAT (1H0,17,115,18X,2H3-,113,110/)
274 FORMAT (1H0,17,115,18X,2H4-,113,110/)
275 FORMAT (1H0,17,115,18X,2H4+,113,110/)
276 FORMAT (1H0,17,115,18X,2H5-,113,110/)
277 FORMAT (1H0,17,115,18X,2H6-,113,110/)
278 FORMAT (1H0,17,115,18X,2H6-,113,110/)
279 FORMAT (1H0,17,115,18X,2H7-,113,110/)
280 FORMAT (1H0,17,115,18X,2H7-,113,110/)
281 FORMAT (1H0,17,115,18X,2H8-,113,110/)
21 FORMAT (41X,2H0,113,110/)
285 FORMAT (41X,2H2-,113,110/)
286 FORMAT (41X,2H2-,113,110/)
287 FORMAT (41X,2H3-,113,110/)
288 FORMAT (41X,2H3-,113,110/)
289 FORMAT (41X,2H4-,113,110/)
290 FORMAT (41X,2H4+,113,110/)
291 FORMAT (41X,2H5-,113,110/)

```



```

292 FORMAT (41X,2H6-,113,110/)
293 FORMAT (41X,2H6.,113,110/)
294 FORMAT (41X,2H7-,113,110/)
295 FORMAT (41X,2H7.,113,110/)
296 FORMAT (41X,2H8-,113,110/)
22  FORMAT (41X,2H0 ,113,110//)
315 FORMAT (41X,2HC ,113,110//)
316 FORMAT (41X,2HC+ ,113,110//)
317 FORMAT (41X,2HD ,113,110//)
318 FORMAT (41X,2HE- ,113,110//)
319 FORMAT (41X,2HE ,113,110//)
320 FORMAT (41X,2HF ,113,110//)
321 FORMAT (41X,2HF+ ,113,110//)
322 FORMAT (41X,2HG ,113,110//)
323 FORMAT (41X,2HA- ,113,110//)
324 FORMAT (41X,2HA ,113,110//)
325 FORMAT (41X,2HB- ,113,110//)
326 FORMAT (41X,2HB ,113,110//)
350 FORMAT (1H0,4HCODE/4X,21H. IS MAJOR OR PERFECT/4X,10H- IS MINOR/
      14X,14H+ IS AUGMENTED)
      PRINT 350
42  FORMAT (1H0,13HMEASURE NO. ,19HRHYTHMIC POSITION ,
      120HHARMONIC INTERVALS ,8HOCIAVE ,12HINSTRUMENT )
41  PRINT 42
      J=64
      DO 302 I=1,16
12  J=J-63
      N=1
      IF (KORD(J)-0) 4,5,4
5    PRINT 20,INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
      GO TO 217
4    IF (KORD(J)-2) 200,201,202
200 PRINT 270, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)

```

```

GO TO 217
201 PRINT 271, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
202 IF (KORD(J)-4) 203,204,205
203 PRINT 272, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
204 PRINT 273, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
205 IF (KORD(J)-6) 206,207,208
206 PRINT 274, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
207 PRINT 275, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
208 IF (KORD(J)-8) 209,210,211
209 PRINT 276, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
210 PRINT 277, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
211 IF (KORD(J)-10) 212,213,214
212 PRINT 278, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
213 PRINT 279, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
214 IF (KORD(J)-12) 215,216,217
215 PRINT 280, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
216 PRINT 281, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
217 J=J+16
DO 253 L=1,3
9 IF (KORD(J)-0) 6,7,6
7 PRINT 21, NOTE(2,N),NOTE(3,N)
GO TO 235

```



```

6 IF (KORD(J)-2) 218,219,220
218 PRINT 285, NOTE(2,N),NOTE(3,N)
GO TO 235
219 PRINT 286, NOTE(2,N),NOTE(3,N)
GO TO 235
220 IF (KORD(J)-4) 221,222,223
221 PRINT 287, NOTE(2,N),NOTE(3,N)
GO TO 235
222 PRINT 288, NOTE(2,N),NOTE(3,N)
GO TO 235
223 IF (KORD(J)-6) 224,225,226
224 PRINT 289, NOTE(2,N),NOTE(3,N)
GO TO 235
225 PRINT 290, NOTE(2,N),NOTE(3,N)
GO TO 235
226 IF (KORD(J)-8) 227,228,229
227 PRINT 291, NOTE(2,N),NOTE(3,N)
GO TO 235
228 PRINT 292, NOTE(2,N),NOTE(3,N)
GO TO 235
229 IF (KORD(J)-10) 230,231,232
230 PRINT 293, NOTE(2,N),NOTE(3,N)
GO TO 235
231 PRINT 294, NOTE(2,N),NOTE(3,N)
GO TO 235
232 IF (KORD(J)-12) 233,234,235
233 PRINT 295, NOTE(2,N),NOTE(3,N)
GO TO 235
234 PRINT 296, NOTE(2,N),NOTE(3,N)
235 J=J+16
N=N+16
253 CONTINUE
N=1
IF (LONG(1,N) -0) 10,11,10
11 PRINT 22, LONG(2,N),LONG(3,N)

```



```

GO TO 302
  10 IF (LONG(1,N)-2) 255,256,257
255 PRINT 315, LONG(2,N),LONG(3,N)
GO TO 302
256 PRINT 316, LONG(2,N),LONG(3,N)
GO TO 302
257 IF (LONG(1,N) -4) 258,259,260
258 PRINT 317, LONG(2,N),LONG(3,N)
GO TO 302
259 PRINT 318, LONG(2,N),LONG(3,N)
GO TO 302
260 IF (LONG(1,N) -6) 261,262,263
261 PRINT 319, LONG(2,N),LONG(3,N)
GO TO 302
262 PRINT 320, LONG(2,N),LONG(3,N)
GO TO 302
263 IF (LONG (1,N) -8) 264,265,266
264 PRINT 321, LONG(2,N),LONG(3,N)
GO TO 302
265 PRINT 322, LONG(2,N),LONG(3,N)
GO TO 302
266 IF (LONG(1,N) -10) 267,268,269
267 PRINT 323, LONG(2,N),LONG(3,N)
GO TO 302
268 PRINT 324, LONG(2,N),LONG(3,N)
GO TO 302
269 IF (LONG(1,N)-12) 327,328,302
327 PRINT 325, LONG (2,N),LONG(3,N)
GO TO 302
328 PRINT 326, LONG (2,N),LONG(3,N)
302 CONTINUE
  15 STOP

```


* QUINTET - NO. 5 - PRINT OUT - RESULTS

CODE

• IS MAJOR OR PERFECT
- IS MINOR
/ IS AUGMENTED

MEASURE NO.	RHYTHMIC POSITION	HARMONIC INTERVALS	SCALE	VE	INSTRUMENT
1	1	8. 8. 3. 6. E	5 4 3 4 2	1 2 3 4 5	
1	2	8. 8. 3. 6. E	5 4 3 4 2	1 2 3 4 5	
1	3	8. 8. 3. 6. E	5 4 3 4 2	1 2 3 4 5	
2	1	6. 6. 2- 6. D	5 4 4 3 2	1 2 3 4 5	

2	2	3. 3. 6- 6. D	5 4 3 3 2	1 2 3 4 5
2	3	6. 6. 2- 6. D	5 4 4 3 2	1 2 3 4 5
3	1	7. 7. 3. 6. E	5 4 3 4 2	4 2 3 4 5
3	2	6. 6. 2- 6. D	5 4 4 3 2	1 2 3 4 5
3	3	4. 4. 6. 7- G	5 4 3 3 2	1 2 3 4 5

4	1	6: 6: 2- 6: D	5 4 4 3 2	1 2 3 4 5
4	2	8: 8: 3: 6: B	5 4 3 4 2	1 2 3 4 5
4	3	8: 8: 3: 6: B	5 4 3 4 2	1 2 3 4 5
5	1	6: 6: 2- 6: D	5 4 4 3 2	1 2 3 4 5
5	2	3: 3: 6- 6: D	5 4 3 3 2	1 2 3 4 5

1 2 3 4 5

5 4 4 3 2

6. 6. 2. 6. D

3

5

1 2 3 4 5

5 4 3 4 2

8. 8. 3. 6. B

1

6

```

*      GILBERT ROLLER - QUINTET - CHORD RETETITION - NO.6
      DIMENSION KORD (4,67)
41  FORMAT (4(I4))
      READ 1, ((KORD(I,N),I=1,4),N=1,67)
      J=1
      DO 40 N=1,67
        K1=KORD(1,N)
        K2=KORD(2,N)
        K3=KORD(3,N)
        K4=KORD(4,N)
        J=J+1
      DO 40 M=J,67
        L1=KORD(1,M)
        L2=KORD(2,M)
        L3=KORD(3,M)
        L4=KORD(4,M)
        IF (K1-L1) 31,32,31
31  GO TO 40
32  IF (K2-L2) 31,34,31
34  IF (K3-L3) 31,35,31
35  IF (K4-L4) 31,36,31
36  IF (K1-0) 43,38,43
37  IF (K2-0) 43,38,43
38  IF (K3-0) 43,39,43
39  IF (K4-0) 43,40,43
43  IF (K1-12) 61,60,61
60  K1=8
      GO TO 100
61  IF (K1-11) 63,62,63
62  K1=7
      GO TO 100
63  IF (K1-10) 65,64,65

```


62 K1=7	
GO TO 100	
63 IF (K1-10)	65.64.65
64 K1=-7	
GO TO 100	
65 IF (K1-9)	67.66.67
66 K1=6	
GO TO 100	
67 IF (K1-8)	69.68.69
68 K1=-6	
GO TO 100	
69 IF (K1-7)	71.70.71
70 K1=5	
GO TO 100	
71 IF (K1-6)	73.72.73
72 K1=-5	
GO TO 100	
73 IF (K1-5)	75.74.75
74 K1=4	
GO TO 100	
75 IF (K1-4)	77.76.77
76 K1=3	
GO TO 100	
77 IF (K1-3)	79.78.79
78 K1=-3	
GO TO 100	
79 IF (K1-2)	81.80.81
80 K1=2	
GO TO 100	
81 IF (K1-1)	83.82.83
82 K1=-2	
GO TO 100	
83 IF (K1-0)	100.84.100
84 K1=1	

100 IF (K2-12)	102,101,102
101 K2=8	
GO TO 130	
102 IF (K2-11)	104,103,104
103 K2=7	
GO TO 130	
104 IF (K2-10)	106,105,106
105 K2=-7	
GO TO 130	
106 IF (K2-9)	108,107,108
107 K2=6	
GO TO 130	
108 IF (K2-8)	110,109,110
109 K2=-6	
GO TO 130	
110 IF (K2-7)	112,111,112
111 K2=5	
GO TO 130	
112 IF (K2-6)	114,113,114
113 K2=-5	
GO TO 130	
114 IF (K2-5)	116,115,116
115 K2=4	
GO TO 130	
116 IF (K2-4)	118,117,118
117 K2=3	
GO TO 130	
118 IF (K2-3)	120,119,120
119 K2=-3	
GO TO 130	
120 IF (K2-2)	122,121,122
121 K2=2	
GO TO 130	
122 IF (K2-1)	124,123,124

123 K2=-2	
GO TO 130	
124 IF (K2=0)	130,125,130
125 K2=1	
130 IF (K3=12)	132,131,132
131 K3=8	
GO TO 156	
132 IF (K3=11)	134,133,134
133 K3=7	
GO TO 156	
134 IF (K3=10)	136,135,136
135 K3=-7	
GO TO 156	
136 IF (K3=9)	138,137,138
137 K3=6	
GO TO 156	
138 IF (K3=8)	140,139,140
139 K3=-6	
GO TO 156	
140 IF (K3=7)	142,141,142
141 K3=5	
GO TO 156	
142 IF (K3=6)	144,143,144
143 K3=-5	
GO TO 156	
144 IF (K3=5)	146,145,146
145 K3=4	
GO TO 156	
146 IF (K3=4)	148,147,148
147 K3=3	
GO TO 156	
148 IF (K3=3)	150,149,150
149 K3=-3	

GU TO 156
 150 IF (K3-2) 152,151,152
 151 K3=2
 GU TO 156
 152 IF (K3-1) 154,153,154
 153 K3=-2
 GU TO 156
 154 IF (K3-0) 156,155,156
 155 K3=1
 156 IF (K4-12) 161,160,161
 160 K4=8
 GU TO 185
 161 IF (K4-11) 163,162,163
 162 K4=7
 GU TO 185
 163 IF (K4-10) 165,164,165
 164 K4=-7
 GU TO 185
 165 IF (K4-9) 167,166,167
 166 K4=6
 GU TO 185
 167 IF (K4-8) 169,168,169
 168 K4=-6
 GU TO 185
 169 IF (K4-7) 171,170,171
 170 K4=5
 GU TO 185
 171 IF (K4-6) 173,172,173
 172 K4=-5
 GU TO 185
 173 IF (K4-5) 175,174,175
 174 K4=4
 GU TO 185
 175 IF (K4-4) 177,176,177

```

176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    KORD(1,M)=0
    KORD(2,M)=0
    KORD(3,M)=0
    KORD(4,M)=0
40 CONTINUE
    STOP
    END
    END

```

```

12 12 9 4
12 12 9 4
12 12 9 4
9 9 9 1
9 8 4 4
9 9 9 1
12 11 9 4
9 9 9 1
10 9 5 5

```


0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	2	2	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	6	5	5	6	4	0
4	2	0	4	4	0	7	6	0	7	7	0	7	7	0	3	3	6	9	11	11	9	9	0	0	
12	12	2	6	4	6	8	7	8	9	7	9	11	9	5	9	7	7	9	12	12	9	9	0	0	

• GILBERT ROLLER - - - - - NO. 6 - - - - - RESULTS

8	8	6	6	6	8	6	8	5	7	7	5	8	5	7	7	8	8	6	5	6	8	6
8	8	6	-6	6	8	6	-6	5	5	-6	4	-6	5	5	-6	7	7	6	-5	6	7	5
6	6	6	3	6	6	6	3	-3	5	2	-2	3	-3	5	2	4	4	-5	3	-5	4	1
3	3	-2	3	-2	3	-2	1	1	1	1	1	1	1	1	1	1	2	2	3	2	3	1


```

* 74981T KULLER          3MIN  1C UP
PROGRAM WUJN7
DIMENSION KURU (4,67)
4 FORMAT (1H0,4X,17HCHUKU PROGRESSION)
PRINT 4
1 FORMAT(4(14))
HEAD 1, ((KURU (1,N),I=1,4),N=1,67)
41 FORMAT (1H0,4(14))
5 FORMAT (1H0, 3HEND)
L=1
J=1
K=1
I=0
M=0
10 K1=KURU(1,J)
K2=KURU(2,J)
K3=KURU (3,J)
K4= KURU (4,J)
L=L+1
30 L=L+1
L1= KURU (1,L)
L2= KURU (2,L)
L3=KURU(3,L)
L4= KURU (4,L)
IF (K1-L1) 34,33,32
32 IF (1-1) 37,40,39
33 IF (K2-L2) 34,34,32
34 IF (K3-L3) 34,35,32
35 IF (K4-L4) 34,2,32
2 IF (K1-0) 53,46,53
46 IF (K2-0) 53,47,53
47 IF (K3-0) 53,3,53
3 IF (K4-0) 53,37,53

```

53 IF (K1-12) 61,60,61
 60 K1=8
 GO TO 100
 61 IF (K1-11) 63,62,63
 62 K1=7
 GO TO 100
 63 IF (K1-10) 65,64,65
 64 K1=7
 GO TO 100
 65 IF (K1-9) 67,66,67
 66 K1=6
 GO TO 100
 67 IF (K1-8) 69,68,69
 68 K1=6
 GO TO 100
 69 IF (K1-7) 71,70,71
 70 K1=5
 GO TO 100
 71 IF (K1-6) 73,72,73
 72 K1=5
 GO TO 100
 73 IF (K1-5) 75,74,75
 74 K1=4
 GO TO 100
 75 IF (K1-4) 77,76,77
 76 K1=3
 GO TO 100
 77 IF (K1-3) 79,78,79
 78 K1=3
 GO TO 100
 79 IF (K1-2) 81,80,81
 80 K1=2
 GO TO 100

81 IF (K1-1)	83.82.83
82 K1=-2	
GO TO 100	
83 IF (K1-0)	100.84.100
84 K1=1	
100 IF (K2-12)	102.101.102
101 K2=8	
GO TO 130	
102 IF (K2-11)	104.103.104
103 K2=7	
GO TO 130	
104 IF (K2-10)	106.105.106
105 K2=-7	
GO TO 130	
106 IF (K2-9)	108.107.108
107 K2=6	
GO TO 130	
108 IF (K2-8)	110.109.110
109 K2=-6	
GO TO 130	
110 IF (K2-7)	112.111.112
111 K2=5	
GO TO 130	
112 IF (K2-6)	114.113.114
113 K2=-5	
GO TO 130	
114 IF (K2-5)	116.115.116
115 K2=4	
GO TO 130	
116 IF (K2-4)	118.117.118
117 K2=3	
GO TO 130	

118 IF (K2-3)	120,119,120
119 K2=-3	
GO TO 130	
120 IF (K2-2)	122,121,122
121 K2=2	
GO TO 130	
122 IF (K2-1)	124,123,124
123 K2=-2	
GO TO 130	
124 IF (K2-0)	130,125,130
125 K2=1	
130 IF (K3-12)	132,131,132
131 K3=8	
GO TO 156	
132 IF (K3-11)	134,133,134
133 K3=7	
GO TO 156	
134 IF (K3-10)	136,135,136
135 K3=-7	
GO TO 156	
136 IF (K3-9)	138,137,138
137 K3=6	
GO TO 156	
138 IF (K3-8)	140,139,140
139 K3=-6	
GO TO 156	
140 IF (K3-7)	142,141,142
141 K3=5	
GO TO 156	
142 IF (K3-6)	144,143,144
143 K3=-5	
GO TO 156	

144 IF (K3-5)	146,145,146
145 K3=4	
GO TO 156	
146 IF (K3-4)	148,147,148
147 K3=3	
GO TO 156	
148 IF (K3-3)	150,149,150
149 K3=-3	
GO TO 156	
150 IF (K3-2)	152,151,152
151 K3=2	
GO TO 156	
152 IF (K3-1)	154,153,154
153 K3=-2	
GO TO 156	
154 IF (K3-0)	156,155,156
155 K3=1	
156 IF (K4-12)	161,160,161
160 K4=8	
GO TO 185	
161 IF (K4-11)	163,162,163
162 K4=7	
GO TO 185	
163 IF (K4-10)	165,164,165
164 K4=-7	
GO TO 185	
165 IF (K4-9)	167,166,167
166 K4=6	
GO TO 185	
167 IF (K4-8)	169,168,169
168 K4=-6	
GO TO 185	
169 IF (K4-7)	171,170,171
170 K4=5	


```

GO TO 185
171 IF (K4-6) 173,172,173
172 K4=-5
GO TO 185
173 IF (K4-5) 175,174,175
174 K4=4
GO TO 185
175 IF (K4-4) 177,176,177
176 K4=3
GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
M=M+1
GO TO 36
20 I=0
J=J-1
L=L-1
GO TO 21
36 J=J+1
I=I+1
21 K1=KORD(1,J)
K2=KORD(2,J)

```

```

K3=KORD(3,J)
K4=KORD(4,J)
37 IF (L-67) 38,39,39
38 GO TO 30
39 J=J+1
I=0
L=J
IF (M-1) 7,6,6
6 PRINT 5
M=0
7 IF (L-67) 40,42,42
40 GO TO 10
42 GO TO 43
43 STOP
END
END

```

12	12	9	4
12	12	9	4
12	12	9	4
9	9	9	1
9	8	4	4
9	9	9	1
12	11	9	4
9	9	9	1
10	9	5	5
9	9	9	1
12	12	9	4
12	12	9	4
9	9	9	1
9	8	4	4
9	9	9	1
12	12	9	4
12	8	4	0
12	8	4	0

• • • • •

● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

0440760770770330

2 6 4 6 2 7 3 9 7 9 1 9 5 9 7 0

* QUINTET - NO. 7 - RESULTS

CHOED PROGRESSION

8	8	6	3
8	8	6	3
8	8	6	3

END

6	6	6	-2
6	6	6	-2
6	6	6	-2
6	6	6	-2
6	-6	3	3
6	6	6	-2

END

6	6	6	-2
6	6	6	-2
6	6	6	-2

END

6	6	6	-2
6	6	6	-2
8	8	6	3

END

6	6	6	-2
---	---	---	----

END

8 8 5 7 7 5 7 7 5
 -6 -6 5 5 -6 4 -6 5 5
 3 3 -3 5 2 -2 2 5 -3
 1 1 1 1 1 1 1 1 1

END

5 5 -3 1

END

-7 5 5 1

END

7 -6 2 1

END

8 8 8 8
 7 7 7 7
 4 4 4 4
 2 2 2 2

END

6 6 6
 -5 -5 -5
 3 3 3

END

5 -5 3 2
6 6 -5 3
8 7 4 2

END

6 6 -5 3

END

6 5 1 1

END



118 IF (K2-3)	120,119,120
119 K2=-3	
GO TO 130	
120 IF (K2-2)	122,121,122
121 K2=2	
GO TO 130	
122 IF (K2-1)	124,123,124
123 K2=-2	
GO TO 130	
124 IF (K2-0)	130,125,130
125 K2=1	
130 IF (K3-12)	132,131,132
131 K3=8	
GO TO 156	
132 IF (K3-11)	134,133,134
133 K3=7	
GO TO 156	
134 IF (K3-10)	136,135,136
135 K3=-7	
GO TO 156	
136 IF (K3-9)	138,137,138
137 K3=6	
GO TO 156	
138 IF (K3-8)	140,139,140
139 K3=-6	
GO TO 156	
140 IF (K3-7)	142,141,142
141 K3=5	
GO TO 156	
142 IF (K3-6)	144,143,144
143 K3=-5	
GO TO 156	

144 IF (K3-5)	146,145,146
145 K3=4	
GO TO 156	
146 IF (K3-4)	148,147,148
147 K3=3	
GO TO 156	
148 IF (K3-3)	150,149,150
149 K3=-3	
GO TO 156	
150 IF (K3-2)	152,151,152
151 K3=2	
GO TO 156	
152 IF (K3-1)	154,153,154
153 K3=-2	
GO TO 156	
154 IF (K3-0)	156,155,156
155 K3=1	
156 IF (K4-12)	161,160,161
160 K4=8	
GO TO 185	
161 IF (K4-11)	163,162,163
162 K4=7	
GO TO 185	
163 IF (K4-10)	165,164,165
164 K4=-7	
GO TO 185	
165 IF (K4-9)	167,166,167
166 K4=6	
GO TO 185	
167 IF (K4-8)	169,168,169
168 K4=-6	
GO TO 185	
169 IF (K4-7)	171,170,171
170 K4=5	


```
GO TO 185
171 IF (K4-6) 173,172,173
172 K4=-5
GO TO 185
173 IF (K4-5) 175,174,175
174 K4=4
GO TO 185
175 IF (K4-4) 177,176,177
176 K4=3
GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
M=M+1
GO TO 36
20 I=0
J=J-1
L=L-1
GO TO 21
36 J=J+1
I=I+1
21 K1=KORD(1,J)
K2=KORD(2,J)
```

```

K3=KORD(3,J)
K4=KORD(4,J)
37 IF (L-67) 38,39,39
38 GO TO 30
39 J=J+1
I=0
L=J
IF (M-1) 7,6,6
6 PRINT 5
M=0
7 IF (L-67) 40,42,42
40 GO TO 10
42 GO TO 43
43 STOP
END
END

```

12	12	9	4
12	12	9	4
12	12	9	4
9	9	9	1
9	8	4	4
9	9	9	1
12	11	9	4
9	9	9	1
10	9	5	5
9	9	9	1
12	12	9	4
12	12	9	4
9	9	9	1
9	8	4	4
9	9	9	1
12	12	9	4
12	8	4	0
12	8	4	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9	7	2	1	2	7	9	4	4	3	7	2	1	2	7	9	5	5	5	6	4	6	5	4	4	6	5	5	6	4	0	0	0
7	7	8	5	8	7	7	8	8	7	7	8	5	8	7	7	11	11	11	9	8	9	11	9	6	9	11	11	9	9	0	4	2
7	10	11	7	11	10	7	12	12	7	10	11	7	11	10	7	12	12	12	9	7	9	12	9	7	9	12	12	9	9	0	12	12

★ QUINTET - NO. 7 - RESULTS

CHORD PROGRESSION

8 8 6 3
8 8 6 3
8 8 6 3

END

6 6 6 -2
6 6 6 -2
6 6 6 -2
6 6 6 -2
6 -6 3 3
6 6 6 -2

END

6 6 6 -2
6 6 6 -2
6 6 6 -2

END

6 6 6 -2
6 6 6 -2
8 8 6 3

END

6 6 6 -2

END

1 1 1 1 1 1 1 1 1
 3 3 3 -3 5 2 -2 2 5 -3
 -6 -6 5 5 -6 4 -6 5 5
 8 8 5 -7 7 5 7 -7 5

END

5 5 -3 1

END

-7 5 5 1

END

7 -6 2 1

END

8 8 8 8
 7 7 7 7
 4 4 4 4
 2 2 2 2

END

6 6 6
 6 6 6
 -5 -5 -5
 3 3 3

END

5 -5 3 2
6 6 -5 3
8 7 4 2

END

6 6 -5 3

END

6 5 1 1

END

```

* 74981T ROLLER D      2MIN IC  OP
PROGRAM QUIB
DIMENSION LUM(27),KORT(132),KORD(132),MU(2,16),MUS(2,16),
1MUSI(2,16),MUSIC(2,16),MOE(26),JOE(14),KOE(10),LOE(2),MITH(4),
3NO(2,16)
1 FORMAT (14)
2 FORMAT (16(13,12))
READ 2, ((MU(I,N),I=1,2),N=1,16)
READ 2, ((MUS(I,N),I=1,2),N=1,16)
READ 2, ((MUSI(I,N),I=1,2),N=1,16)
READ 2, ((MUSIC(I,N),I=1,2),N=1,16)
READ 2, ((NO(I,N),I=1,2),N=1,16)
11 FORMAT (13(13))
READ11, (MOE(I),I=1,13)
14 FORMAT (7(13))
READ14, (JOE(I),I=1,7)
16 FORMAT (5(13))
READ16, (KOE(I),I=1,5)
18 FORMAT (13)
READ18, (LOE(I),I=1,1)
DO 232 M=1,16
J=1
N=1
I=1
K=4
LUM(N)= MU(I,M)
N=N+1
LUM(N)= MUS(I,M)
N=N+1
LUM(N)= MUSI(I,M)
N=N+1
LUM(N)= MUSIC(I,M)
N=N+1

```



```

      LUM(N)=      NO(I,M)
      8 DO 7 I=1,K
        IF (LUM(N)) 4,3,4
      4 IF (LUM(I)) 5,3,5
      5 KORD(I) = LUM(I) - LUM(N)
        IF (KORD(I)) 10,7,7
      10 KORD(I)=-1*KORD(I)
        GO TO 7
      3 KORD(I)=0
      7 CONTINUE
        IF (N-2) 9,9,6
      6 K=K-1
        N=N-1
        GO TO 8
      9 N=12
        DO 183 I=1,10
          J=KORD(I)
          IF (J-N) 80,80,86
      80 KORT(I)=KORD(I)
          GO TO 182
      86 K=0
      163 N=N+88
          K=K+88
          IF (J-N) 164,164,165
      164 GO TO 170
      165 N=N+12
          K=K+12
          IF (J-N) 166,166,167
      166 GO TO 170
      167 IF (L-6) 168,170,170
      168 L=L+1
          GO TO 163
      170 KORT(I)=KORD(I)-K

```

```

182 N=12
183 CONTINUE
    L=1
186 LEW=3
    N=5
    K=4
    IT =1
    DO 66 I=1,10
        IF (KORT(I)-MOE(L)) 140,20,140
20    JROOT=KORT(I)
        DO 32 J=1,7
            IF (JROOT-JOE(J)) 32,31,32
31    MITH(1) =LUM(N)
        GO TO 68
32 CONTINUE
        DO 34 J=1,5
            IF (JROOT-KOE(J)) 34,33,34
33    MITH (1) = LUM(IT)
        GO TO 68
34 CONTINUE
        IF (JROOT-LOE(1)) 140,35,140
35    MITH(1)=0
        GO TO 68
140 IF (I-K) 21,22,66
21 IT=IT+1
22 K=K+LEW
    N=N-1
    IT=IT-LEW
    LEW=LEW-1
66 CONTINUE
67 IF (L-13) 69,68,68
69 L=L+1

```


SECRET - 100-111403 - 100-111403

309
309
309
303
417
303
309
303
407
303
309
309
409
409
409
309
409

```

* 74981T ROLLER (GUI9)      2MIN IC  OP
PROGRAM GUIN9
DIMENSION MUS(3,64),NOTE(3,64),KLANG(3,64),LONG(3,64),KAN(64),
1KON(64),KUN(64),KEN(64),KIN(64),MITH(64)
2 FORMAT (16(13,11,11))
3 FORMAT (16(14))
READ 2, ((MUS(I,N),I=1,3),N=1,64)
READ 2, ((NOTE(I,N),I=1,3),N=1,64)
READ 2, ((KLANG(I,N),I=1,3),N=1,64)
READ 2, ((LONG(I,N),I=1,3),N=1,64)
READ 3, (MITH(I),I=1,64)
13 FORMAT (1H0,5X,17HMELODIC INTERVALS)
14 FORMAT (1H0,5X,8H1 VIOLIN)
15 FORMAT (1H0,16(14))
16 FORMAT (1H0,5X,8H2 VIOLIN)
17 FORMAT (1H0,5X, 5HVIOLA)
18 FORMAT (1H0,5X,5HCCELLO)
19 FORMAT (1H0,5X,15HHINDEMITH ROOTS)
IAN=0
MAN=0
DO 8 N=1,64
IAN=MAN
JAN=MUS(1,N)
IF (IAN) 30,20,30
30 IF (JAN) 21,20,21
20 KAN(N)=0
MAN = JAN
GO TO 8
21 KAN(N)=IAN-JAN
MAN=JAN
8 CONTINUE
IAN=0
MAN=0

```



```

DO 9 N=1,64
  IAN=MAN
  JAN=NOTE(1,N)
  IF (IAN) 31,22,31
31 IF (JAN) 23,22,23
22 KON(N)=0
  MAN=JAN
  GO TO 9
23 KON(N)=IAN-JAN
  MAN=JAN
  9 CONTINUE
  IAN=0
  MAN=0
  DO 10 N=1,64
    IAN=MAN
    JAN=KLANG(1,N)
    IF (IAN) 32,24,32
32 IF (JAN) 25,24,25
24 KUN(N)=0
  MAN=JAN
  GO TO 10
25 KUN(N)=IAN-JAN
  MAN=JAN
  10 CONTINUE
  IAN=0
  MAN=0
  DO 11 N=1,64
    IAN=MAN
    JAN=LONG(1,N)
    IF (IAN) 33,26,33
33 IF (JAN) 27,26,27
26 KEN(N)=0
  MAN=JAN
  GO TO 11

```

```

27 KEN(N)=IAN-JAN
   MAN=JAN
11 CONTINUE
   IAN=0
   MAN=0
   DO 12 N=1,64
     IAN=MAN
     JAN=MITH(N)
     IF (IAN) 34,28,34
34 IF (JAN) 29,28,29
28 KIN(N)=0
   MAN=JAN
   GO TO 12
29 KIN(N)=IAN-JAN
   MAN = JAN
12 CONTINUE
   PRINT 13
   PRINT 14
   PRINT 15, (KAN(I),I=1,64)
   PRINT 16
   PRINT 15, (KUN(I),I=1,64)
   PRINT 17
   PRINT 15, (KUN(I),I=1,64)
   PRINT 18
   PRINT 15, (KEN(I),I=1,64)
   PRINT 19
   PRINT 15, (KIN(I),I=1,64)
   STOP
   END
   END

505515055151251507515125150454512515075151251505515055151251507515125150551
5075150751509515105151251602615125151051509515075150551507515051510515125151051
5055150751505515055150551512515075151251505515125150551505515125150751

```

512510000100001000010000100001000010000100001000010000100001
40542405424054241242407424124240542412424054241242407424124240542
503525035250525073250952510525095250752505250352501525035250552507525085250752
505525035241042410424104241042410424104241042410424104241042410424104241042
41242000025055250525075250752507525065250852508525075250752507525055250552
309333093330933404433113340443309334044331133404433093340443311334044330933
41143411434094340743406434044340643407434094341143409434074340543404434024340443
405434074340443404434074340543407434044340743405434074340443404434074340543
4074300003409434074300003409434074300003409434074300003409434074340743
40244402444312343123431234402443123431234312344024431234312343123440244
0000400004000040000400004000040000400004000040000400004000040000400004
000040000407444074440744409444094440744409444074440944407444094440944
4094400004000040000400004000040000400004000040000400004000040000400004
205 205 205 205 203 203 203 205 203 202 203 203 205 205 205 203 203 203 205
205 205 205 203 203 203 205 203 202 203 202 203 205 205 205 203 203 203 205
205 205 205 203 203 203 205 203 202 203 202 203 205 205 205 203 203 203 205
205 205 205 203 203 203 205 203 202 203 202 203 205 205 205 203 203 203 205


```

* 74981T ROLLER F          IMIN IC  OP
PROGRAM QUI10
  DIMENSION KAN(320),MELO(30)
  30 FORMAT (16(I4))
  READ 30, (KAN(I),I=1,320)
  105 FORMAT (1H0.4H P1.9H MINOR 2.4H M2.9H MINOR 3.4H M3.4H P4.
    14H A4.4H P5.9H MINOR 6.4H M6.9H MINOR 7.4H M7.4H P8)
  106 FORMAT (1H0.1X.4(13.3X).3X.3(13.1X).13.3(3X.13).4X.13.2X.13)
  107 FORMAT (1H0.17HMELODIC INTERVALS)
  111 FORMAT (1H0.8H1 VIOLIN)
  112 FORMAT (1H0.9H11 VIOLIN)
  113 FORMAT (1H0.5HVIOLA)
  114 FORMAT (1H0.5HCCELLO)
  115 FORMAT (1H0.15HHINDEMTIH ROOTS)
  PRINT 111
  M=1
  N=12
  NO = 1
  NOT=64
  86 IT =1
  31 DO 29 I= NO,NOT
    L=1
    51 IF (KAN(I)) 40.41.41
    40 GO TO 52
    41 J=KAN(I)
    IF (J-N) 50.50.57
    50 KAN(I)=KAN(J)
    GO TO 76
    57 K=0
    47 N=N+88
    K=K+88
    IF (J-N) 58.58.59

```



```

58 GO TO 70
59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 KAN(I) = KAN(I) - K
76 N=12
   GO TO 29
52 J=KAN(I)
   IF (J+N) 54,53,53
53 KAN(I) = KAN(I)
   GO TO 17
54 K=0
10 N=N+88
   K=K+88
   IF (J+N) 12,11,11
11 GO TO 16
12 N=N+12
   K=K+12
   IF (J+N) 14,13,13
13 GO TO 16
14 IF (L-6) 15,16,16
15 L=L+1
   GO TO 10
16 KAN(I)=KAN(I)+K
17 N=12
29 CONTINUE
   PRINT 107
   PRINT 106, (KAN(I),I=NO,NOT)
   DO 6 I=1,26
     MELO(I) = 0

```



```

6 CONTINUE
100 MOT = 0
88 DO 83 I= 1,13
   IF (KAN(IT)-MOT) 93,82,93
82 MELO(I) = MELO(I)+1
93 MOT = MOT+1
83 CONTINUE
   MOT = 0
   DO 85 I= 14,26
     IF (KAN(IT) + MOT) 95,84,95
84 MELO(I) = MELO(I) - 1
95 MOT = MOT+1
85 CONTINUE
     IF (IT- NOT) 20,87,87
20 IT = IT +1
   GO TO 100
87 PRINT 105
   PRINT 106, (MELO(I),I=1,13)
   PRINT 106, (MELO(I),I=14,26)
   IF (M-5) 18,18,110
18 NO=NO+64
   NOT=NOT+64
   N=12
   M=M+1
   IF (M-3) 116,117,118
116 PRINT 112
   GO TO 125
117 PRINT 113
   GO TO 125
118 IF (M-5) 119,120,110
119 PRINT 114
   GO TO 125

```


QUINTET - NO. 10 - MELODIC INTERVALS AND ROOT MOVEMENT - RESULTS

I VIOLIN

MELODIC INTERVALS

0	0	0	-7	5	-5	8	-8	5	-5	7	0	-7
5	-5	7	-2	0	-2	-1	-2	-2	2	2	1	2
2	-2	-1	-2	-2	2	2	1	2	0	0	-7	5
-5	7	-7	5	-5	7	0	7	5	-5	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

P1	MINOR 2	M2	MINOR 3	M3	P4	A4	P5	MINOR 6	M6	MINOR 7	M7	P8
23	2	7	0	0	6	0	4	1	0	0	0	0
-23	-2	-7	0	0	-6	0	-5	-1	0	0	0	0

II VIOLIN

MELODIC INTERVALS

0	0	0	-7	5	-5	7	-7	5	-5	7	0	-7
5	-5	7	-10	0	-2	-2	-2	-1	1	2	2	2
2	-2	-2	-2	-1	1	2	2	5	0	0	-2	2
-2	2	-2	2	-2	2	0	-2	2	-2	0	0	0
0	-2	0	0	-1	0	0	1	0	0	2	0	0

P1	MINOR 2	M2	MINOR 3	M3	P4	A4	P5	MINOR 6	M6	MINOR 7	M7	P8
20	3	12	0	0	4	0	3	0	0	0	0	0
-20	-3	-13	0	0	-3	0	-3	0	0	-1	0	0

MELODIC INTERVALS

CELLO

MELODIC INTERVALS

P1	MINOR 2 M2	MINOR 3 M3	P4	A4	P5	MINOR 6 M6	MINOR 7 M7	P8
0	0	0	2	0	-2	0	0	2
0	0	-2	0	0	0	0	0	0
0	0	0	0	0	0	0	-2	0
0	2	-2	0	2	0	0	0	0
0	0	0	0	0	0	0	0	0
P1	MINOR 2 M2	MINOR 3 M3	P4	A4	P5	MINOR 6 M6	MINOR 7 M7	P8
53	0	6	0	0	0	0	0	0
-53	0	-6	0	0	0	0	0	0

FIFTEENTH ROOTS

MELODIC INTERVALS

0	0	0	2	0	0	-2	2	1	-1	-2	0	2
0	0	-2	0	0	0	2	0	0	-2	2	1	-1
-2	0	2	0	0	-2	0	0	0	2	0	0	-2
2	1	-1	-2	0	2	0	0	-2	0	0	0	2
0	0	-2	2	1	-1	-2	0	2	0	0	-2	0
P1	MINOR 2	M2	MINOR 3	M3	P4	M4	P5	MINOR 6	M6	MINOR 7	M7	P8
33	4	12	0	0	0	0	0	0	0	0	0	0
-33	-4	-12	0	0	0	0	0	0	0	0	0	0

```

* GILBERT ROLLER - - SYMPHONY OP.6
  DIMENSION
    MU(3,12),NOT(3,12),NOTE(3,12),KL(3,12),KLA(3,12),LU(3,12),
    2LON(3,12),LONG(3,12),LONGA(3,12),
    95 FORMAT (6(14))
  2 FORMAT (12(13,11,11))
  READ2,((MU(1,N),I=1,3),N=1,12)
  READ2,((MUS(1,N),I=1,3),N=1,12)
  READ 2,((MUSI(1,N),I=1,3),N=1,12)
  READ2,((MUSIC(1,N),I=1,3),N=1,12)
  READ 2,((NO(1,N),I=1,3),N=1,12)
  READ 2,((NOT(1,N),I=1,3),N=1,12)
  READ 2,((NOTE(1,N),I=1,3),N=1,12)
  READ 2,((KL(1,N),I=1,3),N=1,12)
  READ 2,((KLA(1,N),I=1,3),N=1,12)
  READ 2,((LO(1,N),I=1,3),N=1,12)
  READ 2,((LON(1,N),I=1,3),N=1,12)
  READ 2,((LONG(1,N),I=1,3),N=1,12)
  READ 2,((LONGA(1,N),I=1,3),N=1,12)
  M=0
  J=1
  10 M=M+1
  N=1
  I=1
  LUM(N)= MU(I,M)
  N=N+1
  LUM(N)= MUS(I,M)
  N=N+1
  LUM(N)= MUSI(I,M)
  N=N+1
  LUM(N)= MUSIC(I,M)
  N=N+1

```

```

LUM(N)=      NO(I,M)
N=N+1
LUM(N)=      NOT(I,M)
N=N+1
LUM(N)=      NOTE(I,M)
N=N+1
LUM(N)=      KL(I,M)
N=N+1
LUM(N)=      KLA(I,M)
N=N+1
LUM(N)=      LO(I,M)
N=N+1
LUM(N)=      LON(I,M)
N=N+1
LUM(N)=      LONG(I,M)
N=N+1
LUM(N)=      LONGA(I,M)
L = N-12
3 IF (LUM(N)-0) 12,4,12
4 N=N-1
  IF (N-L) 12,12,11
11 GO TO 3
12 L=N-L
13 IF (LUM(1)) 5,7,6
5 IF (LUM(N)-LUM(1)) 7,7,6
6 KE=LUM(N)
  KA= LUM(1)
  LUM(N)=KA
  LUM(1)=KE
7 I=I+1
  L=L-1
  IF (L-0) 9,9,8
8 GO TO 13

```


SYMPHONY - N/.1 - RESULTS

0200	100	100	100	100	100	0
0	100	0	100	100	3	3
200	100	100	100	100	3	3
3	100	3	100	100	0	0
208	108	108	108	108	5	5
5	105	5	108	108	5	5
210	302	302	302	302	2	2
2	100	2	110	110	102	102
209	302	302	302	302	2	2
2	100	2	109	109	102	102
206	206	207	206	206	0	0
0	100	0	106	106	102	102
297	297	297	297	297	0	0
0	190	90	197	197	192	192
503	403	0	0	0	0	0
0	0	0	403	0	0	0
508	408	0	0	0	0	0
0	0	0	403	0	0	0
203	200	200	200	200	0	0
0	100	0	100	103	95	95
203	200	200	200	200	3	3
3	100	3	103	103	0	0
295	203	203	203	203	0	0
0	100	0	103	195	3	3

```

*      GILBERT ROLLER - SYMPHONY - REDUCTION
      DIMENSION KORD(144),KORT1(12),KORT2(12),KORT3(12),KORT4(12),
      1KORT5(12),KORT6(12),KORT7(12),KORT8(12),KORT9(12),KORT10(12),
      2KORT11(12),KORT12(12)
      1 FORMAT (6(I4))
      READ 1, (KORD(I),I=1,144)
      2 FORMAT (1H0,6(I4))
      M=1
      N=12
      IT =1
      KT=133
      5 DO 98 I=M,KT,12
      L=1
      J=KORD(I)
      IF (M-2) 11,13,15
      11 IF (J-N) 12,12,45
      12 KORT1(IT) = KORD(I)
      GO TO 96
      13 IF (J-N) 14,14,45
      14 KORT2(IT) = KORD(I)
      GO TO 96
      15 IF (M-4) 16,16,20
      16 IF (J-N) 17,17,45
      17 KORT3(IT) = KORD(I)
      GO TO 96
      18 IF (J-N) 19,19,45
      19 KORT4(IT) = KORD(I)
      GO TO 96
      20 IF (M-6) 21,23,25
      21 IF (J-N) 22,22,45
      22 KORT5(IT) = KORD(I)
      GO TO 96
      23 IF (J-N) 24,24,45
      24 KORT6(IT) = KORD(I)

```

```

GO TO 96
25 IF (M-8) 26,28,30
26 IF (J-N) 27,27,45
27 KORT7(IT) = KORD(I)
GO TO 96
28 IF (J-N) 29,29,45
29 KORT8(IT) = KORD(I)
GO TO 96
30 IF (M-10) 31,33,35
31 IF (J-N) 32,32,45
32 KORT9(IT) = KORD(I)
GO TO 96
33 IF (J-N) 34,34,45
34 KORT10(IT) = KORD(I)
GO TO 96
35 IF (M-12) 36,38,98
36 IF (J-N) 37,37,45
37 KORT11(IT) = KORD(I)
GO TO 96
38 IF (J-N) 39,39,45
39 KORT12(IT) = KORD(I)
GO TO 96
45 K=0
47 N=N+88
K=K+88
IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
K=K+12
IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70

```



```
66 L=L+1
   GO TO 47
70 IF (M-2) 71,72,73
71 KORT1(IT) = KORD(I) -K
   GO TO 96
72 KORT2(IT) = KORD(I) -K
73 IF (M-4) 74,75,76
74 KORT3(IT) = KORD(I) -K
   GO TO 96
75 KORT4(IT) = KORD(I) -K
   GO TO 96
76 IF (M-6) 77,78,79
77 KORT5(IT) = KORD(I) -K
   GO TO 96
78 KORT6(IT) = KORD(I) -K
   GO TO 96
79 IF (M-8) 80,81,82
80 KORT7(IT) = KORD(I) -K
   GO TO 96
81 KORT8(IT) = KORD(I) -K
   GO TO 96
82 IF (M-10) 83,84,85
83 KORT9(IT) = KORD(I) -K
   GO TO 96
84 KORT10(IT)=KORD(I)-K
   GO TO 96
85 IF (M-12) 86,87,96
86 KORT11(IT)=KORD(I)-K
   GO TO 96
87 KORT12(IT)=KORD(I)-K
   GO TO 96
96 N=12
   IT = IT+1
98 CONTINUE
```

```

IT =1
M=M+1
KT=KT+1
N=12
IF (M-13)99,100,100
99 GO TO 5
100 PRINT 2, (KORT1(I),I=1,12)
PRINT 2, (KORT2(I),I=1,12)
PRINT 2, (KORT3(I),I=1,12)
PRINT 2, (KORT4(I),I=1,12)
PRINT 2, (KORT5(I),I=1,12)
PRINT 2, (KORT6(I),I=1,12)
PRINT 2, (KORT7(I),I=1,12)
PRINT 2, (KORT8(I),I=1,12)
PRINT 2, (KORT9(I),I=1,12)
PRINT 2, (KORT10(I),I=1,12)
PRINT 2, (KORT11(I),I=1,12)
PRINT 2, (KORT12(I),I=1,12)
PUNCH 1, (KORT1(I),I=1,12)
PUNCH 1, (KORT2(I),I=1,12)
PUNCH 1, (KORT3(I),I=1,12)
PUNCH 1, (KORT4(I),I=1,12)
PUNCH 1, (KORT5(I),I=1,12)
PUNCH 1, (KORT6(I),I=1,12)
PUNCH 1, (KORT7(I),I=1,12)
PUNCH 1, (KORT8(I),I=1,12)
PUNCH 1, (KORT9(I),I=1,12)
PUNCH 1, (KORT10(I),I=1,12)
PUNCH 1, (KORT11(I),I=1,12)
PUNCH 1, (KORT12(I),I=1,12)
STOP
END

```

0200	100	100	100	100	100	0
0	100	0	100	100	100	3
200	100	100	100	100	100	3
3	100	3	100	100	100	0
208	108	108	108	108	108	5
5	105	5	108	108	108	5
210	302	302	302	302	302	2
2	100	2	110	110	110	102
209	302	302	302	302	302	2
2	100	2	109	109	109	102
206	206	207	206	206	206	0
0	100	0	106	106	106	102
297	297	297	297	297	297	0
0	190	90	197	197	197	192
503	403	0	0	0	0	0
0	0	0	403	0	0	0
508	408	0	0	0	0	0
0	0	0	403	0	0	0
203	200	200	200	200	200	0
0	100	0	100	103	103	95
203	200	200	200	200	200	3
3	100	3	103	103	103	0
295	203	203	203	203	203	0
0	100	0	103	195	195	3

```

*      GILBERT ROLLER - SYMPHONY - SORTING - NO.4
      DIMENSION INTA(12),INTB(12),INTC(12),INTD(12),INTE(12),INTG(12),
1      INTH(12),INTJ(12),INTK(12),INTL(12),INTM(12),INTN(12),KOR(12)
149 FORMAT (12(I4))
      1 FORMAT (6(I4))
      READ 1, (INTA(I),I=1,12)
      READ 1, (INTB(I),I=1,12)
      READ 1, (INTC(I),I=1,12)
      READ 1, (INTD(I),I=1,12)
      READ 1, (INTE(I),I=1,12)
      READ 1, (INTG(I),I=1,12)
      READ 1, (INTH(I),I=1,12)
      READ 1, (INTJ(I),I=1,12)
      READ 1, (INTK(I),I=1,12)
      READ 1, (INTL(I),I=1,12)
      READ 1, (INTM(I),I=1,12)
      READ 1, (INTN(I),I=1,12)
      DO 61 I=1,12
      MA = INTA(I)
      MB = INTB(I)
      MC = INTC(I)
      MD = INTD(I)
      ME = INTE(I)
      MG = INTG(I)
      MH = INTH(I)
      MJ = INTJ(I)
      MK = INTK(I)
      ML = INTL(I)
      MM = INTM(I)
      MN = INTN(I)
      DO 60 K=1,12
      KORTS =0
      IF (KORTS - MA) 12,13,13

```


12 KORTS = MA
13 IF (KORTS - MB) 14,15,15
14 KORTS = MB
15 IF (KORTS - MC) 16,17,17
16 KORTS = MC
17 IF (KORTS - MD) 18,19,19
18 KORTS = MD
19 IF (KORTS - ME) 20,21,21
20 KORTS = ME
21 IF (KORTS - MG) 22,23,23
22 KORTS = MG
23 IF (KORTS - MH) 24,25,25
24 KORTS = MH
25 IF (KORTS - MJ) 26,27,27
26 KORTS = MJ
27 IF (KORTS - MK) 28,29,29
28 KORTS = MK
29 IF (KORTS - ML) 30,31,31
30 KORTS = ML
31 IF (KORTS - MM) 32,33,33
32 KORTS = MM
33 IF (KORTS - MN) 34,35,35
34 KORTS = MN
35 IF (KORTS - MA) 37,36,37
36 MA = 0
GO TO 59
37 IF (KORTS - MB) 39,38,39
38 MB = 0
GO TO 59
39 IF (KORTS - MC) 41,40,41
40 MC = 0
GO TO 59
41 IF (KORTS - MD) 43,42,43

```

42 MJ=0
   GO TO 59
43 IF (KORTS - ME)45,44,45
44 ME=0
   GO TO 59
45 IF (KORTS - MG)47,46,47
46 MG=0
   GO TO 59
47 IF (KORTS - MH)49,48,49
48 MH=0
   GO TO 59
49 IF (KORTS - MJ)51,50,51
50 MJ=0
   GO TO 59
51 IF (KORTS - MK)53,52,53
52 MK=0
   GO TO 59
53 IF (KORTS - ML)55,54,55
54 ML=0
   GO TO 59
55 IF (KORTS - MM)57,56,57
56 MM=0
   GO TO 59
57 IF (KORTS - MN)59,58,59
58 MN=0
59 KOR(K) = KORTS
60 CONTINUE
   PUNCH 149, (KOR(K),K=1,12)
61 CONTINUE
   STOP
   END

```



```

* 74981T ROLLER X          1MIN IC GP
PROGRAM GRITT
  DIMENSION KORD(144),KEEP(48)
  1 FORMAT (12(14))
  15 FORMAT (4(14))
  READ 1, (KORD(I),I=1,144)
  J=0
  I=1
  N=2
  K=12
  4 IF (KORD(I)-KORD(N)) 6,2,6
  2 I=I+2
  IF (N-K) 16,14,14
  14 J=J+1
  KEEP(J)=KORD(N)
  GO TO 5
  16 IF (KORD(N)-KORD(I)) 12,3,12
  3 N=N+2
  GO TO 4
  6 J=J+1
  KEEP(J)=KORD(I)
  IF (N-K) 2,17,17
  17 J=J+1
  KEEP(J)=KORD(N)
  GO TO 5
  12 J=J+1
  KEEP(J)=KORD(N)
  GO TO 3
  5 IF (J-4) 7,9,9
  7 J=J+1
  DO 8 I=J,4
  KEEP(I)=0
  8 CONTINUE

```

[illegible]


```

GILBERT ROLLER - SYM. RE-ARR)
PROGRAM SYMRE
DIMENSION KORD(12,12),INTERV(144)
1  FORMAT (12(14))
2  READ 1, ((KORD(I,N),I=1,12),N=1,12)
J=1
I=1
4  DO 5 N=1,12
    INTERV(J)=KORD(I,N)
    J=J+1
5  CONTINUE
    IF (I-12) 6,7,7
6  I = I+1
    GO TO 4
7  PRINT 1, (INTERV(I), I=1,144)
    STOP
END
END

```

[illegible]

SYMPHONY - NO. Va - REDUCTION OF LONG
USE QUARTET PROGRAM Va

```

* 74981T ROLLER (SY5)      5 MIN. 1C  OP
      PROGRAM SYM5
      DIMENSION INFO(2,36),MUS(3,12),NOTE(3,132),LONG(3,12),KORD(144)
      1 FORMAT (12(13,12))
      2 FORMAT (12(13,11,12))
      READ 1, ((INFO(I,N),I=1,2),N=1,36)
      READ 2, ((MUS(I,N),I=1,3),N=1,12)
      READ 2, ((NOTE(I,N),I=1,3),N=1,132)
      READ 2, ((LONG(I,N),I=1,3),N=1,12)
      3 FORMAT (12(14))
      READ 3, (KORD (I),I=1,144)
      20 FORMAT (1H0,17,115,16X,2H0 ,113,110/)
      270 FORMAT (1H0,17,115,16X,2H2-,113,110/)
      271 FORMAT (1H0,17,115,16X,2H2.,113,110/)
      272 FORMAT (1H0,17,115,16X,2H3-,113,110/)
      273 FORMAT (1H0,17,115,16X,2H3.,113,110/)
      274 FORMAT (1H0,17,115,16X,2H4.,113,110/)
      275 FORMAT (1H0,17,115,16X,2H4+,113,110/)
      276 FORMAT (1H0,17,115,16X,2H5.,113,110/)
      277 FORMAT (1H0,17,115,16X,2H6-,113,110/)
      278 FORMAT (1H0,17,115,16X,2H6.,113,110/)
      279 FORMAT (1H0,17,115,16X,2H7-,113,110/)
      280 FORMAT (1H0,17,115,16X,2H7.,113,110/)
      281 FORMAT (1H0,17,115,16X,2H8.,113,110/)
      21 FORMAT (41X,2H0 ,113,110/)
      285 FORMAT (41X,2H2-,113,110/)
      286 FORMAT (41X,2H2.,113,110/)
      287 FORMAT (41X,2H3-,113,110/)
      288 FORMAT (41X,2H3.,113,110/)
      289 FORMAT (41X,2H4.,113,110/)
      290 FORMAT (41X,2H4+,113,110/)
      291 FORMAT (41X,2H5.,113,110/)
      292 FORMAT (41X,2H6-,113,110/)

```



```

293 FORMAT (41X,2H6.,.113,.110/)
294 FORMAT (41X,2H7-,.113,.110/)
295 FORMAT (41X,2H7.,.113,.110/)
296 FORMAT (41X,2H8.,.113,.110/)
22  FORMAT (41X,2H0 ,.113,.110//)
315 FORMAT (41X,2HC ,.113,.110//)
316 FORMAT (41X,2HC+ ,.113,.110//)
317 FORMAT (41X,2HD ,.113,.110//)
318 FORMAT (41X,2HE- ,.113,.110//)
319 FORMAT (41X,2HE ,.113,.110//)
320 FORMAT (41X,2HF ,.113,.110//)
321 FORMAT (41X,2HF+ ,.113,.110//)
322 FORMAT (41X,2HG ,.113,.110//)
323 FORMAT (41X,2HA- ,.113,.110//)
324 FORMAT (41X,2HA ,.113,.110//)
325 FORMAT (41X,2HJ- ,.113,.110//)
326 FORMAT (41X,2HJ ,.113,.110//)
350 FORMAT (1H0,4HCODE/4X,21H. IS MAJOR OR PERFECT/4X,10H- IS MINOR/
14X,14H+ IS AUGMENTED)
PRINT 350
42  FORMAT (1H0,13HMEASURE NO. ,19HRHYTHMIC POSITION ,
120HHARMONIC INTERVALS ,8HOCTAVE ,12HINSTRUMENT )
41  PRINT 42
J=144
M=121
K=120
N=1
12  J=J-143
IF (KORD(J)-0) 4,5,4
5  PRINT 20,INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
4  IF (KORD(J)-2) 200,201,202
200 PRINT 270, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)

```

```

GO TO 217
201 PRINT 271, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
202 IF (KORD(J)-4) 203,204,205
203 PRINT 272, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
204 PRINT 273, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
205 IF (KORD(J)-6) 206,207,208
206 PRINT 274, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
207 PRINT 275, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
208 IF (KORD(J)-8) 209,210,211
209 PRINT 276, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
210 PRINT 277, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
211 IF (KORD(J)-10) 212,213,214
212 PRINT 278, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
213 PRINT 279, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
214 IF (KORD(J)-12) 215,216,217
215 PRINT 280, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
GO TO 217
216 PRINT 281, INFO(1,N), INFO(2,N), MUS(2,N), MUS(3,N)
217 J=J+12
9 IF (KORD(J)-0) 0,7,6
7 PRINT 21, NOTE(2,N), NOTE(3,N)
GO TO 235
6 IF (KORD(J)-2) 218,219,220

```

```

218 PRINT 285, NOTE(2,N),NOTE(3,N)
   GO TO 235
219 PRINT 286, NOTE(2,N),NOTE(3,N)
   GO TO 235
220 IF (KORD(J)-4) 221,222,223
221 PRINT 287, NOTE(2,N),NOTE(3,N)
   GO TO 235
222 PRINT 288, NOTE(2,N),NOTE(3,N)
   GO TO 235
223 IF (KORD(J)-6) 224,225,226
224 PRINT 289, NOTE(2,N),NOTE(3,N)
   GO TO 235
225 PRINT 290, NOTE(2,N),NOTE(3,N)
   GO TO 235
226 IF (KORD(J)-8) 227,228,229
227 PRINT 291, NOTE(2,N),NOTE(3,N)
   GO TO 235
228 PRINT 292, NOTE(2,N),NOTE(3,N)
   GO TO 235
229 IF (KORD(J)-10) 230,231,232
230 PRINT 293, NOTE(2,N),NOTE(3,N)
   GO TO 235
231 PRINT 294, NOTE(2,N),NOTE(3,N)
   GO TO 235
232 IF (KORD(J)-12) 233,234,235
233 PRINT 295, NOTE(2,N),NOTE(3,N)
   GO TO 235
234 PRINT 296, NOTE(2,N),NOTE(3,N)
235 J=J+12
   IF (N-M) 8,253,255
      8 N=N+12
   GO TO 9
253 M=M+1

```

```

N=N-K
  IF (LONG(1,N) -0) 10,11,10
11 PRINT 22, LONG(2,N),LONG(3,N)
   GO TO 302
  10 IF (LONG(1,N)-2) 255,256,257
255 PRINT 315, LONG(2,N),LONG(3,N)
   GO TO 302
256 PRINT 316, LONG(2,N),LONG(3,N)
   GO TO 302
257 IF (LONG(1,N) -4) 258,259,260
258 PRINT 317, LONG(2,N),LONG(3,N)
   GO TO 302
259 PRINT 318, LONG(2,N),LONG(3,N)
   GO TO 302
260 IF (LONG(1,N) -5) 261,262,263
261 PRINT 319, LONG(2,N),LONG(3,N)
   GO TO 302
262 PRINT 320, LONG(2,N),LONG(3,N)
   GO TO 302
263 IF (LONG (1,N) -6) 264,265,266
264 PRINT 321, LONG(2,N),LONG(3,N)
   GO TO 302
265 PRINT 322, LONG(2,N),LONG(3,N)
   GO TO 302
266 IF (LONG(1,N) -10) 267,268,269
267 PRINT 323, LONG(2,N),LONG(3,N)
   GO TO 302
268 PRINT 324, LONG(2,N),LONG(3,N)
   GO TO 302
269 IF (LONG(1,N)-12) 327,328,302
327 PRINT 325, LONG (2,N),LONG(3,N)
   GO TO 302
328 PRINT 326, LONG (2,N),LONG(3,N)
302 IF (N-13) 13,14,14

```


* SYMPHONY - NO. 5 - PRINT OUT - RESULTS

CODE

• IS MAJOR OR PERFECT
- IS MINOR
/ IS AUGMENTED

MEASURE NO.	RHYTHMIC POSITION	MEASURE NO.	IC INTERVALS	OCTAVE	INSTRUMENT
1	1		8.	5	1
			8.	4	2
			8.	4	3
			8.	4	4
			8.	4	5
			8.	3	6
			8.	3	7
			8.	4	8
			3-	3	9
			0	4	10
			0	4	11
			0	3	12
			0	3	13
1	2		8.	5	1
			8.	4	2
			8.	4	3
			8.	4	4
			8.	4	5
			8.	3	6
			8.	3	7
			8.	4	8
			3-	3	9
			3-	4	10

[illegible]

4 3 3 5 4 4 4 4 3 3 4 3 4 4 3 3 5 6 6 6 6 3 3 4 3 4 4 4 3

3- C B- 6- 6- 6- 6- 6- 6- 4- 4- 4- 4- 4- G 8- 7- 7- 7- 2- 2- 2- 2- 2- 2- 2- D

3

I

1

2

1 2 3 4 5 6 7 8 9 10 11 12 13

5 6 6 6 6 3 3 4 3 4 4 4 3

8. 6. 6. 6. 2. 2. 2. 2. 2. 2. 2. D

2

2

1 2 3 4 5 6 7 8 9 10 11 12 13

5 5 5 5 5 3 3 4 3 4 4 4 3

8. 5. 4. 4. 4. 4. 4. 4. 2. 0 0 0 0

3

2

1 2 3

5 5 5

6. 6. 6.

1

3

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|----|----|----|----|---|----|----|----|---|---|---|---|
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 5 | 5 | 2 | 2 | 4 | 3 | 4 | 4 | 4 | 2 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6. | 6. | 6. | 6. | 3. | 2. | 2. | 0 | 0 | B- | 3- | 3- | 3- | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6- | 6- | 3- | 0 | 0 | 0 | 0 |

2

3

3

3

8 9 10 11 12 13

1 2 3 4 5 6 7 8 9 10 11 12 13

1 2 3 4 5 6 7 8 9 10 11 12 13

004700

5 5 5 5 5 3 3 4 3 4 4 4 3

555553343443₃

○ ○ ○ ○ ○ ○

ထိုစဉ်က နေပြည်တော်ရှိ အမတ်ချုပ်ရုံးတွင်

ထိုထိုထိုထိုထိုကကကကကကကကကက

I

7

2

7

1 2 3 4 5 6 7 8 9
10 11 12 13

6 5 5 5 5 5 3 3 4 3 4 5 3 3

8. 5. 5. 3- 3- 3- 3- 3- 3- 0 0 0 0

3

4

```

* 74981T ROLLER Q          1 MINIC  OP
      PROGRAM  SYM6
      DIMENSION KORD(4,12)
41  FORMAT (4(I4))
      1  FORMAT (4(I4))
      READ 1, ((KORD(1,N),I=1,4),N=1,12)
      J=1
      DO 40 N=1,12
      K1=KORD(1,N)
      K2=KORD(2,N)
      K3=KORD(3,N)
      K4=KORD(4,N)
      J=J+1
      DO 40 M=J,12
      L1=KORD(1,M)
      L2=KORD(2,M)
      L3=KORD(3,M)
      L4=KORD(4,M)
      IF (K1-L1) 31,32,31
31  GO TO 40
32  IF (K2-L2) 31,34,31
34  IF (K3-L3) 31,35,31
35  IF (K4-L4) 31,36,31
36  IF (K1-0) 43,38,43
37  IF (K2-0) 43,38,43
38  IF (K3-0) 43,39,43
39  IF (K4-0) 43,40,43
43  IF (K1-12) 61,60,61
60  K1=8
      GO TO 100
61  IF (K1-11) 63,62,63

```


| | | |
|----|------------|------------|
| 62 | K1=7 | |
| | GO TO 100 | |
| 63 | IF (K1-10) | 65,64,65 |
| 64 | K1=-7 | |
| | GO TO 100 | |
| 65 | IF (K1-9) | 67,66,67 |
| 66 | K1=6 | |
| | GO TO 100 | |
| 67 | IF (K1-8) | 69,68,69 |
| 68 | K1=-6 | |
| | GO TO 100 | |
| 69 | IF (K1-7) | 71,70,71 |
| 70 | K1=5 | |
| | GO TO 100 | |
| 71 | IF (K1-6) | 73,72,73 |
| 72 | K1=-5 | |
| | GO TO 100 | |
| 73 | IF (K1-5) | 75,74,75 |
| 74 | K1=4 | |
| | GO TO 100 | |
| 75 | IF (K1-4) | 77,76,77 |
| 76 | K1=3 | |
| | GO TO 100 | |
| 77 | IF (K1-3) | 79,78,79 |
| 78 | K1=-3 | |
| | GO TO 100 | |
| 79 | IF (K1-2) | 81,80,81 |
| 80 | K1=2 | |
| | GO TO 100 | |
| 81 | IF (K1-1) | 83,82,83 |
| 82 | K1=-2 | |
| | GO TO 100 | |
| 83 | IF (K1-0) | 100,84,100 |
| 84 | K1=1 | |

| | |
|----------------|-------------|
| 100 IF (K2-12) | 102,101,102 |
| 101 K2=8 | |
| GO TO 130 | |
| 102 IF (K2-11) | 104,103,104 |
| 103 K2=7 | |
| GO TO 130 | |
| 104 IF (K2-10) | 106,105,106 |
| 105 K2=-7 | |
| GO TO 130 | |
| 106 IF (K2-9) | 108,107,108 |
| 107 K2=6 | |
| GO TO 130 | |
| 108 IF (K2-8) | 110,109,110 |
| 109 K2=-6 | |
| GO TO 130 | |
| 110 IF (K2-7) | 112,111,112 |
| 111 K2=5 | |
| GO TO 130 | |
| 112 IF (K2-6) | 114,113,114 |
| 113 K2=-5 | |
| GO TO 130 | |
| 114 IF (K2-5) | 116,115,116 |
| 115 K2=4 | |
| GO TO 130 | |
| 116 IF (K2-4) | 118,117,118 |
| 117 K2=3 | |
| GO TO 130 | |
| 118 IF (K2-3) | 120,119,120 |
| 119 K2=-3 | |
| GO TO 130 | |
| 120 IF (K2-2) | 122,121,122 |
| 121 K2=2 | |
| GO TO 130 | |
| 122 IF (K2-1) | 124,123,124 |

| | |
|----------------|-------------|
| 123 K2=-2 | |
| GO TO 130 | |
| 124 IF (K2=0) | 130,125,130 |
| 125 K2=1 | |
| 130 IF (K3=12) | 132,131,132 |
| 131 K3=8 | |
| GO TO 156 | |
| 132 IF (K3=11) | 134,133,134 |
| 133 K3=7 | |
| GO TO 156 | |
| 134 IF (K3=10) | 136,135,136 |
| 135 K3=-7 | |
| GO TO 156 | |
| 136 IF (K3=9) | 138,137,138 |
| 137 K3=6 | |
| GO TO 156 | |
| 138 IF (K3=8) | 140,139,140 |
| 139 K3=-6 | |
| GO TO 156 | |
| 140 IF (K3=7) | 142,141,142 |
| 141 K3=5 | |
| GO TO 156 | |
| 142 IF (K3=6) | 144,143,144 |
| 143 K3=-5 | |
| GO TO 156 | |
| 144 IF (K3=5) | 146,145,146 |
| 145 K3=4 | |
| GO TO 156 | |
| 146 IF (K3=4) | 148,147,148 |
| 147 K3=3 | |
| GO TO 156 | |
| 148 IF (K3=3) | 150,149,150 |
| 149 K3=-3 | |

GO TO 156
 150 IF (K3-2) 152,151,152
 151 K3=2
 GO TO 156
 152 IF (K3-1) 154,153,154
 153 K3=-2
 GO TO 156
 154 IF (K3-0) 156,155,156
 155 K3=1
 156 IF (K4-12) 161,160,161
 160 K4=8
 GO TO 185
 161 IF (K4-11) 163,162,163
 162 K4=7
 GO TO 185
 163 IF (K4-10) 165,164,165
 164 K4=-7
 GO TO 185
 165 IF (K4-9) 167,166,167
 166 K4=6
 GO TO 185
 167 IF (K4-8) 169,168,169
 168 K4=-6
 GO TO 185
 169 IF (K4-7) 171,170,171
 170 K4=5
 GO TO 185
 171 IF (K4-6) 173,172,173
 172 K4=-5
 GO TO 185
 173 IF (K4-5) 175,174,175
 174 K4=4
 GO TO 185
 175 IF (K4-4) 177,176,177

```

176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    KORD(1,M)=0
    KORD(2,M)=0
    KORD(3,M)=0
    KORD(4,M)=0
    40 CONTINUE
    STOP
    END
    END

```

```

12 3 0 0
12 3 0 0
8 5 0 0
12 10 2 0
12 9 2 0
12 7 6 2
9 4 2 0
3 0 0 0
8 3 0 0
12 7 3 0 0
12 3 0 0
12 7 3 0

```

* SYMPHONY - NO. 6 - CHILD REPETITION - RESULTS

| | | | |
|---|----|----|---|
| 8 | -3 | 1 | 1 |
| 8 | 5 | -3 | 1 |

```

* 74961T ROLLER T 3MIN IC OP
PROGRAM SYM7
DIMENSION KORD (4,12)
4 FORMAT (1H0,4X,17HCHORD PROGRESSION)
PRINT 4
1 FORMAT(4(14))
READ 1, ((KORD (1,N),1=1,4),N=1,12)
41 FORMAT (1H0,4(14))
5 FORMAT (1H0, 3HEND)
L=1
J=1
K=1
I=0
M=0
10 K1=KORD(1,J)
K2=KORD(2,J)
K3=KORD (3,J)
K4= KORD (4,J)
L=L+1
30 L=L+1
L1= KORD (1,L)
L2= KORD (2,L)
L3=KORD(3,L)
L4= KORD (4,L)
IF (K1-L1) 32,33,32
32 IF (1-1) 37,20,39
33 IF (K2-L2) 32,34,32
34 IF (K3-L3) 32,35,32
35 IF (K4-L4) 32,2,32
2 IF (K1-0) 53,46,53
46 IF (K2-0) 53,47,53
47 IF (K3-0) 53,3,53
3 IF (K4-0) 53,37,53

```


53 IF (K1-12) 61,60,61
 60 K1=8
 GO TO 100
 51 IF (K1-11) 63,62,63
 62 K1=7
 GO TO 100
 63 IF (K1-10) 65,64,65
 64 K1=-7
 GO TO 100
 65 IF (K1-9) 67,66,67
 66 K1=6
 GO TO 100
 67 IF (K1-8) 69,68,69
 68 K1=-6
 GO TO 100
 69 IF (K1-7) 71,70,71
 70 K1=5
 GO TO 100
 71 IF (K1-6) 73,72,73
 72 K1=-5
 GO TO 100
 73 IF (K1-5) 75,74,75
 74 K1=4
 GO TO 100
 75 IF (K1-4) 77,76,77
 76 K1=3
 GO TO 100
 77 IF (K1-3) 79,78,79
 78 K1=-3
 GO TO 100
 79 IF (K1-2) 81,80,81
 80 K1=2
 GO TO 100
 81 IF (K1-1) 83,82,83

| | |
|----------------|-------------|
| 82 K1=-2 | |
| GO TO 100 | |
| 83 IF (K1=0) | 100,84,100 |
| 84 K1=1 | |
| 100 IF (K2=12) | 102,101,102 |
| 101 K2=8 | |
| GO TO 130 | |
| 102 IF (K2=11) | 104,103,104 |
| 103 K2=7 | |
| GO TO 130 | |
| 104 IF (K2=10) | 106,105,106 |
| 105 K2=-7 | |
| GO TO 130 | |
| 106 IF (K2=9) | 108,107,108 |
| 107 K2=6 | |
| GO TO 130 | |
| 108 IF (K2=8) | 110,109,110 |
| 109 K2=-6 | |
| GO TO 130 | |
| 110 IF (K2=7) | 112,111,112 |
| 111 K2=5 | |
| GO TO 130 | |
| 112 IF (K2=6) | 114,113,114 |
| 113 K2=-5 | |
| GO TO 130 | |
| 114 IF (K2=5) | 116,115,116 |
| 115 K2=4 | |
| GO TO 130 | |
| 116 IF (K2=4) | 118,117,118 |
| 117 K2=3 | |
| GO TO 130 | |
| 118 IF (K2=3) | 120,119,120 |
| 119 K2=-3 | |
| GO TO 130 | |

| | |
|----------------|-------------|
| 120 IF (K2-2) | 122,121,122 |
| 121 K2=2 | |
| GO TO 130 | |
| 122 IF (K2-1) | 124,123,124 |
| 123 K2=-2 | |
| GO TO 130 | |
| 124 IF (K2-0) | 130,125,130 |
| 125 K2=1 | |
| 130 IF (K3-12) | 132,131,132 |
| 131 K3=8 | |
| GO TO 156 | |
| 132 IF (K3-11) | 134,133,134 |
| 133 K3=7 | |
| GO TO 156 | |
| 134 IF (K3-10) | 136,135,136 |
| 135 K3=-7 | |
| GO TO 156 | |
| 136 IF (K3-9) | 138,137,138 |
| 137 K3=6 | |
| GO TO 156 | |
| 138 IF (K3-8) | 140,139,140 |
| 139 K3=-6 | |
| GO TO 156 | |
| 140 IF (K3-7) | 142,141,142 |
| 141 K3=5 | |
| GO TO 156 | |
| 142 IF (K3-6) | 144,143,144 |
| 143 K3=-5 | |
| GO TO 156 | |
| 144 IF (K3-5) | 146,145,146 |
| 145 K3=4 | |
| GO TO 156 | |
| 146 IF (K3-4) | 148,147,148 |

147 K3=3
 GO TO 156
 148 IF (K3-3) 150,149,150
 149 K3=-3
 GO TO 156
 150 IF (K3-2) 152,151,152
 151 K3=2
 GO TO 156
 152 IF (K3-1) 154,153,154
 153 K3=-2
 GO TO 156
 154 IF (K3-0) 156,155,156
 155 K3=1
 156 IF (K4-12) 161,160,161
 160 K4=8
 GO TO 185
 161 IF (K4-11) 163,162,163
 162 K4=7
 GO TO 185
 163 IF (K4-10) 165,164,165
 164 K4=-7
 GO TO 185
 165 IF (K4-9) 167,166,167
 166 K4=6
 GO TO 185
 167 IF (K4-8) 169,168,169
 168 K4=-6
 GO TO 185
 169 IF (K4-7) 171,170,171
 170 K4=5
 GO TO 185
 171 IF (K4-6) 173,172,173
 172 K4=-5
 GO TO 185


```

173 IF (K4-5) 175,174,175
174 K4=4
    GO TO 185
175 IF (K4-4) 177,176,177
176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=2
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    M=M+1
    GO TO 36
20 I=0
    J=J-1
    L=L-1
    GO TO 21
36 J=J+1
    I=I+1
21 K1=KORD(1,J)
    K2=KORD(2,J)
    K3=KORD(3,J)
    K4=KORD(4,J)
37 IF (L-12) 38,J9,39
38 GO TO 30

```

```

39 J=J+1
   I=0
   L=J
   IF (M-1) 7,6,6
6 PRINT 5
   M=0
7 IF (L-12) 40,42,42
40 GO TO 10
42 GO TO 43
43 STOP
   END
   END

```

0

0 0 0 0 0 2 0 0 0 0 0 0

0 0 0 2 2 6 2 0 0 3 0 3

3 3 3 1 0 9 7 4 0 3 7 7

12 12 3 12 12 12 9 3 3 12 12 12

* SY'PHONY - NO. 7 - CHORD PROGRESSION - RESULTS

8 -3 1 1

END

8 -3 1 1

END

8 5 -3 1

END

```

* 7498IT ROLLER A      2MIN IC OP
PROGRAM SYMB
DIMENSION LUM(27),KURT(132),KORD(132),MU(2,12),MUS(2,12),
1MUS1(2,12),MUSIC(2,12),NO(2,12),NOT(2,12),NOTE(2,12),KL(2,12),
2KLA(2,12),LO(2,12),LON(2,12),LONG(2,12),LONGA(2,12),MOE(26),
3JOE(14),KOE(10),LOE(2),MITH(4)
1 FORMAT (14)
2 FORMAT (12(13,12))
READ 2,((MU(1,N),I=1,2),N=1,12)
READ 2, ((MUS(1,N),I=1,2),N=1,12)
READ 2, ((MUS1(1,N),I=1,2),N=1,12)
READ 2,((MUSIC(1,N),I=1,2),N=1,12)
READ 2, ((NO(1,N),I=1,2),N=1,12)
READ 2, ((NOT(1,N),I=1,2),N=1,12)
READ 2, ((NOTE(1,N),I=1,2),N=1,12)
READ 2, ((KL(1,N),I=1,2),N=1,12)
READ 2, ((KLA(1,N),I=1,2),N=1,12)
READ 2, ((LO(1,N),I=1,2),N=1,12)
READ 2, ((LON(1,N),I=1,2),N=1,12)
READ 2, ((LONG(1,N),I=1,2),N=1,12)
READ 2,((LONGA(1,N),I=1,2),N=1,12)
11 FORMAT (13(13))
READ11, (MOE(1),I=1,13)
14 FORMAT (7(13))
READ14, (JOE(1),I=1,7)
16 FORMAT (5(13))
READ16, (KOE(1),I=1,5)
18 FORMAT (13)
READ18, (LOE(1),I=1,1)
DO 232 M=1,12
J=1
N=1

```

```

I=1
K=12
LUM(N)= MU(I,M)
N=N+1
LUM(N)= MUS(I,M)
N=N+1
LUM(N)= MUSI(I,M)
N=N+1
LUM(N)= MUSIC(I,M)
N=N+1
LUM(N)= NO(I,M)
N=N+1
LUM(N)= NOT(I,M)
N=N+1
LUM(N)= NOTE(I,M)
N=N+1
LUM(N)= KL(I,M)
N=N+1
LUM(N)= KLA(I,M)
N=N+1
LUM(N)= LO(I,M)
N=N+1
LUM(N)= LOM(I,M)
N=N+1
LUM(N)= LONG(I,M)
N=N+1
LUM(N)= LONGA(I,M)
8 DO 7 I=1,K
  IF (LUM(N)) 4,3,4
  4 IF (LUM(I)) 5,3,5
  5 KORU(I) = LUM(I) - LUM(N)
  IF (KORU(I)) 10,7,7

```

```

10 KORD(I)=-1*KORD(I)
   GO TO 7
3  KORD(I)=0
7  CONTINUE
   IF (N-2) 9,9,6
6  K=K-1
   N=N-1
   GO TO 8
9  N=12
   DO 183 I=1,78
   J=KORD(I)
   IF (J-N) 80,80,86
80  KORT(I)=KORD(I)
   GO TO 182
86  K=0
163 N=N+88
   K=K+88
   IF (J-N) 164,164,165
164 GO TO 170
165 N=N+12
   K=K+12
   IF (J-N) 166,166,167
166 GO TO 170
167 IF (L-6) 168,170,170
168 L=L+1
   GO TO 163
170 KORT(I)=KORD(I)-K
182 N=12
183 CONTINUE
   L=1
186 LEW=11
   N=13
   K=12
   IT =1

```



```

DO 66 I=1,78
  IF (KORT(I)-MOE(L)) 140,20,140
20  JROOT=KORT(I)
  DO 32 J=1,7
    IF (JROOT-JOE(J)) 32,31,32
31  MITH(1) =LUM(N)
    GO TO 68
32  CONTINUE
  DO 34 J=1,5
    IF (JROOT-KOE(J)) 34,33,34
33  MITH(1) = LUM(IT)
    GO TO 68
34  CONTINUE
  IF (JROOT-LOE(1)) 140,35,140
35  MITH(1)=0
    GO TO 68
140 IF (1-K) 21,22,66
21  IT=IT+1
    GO TO 66
22  K=K+LEW
    N=N-1
    IT=IT-LEW
    LEW=LEW-1
66  CONTINUE
67  IF (L-13) 69,65,66
69  L=L+1
    GO TO 186
68  PRINT 1, (MITH(1))
232 CONTINUE
      STOP
      END
      END

```


50851508515115115151515150751508515035150851511515115160361
 4084240842411426036260362507525085200000000000508525085251152
 408434084341143603636036350853508530000000000508535085351153
 408444084441144603646036450754508540000000000508545085451154
 40845408454114560365603655075550855000000000508555085551155
 308363113630836303363033630136211260000000000308363113630836
 308373113730837303373033730137211270000000000308373113730837
 4084840848401484014840148401480000000000408484084840848
 308393083930839301393013930139301390000000000308393083930839
 40840408404114041140410404074040840403404034040404114041140
 408404084041140411404104040740408404034040840411404114050350
 3113030830303040340403404034040340000000000403403083031130
 308303113030830303030303030130211100000000000308303113030830
 7 5 4 8 3 9 2 10 1 11 12 0 6
 0 3 4 7 10 11 12
 1 2 5 8 9
 6

SYPHONY - NO. 8 - HINDENBURG ROOTS - RESULTS

308
311
303
303
303
301
211
0
403
308
311
308

```

* 74981T ROLLER (5Y9)      2MIN IC  OP
PROGRAM SYM9
DIMENSION MUS(3,12),NOTE(3,12),KLANG(3,12),LONG(3,12),KAN(12),
1 KUN(12),KEN(12),KIN(12),MITH(12)
2 FORMAT (12(13,11,11))
READ 2, ((MUS(1,N),I=1,3),N=1,12)
READ 2, ((NOTE(1,N),I=1,3),N=1,12)
READ 2, ((KLANG(1,N),I=1,3),N=1,12)
READ 2, ((LONG(1,N),I=1,3),N=1,12)
3 FORMAT (12(14))
READ 3, (MITH(1),I=1,12)
13 FORMAT (1H0,5X,17HMELODIC INTERVALS)
14 FORMAT (1H0,5X,8H1 VIOLIN)
15 FORMAT (1H0,16(14))
16 FORMAT (1H0,5X,8H2 VIOLIN)
17 FORMAT (1H0,5X, 8HVIOLA)
18 FORMAT (1H0,5X,8HCELLO)
19 FORMAT (1H0,5X,15HINDEMITH ROOTS)
IAN=0
KAN=0
DO 3 N=1,12
IAN=IAN
JAN=MUS(1,N)
1F (IAN) 30,20,30
30 1F (JAN) 21,20,21
20 KAN(N)=0
KAN = JAN
GO TO 3
21 KAN(N)=IAN-JAN
KAN=JAN
6 CONTINUE
IAN=0
KAN=0

```

```

DO 9 N=1,12
  IAN=MAN
  JAN=NOTE(1,N)
  IF (IAN) 31,22,31
  31 IF (JAN) 23,24,23
  22 KUN(N)=0
  MAN=JAN
  GO TO 9
  23 KUN(N)=IAN-JAN
  MAN=JAN
  9 CONTINUE
  IAN=0
  MAN=0
  DO 10 N=1,12
    IAN=MAN
    JAN=KLANG(1,N)
    IF (IAN) 32,24,32
    32 IF (JAN) 25,24,25
    24 KUN(N)=0
    MAN=JAN
    GO TO 10
    25 KUN(N)=IAN-JAN
    MAN=JAN
    10 CONTINUE
    IAN=0
    MAN=0
    DO 11 N=1,12
      IAN=MAN
      JAN=LONG(1,N)
      IF (IAN) 33,26,33
      33 IF (JAN) 27,26,27
      26 KUN(N)=0
      MAN=JAN
      GO TO 11

```

```

27 KEN(N)=IAN-JAN
   MAN=JAN
11 CONTINUE
   IAN=0
   MAN=0
   DO 12 N=1,12
     IAN=MAN
     JAN=MITH(N)
     IF (IAN) 34,28,34
     IF (JAN) 29,20,29
34  KIN(N)=0
28  KIN(N)=0
   MAN=JAN
   GO TO 12
29  KIN(N)=IAN-JAN
   MAN = JAN
12 CONTINUE
   PRINT 13
   PRINT 14
   PRINT 15, (KAN(1),I=1,12)
   PRINT 16
   PRINT 15, (KUN(1),I=1,12)
   PRINT 17
   PRINT 15, (KUN(1),I=1,12)
   PRINT 18
   PRINT 15, (KEN(1),I=1,12)
   PRINT 19
   PRINT 15, (KIN(1),I=1,12)
   STOP
   END
   END

```

```

* 74981T ROLLER                               1MIN IC  OP
PROGRAM SYM10
DIMENSION KAN(60),MELO(30)
30 FORMAT (12(14))
READ 30, (KAN(I),I=1,60)
105 FORMAT (1H0,4H P1,9H MINOR 2,4H M2,9H MINOR 3,4H M3,4H P4,
14H A4,4H P5,9H MINOR 6,4H M6,9H MINOR 7,4H M7,4H P8)
100 FORMAT (1H0,1X,4(13,3X),3X,3(13,1X),13,3(3X,13),4X,13,2X,13)
107 FORMAT (1H0,17HMELODIC INTERVALS)
111 FORMAT (1H0,8H1 VIOLIN)
112 FORMAT (1H0,9H11 VIOLIN)
113 FORMAT (1H0,5HVIOLA)
114 FORMAT (1H0,5HCELLO)
115 FORMAT (1H0,15HHINDUETH ROOTS)
PRINT 111
M=1
N=12
NO = 1
NOT=12
80 IT = 1
91 DO 29 I= NO,NOT
L=1
91 IF (KAN(I)) 40,41,41
40 GO TO 92
41 J=KAN(I)
IF (J-N) 50,50,57
50 KAN(I)=KAN(J)
GO TO 76
57 K=0
47 N=K+88
K=K+88
IF (J-N) 68,68,59
58 GO TO 70

```

```

59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 KAN(I) = KAN(I) - K
76 N=12
   GO TO 29
82 J=KAN(I)
   IF (J+N) 84,83,83
83 KAN(I) = KAN(I)
   GO TO 17
84 K=0
10 N=N+88
   K=K+88
   IF (J+N) 12,11,11
11 GO TO 16
12 N=N+12
   K=K+12
   IF (J+N) 14,13,13
13 GO TO 16
14 IF (L-6) 15,16,16
15 L=L+1
   GO TO 10
16 KAN(I)=KAN(I)+K
17 N=12
29 CONTINUE
   PRINT 107
   PRINT 106, (KAN(I),I=NO,NOT)
   DO 6 I=1,26
     MCLO(I) = 0

```

```

6 CONTINUE
100 MOT = 0
88 DO 83 I= 1,13
   IF (KAN(IT)-MOT) 93,82,93
82 MELO(I) = MELO(I)+1
93 MOT = MOT+1
83 CONTINUE
MOT = 0
DO 85 I= 14,26
   IF (KAN(IT) + MOT) 95,84,95
84 MELO(I) = MELO(I) - 1
95 MOT = MOT+1
85 CONTINUE
   IF (IT- NOT) 20,87,87
20 IT = IT +1
   GO TO 100
87 PRINT 105
   PRINT 106, (MELO(I),I=1,13)
   PRINT 106, (MELO(I),I=14,26)
   IF (M-5) 18,18,110
18 NO=NO+12
   NOT=NOT+12
   N=12
   M=M+1
   IF (M-3) 116,117,118
116 PRINT 112
   GO TO 125
117 PRINT 113
   GO TO 125
118 IF (M-5) 119,120,110
119 PRINT 114
   GO TO 125

```



```

120 PRINT 115
125 GO TO 31
110 GO TO 121
121 STOP
    END
    END

```

| | | | | | | | | | | | |
|---|----|--------|---|---|----|--------|----|-----|----|----|-----|
| 0 | 0 | -3 | 0 | 1 | 3 | -1 | 5 | -5 | -3 | 0 | -92 |
| 0 | 0 | -3-192 | 0 | 0 | 96 | -1 | 0 | 0 | 0 | 0 | -3 |
| 0 | 0 | -3-192 | 0 | 0 | 96 | 0 | 0 | 0 | 0 | 0 | -3 |
| 0 | 0 | -3-192 | 0 | 0 | 96 | -1 | 0 | 0 | 0 | 0 | -3 |
| 0 | -3 | 3 | 5 | 0 | 2 | 90-192 | 95 | -95 | 92 | 92 | 3 |

MELODIC INTERVALS

| | P1 | MINOR 2 | M2 | MINOR 3 | M3 | P4 | A4 | P5 | MINOR 6 | M6 | MINOR 7 | M7 | P8 |
|--|----|---------|----|---------|----|----|----|----|---------|----|---------|----|----|
| | 0 | -3 | 0 | 1 | 3 | -1 | 5 | -5 | -3 | 0 | -4 | | |
| | 4 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | |
| | -4 | -1 | 0 | -2 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | | |

MELODIC INTERVALS

| | | | | | | | | | | | | | | |
|----|----|----|----|----|---|----|---|---|---|----|----|---------|----|----|
| P1 | 0 | -3 | -4 | 0 | B | -1 | 0 | 0 | 0 | -3 | M6 | MINOR 7 | M7 | P8 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| -7 | -1 | 0 | -2 | -2 | 0 | -2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

MELODIC INTERVALS

| | P1 | MINOR 2 | M2 | -3 | -4 | MINOR 3 | M3 | P4 | A4 | P5 | MINOR 6 | M6 | MINOR 7 | M7 | P8 |
|----|----|---------|----|----|----|---------|----|----|----|----|---------|----|---------|----|----|
| | 0 | 0 | -3 | | | | | 0 | 7 | 0 | 0 | 0 | 0 | 0 | -3 |
| B | 8 | 0 | 0 | | 0 | | 0 | | | 1 | 0 | 0 | 0 | 0 | |
| -B | -8 | 0 | 0 | | -3 | | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

MELODIC INTERVALS


```

*      GILBERT ROLLER - WEBERN - INTERVALS - NO.1
      DIMENSION MU(3,33),MUS(3,33),NO(3,33),NOT(3,33),KL(3,33),
      IKLA(3,33),LO(3,33),LON(3,33),LUM(10),KORD(10)
      195 FORMAT (1H0,7(I4))
      95 FORMAT (7(I4))
      2 FORMAT (11(I3,11,11))
      READ2,((MU(I,N),I=1,3),N=1,33)
      READ2,((MUS(I,N),I=1,3),N=1,33)
      READ 2,((NO(I,N),I=1,3),N=1,33)
      READ 2,((NOT(I,N),I=1,3),N=1,33)
      READ 2,((KL(I,N),I=1,3),N=1,33)
      READ 2,((KLA(I,N),I=1,3),N=1,33)
      READ 2,((LO(I,N),I=1,3),N=1,33)
      READ 2,((LON(I,N),I=1,3),N=1,33)
      M=0
      J=1
      10 M=M+1
      N=1
      I=1
      LUM(N)= MU(I,M)
      N=N+1
      LUM(N)= MUS(I,M)
      N=N+1
      LUM(N)= NO(I,M)
      N=N+1
      LUM(N)= NOT(I,M)
      N=N+1
      LUM(N)= KL(I,M)
      N=N+1
      LUM(N)= KLA(I,M)
      N=N+1
      LUM(N)= LO(I,M)
      N=N+1

```



```

LUM(N)=      LON(I,M)
L=N-7
3 IF (LUM(N)-0) 12,4,12
4 N=N-1
  IF (N-L) 12,12,11
11 GO TO 3
12 L=N-L
13 IF (LUM(I)) 5,7,5
5 IF (LUM(N)-LUM(I)) 7,7,6
6 KE=LUM(N)
  KA= LUM(I)
  LUM(N)=KA
  LUM(I)=KE
7 I=I+1
  L=L-1
  IF (L-0) 9,9,8
8 GO TO 13
9 I=1
91 IF (LUM(I)-0) 93,93,92
92 KORD(I)=LUM(I)-LUM(N)
  LON(I,M)=LUM(N)
  GO TO 94
93 KORD(I)=0
94 IF (I-7) 105,106,106
105 I=I+1
  GO TO 91
106 PRINT 195, (KORD(I),I=1,7)
  PUNCH 95, (KORD(I),I=1,7)
98 J=J+1
  IF (J-34) 102,103,103
102 GO TO 10
103 PUNCH 2, ((LON(I,N),I=1,3),N=1,33)
104 STOP
      END

```


* GILBERT ROLLER - WEBERN - INTERVALS - NO.1 - RESULTS

| | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|
| 196 | 103 | 0 | 0 | 0 | 0 | 0 |
| 290 | 197 | 199 | 191 | 0 | 0 | 0 |
| 100 | 100 | 295 | 295 | 0 | 0 | 196 |
| 192 | 192 | 297 | 297 | 0 | 0 | 198 |
| 108 | 0 | 0 | 0 | 0 | 0 | 6 |
| 201 | 3 | 96 | 96 | 3 | 299 | 99 |
| 198 | 3 | 193 | 193 | 3 | 299 | 99 |
| 198 | 93 | 197 | 197 | 93 | 299 | 99 |
| 7 | 7 | 107 | 107 | 4 | 303 | 103 |
| 0 | 0 | 101 | 101 | 97 | 96 | 0 |
| 0 | 0 | 101 | 101 | 7 | 0 | 0 |
| 0 | 0 | 202 | 202 | 10 | 4 | 0 |
| 107 | 103 | 201 | 201 | 99 | 0 | 0 |
| 200 | 198 | 0 | 0 | 0 | 0 | 4 |
| 398 | 398 | 293 | 293 | 0 | 97 | 0 |
| 301 | 301 | 196 | 196 | 0 | 0 | 4 |
| 400 | 400 | 295 | 295 | 99 | 95 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 |
| 402 | 402 | 297 | 297 | 101 | 2 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 |
| 99 | 5 | 95 | 95 | 2 | 0 | 1 |
| 197 | 99 | 104 | 104 | 102 | 0 | 0 |
| 102 | 10 | 9 | 9 | 0 | 4 | 4 |
| 193 | 98 | 190 | 94 | 0 | 92 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 200 |
| 297 | 198 | 293 | 192 | 0 | 101 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 98 |
| 106 | 101 | 98 | 6 | 204 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 196 | 0 |
| 205 | 200 | 197 | 104 | 301 | 96 | 0 |
| 207 | 107 | 199 | 106 | 303 | 0 | 98 |

| | | | | | | |
|---|-----|-----|----|----|---|---|
| 200 | 100 | 192 | 99 | 91 | 0 | 0 |
| 205 | 104 | 197 | 96 | 95 | 0 | 0 |
| 4054831138211282092840158308583085830858304383110840108 | | | | | | |
| 4014740307407473100740747308073084730607308474064730607 | | | | | | |
| 4014831108211482090841148403084114830608304483114830608 | | | | | | |

```

*   GILBERT ROLLER - WEBERN - REDUCTION - NO.3
    DIMENSION KORD(264),KORT1(33),KORT2(33),KORT3(33),KORT4(33),
           1KORT5(33),KORT6(33),KORT7(33),KORT8(33)
    1 FORMAT (7(14))
    READ 1, (KORD(I),I=1,224)
    2 FORMAT (1H0,7(14))
    M=1
    N=12
    IT = 1
    KT = 218
    5 DO 96 I=M,KT,7
      L=1
      J=KORD(I)
      IF (M-2) 11,13,15
      11 IF (J-N) 12,12,45
      12 KORT1(IT) = KORD(I)
      GO TO 96
      13 IF (J-N) 14,14,45
      14 KORT2(IT) = KORD(I)
      GO TO 96
      15 IF (M-4) 16,16,40
      16 IF (J-N) 17,17,45
      17 KORT3(IT) = KORD(I)
      GO TO 96
      18 IF (J-N) 19,19,45
      19 KORT4(IT) = KORD(I)
      GO TO 96
      20 IF (M-6) 21,21,25
      21 IF (J-N) 22,22,45
      22 KORT5(IT) = KORD(I)
      GO TO 96
      23 IF (J-N) 24,24,45

```

THE
JOURNAL
OF
THE
ROYAL
ANTHROPOLOGICAL
INSTITUTE
OF GREAT
BRITAIN
AND IRELAND
PART I
1901

```

24 KORT6(IT) = KORD(I)
   GO TO 96
25 IF (M-8) 26,98,98
26 IF (J-N) 27,27,45
27 KORT7(IT) = KORD(I)
   GO TO 96
45 K=0
47 N=N+BB
   K=K+BB
   IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 IF (M-2) 71,72,73
71 KORT1(IT) = KORD(I) -K
   GO TO 96
72 KORT2(IT) = KORD(I) -K
73 IF (M-4) 74,75,76
74 KORT3(IT) = KORD(I) -K
   GO TO 96
75 KORT4(IT) = KORD(I) -K
   GO TO 96
76 IF (M-6) 77,78,79
77 KORT5(IT) = KORD(I) -K
   GO TO 96

```



```

78 KORT6(IT) = KORU(1) -K
   GO TO 96
79 IF (M-8) 80,96,96
80 KORT7(IT) = KORU(1) -K
   GO TO 96
96 N=12
   IT = IT+1
98 CONTINUE
   IT = 1
   M=M+1
   KI=KI+1
   N=12
   IF (M-8) 99,100,100
99 GO TO 5
100 PRINT 2, (KORT1(1),I=1,32)
    PRINT 2, (KORT2(1),I=1,32)
    PRINT 2, (KORT3(1),I=1,32)
    PRINT 2, (KORT4(1),I=1,32)
    PRINT 2, (KORT5(1),I=1,32)
    PRINT 2, (KORT6(1),I=1,32)
    PRINT 2, (KORT7(1),I=1,32)
    PUNCH 1, (KORT1(1),I=1,32)
    PUNCH 1, (KORT2(1),I=1,32)
    PUNCH 1, (KORT3(1),I=1,32)
    PUNCH 1, (KORT4(1),I=1,32)
    PUNCH 1, (KORT5(1),I=1,32)
    PUNCH 1, (KORT6(1),I=1,32)
    PUNCH 1, (KORT7(1),I=1,32)
    STOP

```



```

* 74981T ROLLER WEB4      1MIN IC  OP
PROGRAM WEB4
  DIMENSION INTA(33),INTB(33),INTC(33),INTD(33),INTE(33),INTG(33),
1 INTK(33)
49 FORMAT (7(14))
1 FORMAT (7(14))
  READ 1, (INTA(I), I=1,28)
  READ 1, (INTB(I), I=1,28)
  READ 1, (INTC(I), I=1,28)
  READ 1, (INTD(I), I=1,28)
  READ 1, (INTE(I), I=1,28)
  READ 1, (INTG(I), I=1,28)
  READ 1, (INTK(I), I=1,28)
  DO 30 I=1,28
    M1 = INTA (I)
    M2 = INTB(I)
    M3 = INTC(I)
    M4 = INTD(I)
    M5 = INTE(I)
    M6 = INTG(I)
    M7 = INTK(I)
    DO 30 K=1,7
      KORTS = 0
      IF (KORTS - M1) 12,13,13
12 KORTS = M1
13 IF (KORTS -M2) 14,15,15
14 KORTS = M2
15 IF (KORTS -M3) 16,17,17
16 KORTS = M3
17 IF (KORTS - M4) 40,41,41
40 KORTS = M4
41 IF (KORTS - M5) 50,51,51
50 KORTS = M5

```


END

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|---|---|---|
| 196 | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 290 | 197 | 199 | 199 | 191 | 0 | 0 | 0 | 0 | 0 |
| 100 | 100 | 295 | 295 | 0 | 0 | 196 | 0 | 0 | 0 |
| 192 | 192 | 297 | 297 | 0 | 0 | 198 | 0 | 0 | 0 |
| 108 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 |
| 201 | 3 | 96 | 96 | 3 | 299 | 99 | 0 | 0 | 0 |
| 198 | 3 | 193 | 193 | 3 | 299 | 99 | 0 | 0 | 0 |
| 198 | 93 | 197 | 197 | 93 | 299 | 99 | 0 | 0 | 0 |
| 7 | 7 | 107 | 107 | 4 | 303 | 103 | 0 | 0 | 0 |
| 0 | 0 | 101 | 101 | 97 | 96 | 0 | 0 | 0 | 0 |
| 0 | 0 | 101 | 101 | 7 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 202 | 202 | 10 | 4 | 0 | 0 | 0 | 0 |
| 107 | 103 | 201 | 201 | 99 | 0 | 0 | 0 | 0 | 0 |
| 200 | 198 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| 398 | 398 | 293 | 293 | 0 | 97 | 0 | 0 | 0 | 0 |
| 301 | 301 | 196 | 196 | 0 | 0 | 4 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 99 | 95 | 0 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 | 0 | 0 | 0 |
| 402 | 402 | 297 | 297 | 101 | 2 | 0 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 | 0 | 0 | 0 |
| 99 | 5 | 95 | 95 | 2 | 0 | 1 | 0 | 0 | 0 |
| 197 | 99 | 104 | 104 | 102 | 0 | 0 | 0 | 0 | 0 |
| 102 | 10 | 9 | 9 | 0 | 4 | 4 | 0 | 0 | 0 |
| 193 | 98 | 190 | 94 | 0 | 92 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 297 | 198 | 293 | 192 | 0 | 101 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 98 | 0 | 0 | 0 |
| 106 | 101 | 98 | 6 | 204 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 196 | 0 | 95 | 0 | 0 | 0 |
| 205 | 200 | 197 | 104 | 301 | 96 | 0 | 0 | 0 | 0 |
| 207 | 107 | 199 | 106 | 303 | 0 | 98 | 0 | 0 | 0 |
| 200 | 100 | 192 | 99 | 91 | 0 | 0 | 0 | 0 | 0 |
| 196 | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 290 | 197 | 199 | 199 | 191 | 0 | 0 | 0 | 0 | 0 |
| 100 | 100 | 295 | 295 | 0 | 0 | 196 | 0 | 0 | 0 |
| 192 | 192 | 297 | 297 | 0 | 0 | 198 | 0 | 0 | 0 |
| 108 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 |
| 201 | 3 | 96 | 96 | 3 | 299 | 99 | 0 | 0 | 0 |
| 198 | 3 | 193 | 193 | 3 | 299 | 99 | 0 | 0 | 0 |
| 198 | 93 | 197 | 197 | 93 | 299 | 99 | 0 | 0 | 0 |
| 7 | 7 | 107 | 107 | 4 | 303 | 103 | 0 | 0 | 0 |
| 0 | 0 | 101 | 101 | 97 | 96 | 0 | 0 | 0 | 0 |
| 0 | 0 | 101 | 101 | 7 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 202 | 202 | 10 | 4 | 0 | 0 | 0 | 0 |
| 107 | 103 | 201 | 201 | 99 | 0 | 0 | 0 | 0 | 0 |
| 200 | 198 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| 398 | 398 | 293 | 293 | 0 | 97 | 0 | 0 | 0 | 0 |
| 301 | 301 | 196 | 196 | 0 | 0 | 4 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 99 | 95 | 0 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 | 0 | 0 | 0 |
| 402 | 402 | 297 | 297 | 101 | 2 | 0 | 0 | 0 | 0 |
| 400 | 400 | 295 | 295 | 103 | 0 | 99 | 0 | 0 | 0 |
| 99 | 5 | 95 | 95 | 2 | 0 | 1 | 0 | 0 | 0 |
| 197 | 99 | 104 | 104 | 102 | 0 | 0 | 0 | 0 | 0 |
| 102 | 10 | 9 | 9 | 0 | 4 | 4 | 0 | 0 | 0 |
| 193 | 98 | 190 | 94 | 0 | 92 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 0 |
| 297 | 198 | 293 | 192 | 0 | 101 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 98 | 0 | 0 | 0 |
| 106 | 101 | 98 | 6 | 204 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 196 | 0 | 95 | 0 | 0 | 0 |
| 205 | 200 | 197 | 104 | 301 | 96 | 0 | 0 | 0 | 0 |
| 207 | 107 | 199 | 106 | 303 | 0 | 98 | 0 | 0 | 0 |
| 200 | 100 | 192 | 99 | 91 | 0 | 0 | 0 | 0 | 0 |

```

* 74981T ROLLER WEB4      IMIN IC  OP
PROGRAM WEB4
  DIMENSION INTA(33),INTB(33),INTC(33),INTD(33),INTE(33),INTG(33),
    1 INTK(33)
  49 FORMAT (7(14))
  1 FORMAT (7(14))
  READ 1, (INTA(I), I=1,28)
  READ 1, (INTB(I), I=1,28)
  READ 1, (INTC(I), I=1,28)
  READ 1, (INTD(I), I=1,28)
  READ 1, (INTE(I), I=1,28)
  READ 1, (INTG(I), I=1,28)
  READ 1, (INTK(I), I=1,28)
  DO 30 I=1,28
    M1 = INTA (I)
    M2 = INTB(I)
    M3 = INTC(I)
    M4 = INTD(I)
    M5 = INTE(I)
    M6 = INTG(I)
    M7 = INTK(I)
    DO 30 K=1,7
      KURTS = 0
      IF (KURTS - M1) 12,13,13
12 KURTS = M1
13 IF (KURTS -M2) 14,15,15
14 KURTS = M2
15 IF (KURTS -M3) 16,17,17
16 KURTS = M3
17 IF (KURTS - M4) 40,41,41
40 KURTS = M4
41 IF (KURTS - M5) 50,51,51
50 KURTS = M5

```



```

51 IF (KORTS - M6) 52,53,53
52 KORTS = M6
53 IF (KORTS - M7) 54,55,55
54 KORTS = M7
55 IF (KORTS - M1) 19,18,19
18 M1=0
    GO TO 23
19 IF (KORTS - M2) 21,20,21
20 M2=0
    GO TO 23
21 IF (KORTS-M3) 42,22,42
22 M3=0
    GO TO 23
23 IF (KORTS - M4) 56,43,56
43 M4=0
    GO TO 23
25 IF (KORTS - M5) 58,57,58
57 M5=0
    GO TO 23
58 IF (KORTS - M6) 60,59,60
59 M6=0
    GO TO 23
60 IF (KORTS - M7) 23,61,23
61 M7=0
23 IF (K-1) 25,24,25
24 K0=KORTS
25 IF (K-2) 27,26,27
26 K0R = KORTS
27 IF (K-3) 44,28,44
28 KORT=KORTS
44 IF (K-4) 63,62,63
62 KORTA=KORTS
63 IF (K-5) 65,64,65
64 KORTB=KORTS

```



```

65 IF (K-6) 67,66,67
66 KURTC=KURTS
67 IF (K-7) 30,66,30
68 KURTD=KURTS
PRINT 49, K0,N0R,KURT,KURTA,KURTB,KURTC,KURTD
30 CONTINUE
STOP
END
END

```

```

8 2 12 4 8 1 10
10 7 0 0 0 7 12
10 1 12 12 2 12 11
9 2 5 9 0 6
3 9 12 4 0 3 3
5 7 0 0 0 3 10
10 1 12 12 2 12 5
11 10 10 0 10 0 1
0 11 7 9 0 8 5
9 7 1 1 2 1 0
5 8 7 7 9 7 7
4 9 2 0 5 0 10
0 11 7 9 0 8 5
9 7 1 1 2 1 0
5 8 7 7 9 7 7
4 9 6 0 4 0 6
0 3 0 0 0 0 3

```

| | | | | | | | | | | |
|----|----|---|----|----|---|---|----|---|----|----|
| 0 | 2 | 4 | 11 | 0 | 0 | 2 | 11 | 4 | 1 | 0 |
| 11 | 3 | 0 | 11 | 0 | 0 | 0 | 11 | 0 | 11 | 10 |
| 10 | 1 | 0 | 0 | 4 | 2 | 1 | 0 | 0 | 0 | 0 |
| 7 | 3 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 11 | 12 |
| 5 | 11 | 0 | 0 | 8 | 7 | 4 | 8 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 3 | 0 | 4 | 0 | 3 | 4 | 4 |
| 5 | 0 | 2 | 0 | 11 | 9 | 0 | 0 | 0 | 11 | 0 |

```

* 74981T ROLLER WOM          2MIN IC OP
PROGRAM WOM
  DIMENSION KORD(98),KEEP(48)
  1 FORMAT (7(I4))
  15 FORMAT (4(I4))
  READ 1, (KORD(I),I=1,98)
  J=0
  I=1
  N=2
  K=7
  4 IF (KORD(I)-KORD(N)) 6,2,6
  2 I=I+2
  GO TO 16
  14 J=J+1
  KEEP(J)=KORD(I)
  GO TO 5
  16 IF (KORD(N)-KORD(I)) 12,3,12
  3 N=N+2
  IF (N-K) 4,14,14
  6 J=J+1
  KEEP(J)=KORD(I)
  IF (N-K) 2,17,17
  17 J=J+1
  KEEP(J)=KORD(N)
  GO TO 5
  12 J=J+1
  KEEP(J)=KORD(N)
  GO TO 3
  5 IF (J-4) 7,9,9
  7 J=J+1
  DO 8 I=J,4
  KEEP(I)=0
  8 CONTINUE

```



```

*      GILBERT ROLLER - IIIA - REARRANGE      - WEIBERN
      DIMENSION KORD(7,33),INTERV(231)
      1 FORMAT (11(14))
      2 FORMAT (7(14))
      READ 2, ((KORD(I,N),I=1,7),N=1,33)
      J=1
      I=1
      4 DO 5 N=1,33
        INTERV(J)=KORD(I,N)
        J=J+1
      5 CONTINUE
      IF (I-12)6,7,7
      6 I=I+1
      GO TO 4
      7 PUNCH 1, (INTERV(I),I=1,231)
      STOP
      END

```

WEERN - NO. 5 - PRINT OUT
USE QUINTET PROGRAM V

WEBERN - NO. Va - REDUCTION OF LONG
USE QUARTET PROGRAM Va

* WEBER - NO. 5 - PRINT OUT - RESULTS

CODE

. IS MAJOR OR PERFECT
- IS MINOR
/ IS AUGMENTED

| MEASURE NO. | RHYTHMIC POSITION | HARMONIC INTERVALS | OCTAVE | INSTRUMENT |
|-------------|-------------------|--------------------|--------|------------|
| 1 | 1 | 6- | 0 | 1 |
| | | 3- | 0 | 2 |
| | | 0 | 0 | 3 |
| | | 0 | 0 | 4 |
| | | B- | 4 | 7 |
| 1 | 2 | 7. | 0 | 1 |
| | | 6. | 0 | 2 |
| | | 3- | 0 | 3 |
| | | 2. | 0 | 4 |
| | | 0 | 0 | 7 |
| 1 | 3 | 8. | 0 | 1 |
| | | 6- | 0 | 2 |
| | | 5. | 0 | 3 |
| | | 0 | 0 | 4 |
| | | B- | 4 | 7 |
| 2 | 1 | 7- | 0 | 1 |
| | | 6. | 0 | 2 |
| | | 3. | 0 | 3 |
| | | 0 | 0 | 4 |
| | | 0 | 0 | 7 |

| | | | | | |
|---|---|----------------------------|--------|-------------|-----------|
| 2 | 2 | 6-
44 | 0 0 B- | 0 0 0 0 4 | 1 2 3 4 7 |
| | | | | | |
| 2 | 3 | 7-
6-
3-
2-
0 | | 0 0 0 0 0 0 | 1 2 3 4 7 |
| | | | | | |
| 3 | 1 | 7-
7-
4-
3-
B- | | 0 0 0 0 0 4 | 1 2 3 4 7 |
| | | | | | |
| 3 | 2 | 7-
7-
6-
4-
0 | | 0 0 0 0 0 0 | 1 2 3 4 7 |
| | | | | | |
| 3 | 3 | 5-
3-
3-
0
B- | | 0 0 0 0 0 4 | 1 2 3 4 7 |

| | | | | |
|---|---|------------|-----------|-----------|
| 4 | 1 | 6-6-2-0 B- | 0 0 0 0 4 | 1 2 3 4 7 |
| 4 | 2 | 5-2-0 0 0 | 0 0 0 0 0 | 1 2 3 4 7 |
| 4 | 3 | 7-3-2-0 B- | 0 0 0 0 4 | 1 2 3 4 7 |
| 5 | 1 | 7-5-3-2-0 | 0 0 0 0 0 | 1 2 3 4 7 |
| 5 | 2 | 8-7-3-0 B- | 0 0 0 0 4 | 1 2 3 4 7 |

2 4 3 1 2 3 4 7

7 0 0 0 0 0 0 0

0 0 0 0 0 0 0 B-

3

1

5

6

```

* 74981T ROLLER Q3      1 MINIC  OP
PROGRAM WEB6
DIMENSION KORD(4,14)
41 FORMAT (4(14))
1 FORMAT (4(14))
1 READ 1, ((KORD(1,N),I=1,4),N=1,14)
J=1
DO 40 N=1,14
K1=KORD(1,N)
K2=KORD(2,N)
K3=KORD(3,N)
K4=KORD(4,N)
J=J+1
DO 40 M=J,14
L1=KORD(1,M)
L2=KORD(2,M)
L3=KORD(3,M)
L4=KORD(4,M)
IF (K1-L1) 31,32,31
31 GO TO 40
32 IF (K2-L2) 31,34,31
34 IF (K3-L3) 31,35,31
35 IF (K4-L4) 31,36,31
36 IF (K1-0) 43,38,43
37 IF (K2-0) 43,38,43
38 IF (K3-0) 43,39,43
39 IF (K4-0) 43,40,43
43 IF (K1-12) 61,60,61
60 K1=8
GO TO 100
61 IF (K1-11) 63,62,63
62 K1=7
GO TO 100

```


| | |
|----------------|-------------|
| 63 IF (K1-10) | 65,64,65 |
| 64 K1=-7 | |
| GO TO 100 | |
| 65 IF (K1-9) | 67,66,67 |
| 66 K1=6 | |
| GO TO 100 | |
| 67 IF (K1-8) | 69,68,69 |
| 68 K1=-6 | |
| GO TO 100 | |
| 69 IF (K1-7) | 71,70,71 |
| 70 K1=5 | |
| GO TO 100 | |
| 71 IF (K1-6) | 73,72,73 |
| 72 K1=-5 | |
| GO TO 100 | |
| 73 IF (K1-5) | 75,74,75 |
| 74 K1=4 | |
| GO TO 100 | |
| 75 IF (K1-4) | 77,76,77 |
| 76 K1=3 | |
| GO TO 100 | |
| 77 IF (K1-3) | 79,78,79 |
| 78 K1=-3 | |
| GO TO 100 | |
| 79 IF (K1-2) | 81,80,81 |
| 80 K1=2 | |
| GO TO 100 | |
| 81 IF (K1-1) | 83,82,83 |
| 82 K1=-2 | |
| GO TO 100 | |
| 83 IF (K1-0) | 100,84,100 |
| 84 K1=1 | |
| 100 IF (K2-12) | 102,101,102 |
| 101 K2=8 | |
| GO TO 130 | |

| | |
|----------------|-------------|
| 102 IF (K2-11) | 104,103,104 |
| 103 K2=7 | |
| GO TO 130 | |
| 104 IF (K2-10) | 106,105,106 |
| 105 K2=-7 | |
| GO TO 130 | |
| 106 IF (K2-9) | 108,107,108 |
| 107 K2=6 | |
| GO TO 130 | |
| 108 IF (K2-8) | 110,109,110 |
| 109 K2=-6 | |
| GO TO 130 | |
| 110 IF (K2-7) | 112,111,112 |
| 111 K2=5 | |
| GO TO 130 | |
| 112 IF (K2-6) | 114,113,114 |
| 113 K2=-5 | |
| GO TO 130 | |
| 114 IF (K2-5) | 116,115,116 |
| 115 K2=4 | |
| GO TO 130 | |
| 116 IF (K2-4) | 118,117,118 |
| 117 K2=3 | |
| GO TO 130 | |
| 118 IF (K2-3) | 120,119,120 |
| 119 K2=-3 | |
| GO TO 130 | |
| 120 IF (K2-2) | 122,121,122 |
| 121 K2=2 | |
| GO TO 130 | |
| 122 IF (K2-1) | 124,123,124 |
| 123 K2=-2 | |
| GO TO 130 | |
| 124 IF (K2-0) | 130,125,130 |

| | |
|----------------|-------------|
| 125 K2=1 | |
| 130 IF (K3-12) | 132,131,132 |
| 131 K3=8 | |
| GO TO 156 | |
| 132 IF (K3-11) | 134,133,134 |
| 133 K3=7 | |
| GO TO 156 | |
| 134 IF (K3-10) | 136,135,136 |
| 135 K3=-7 | |
| GO TO 156 | |
| 136 IF (K3-9) | 138,137,138 |
| 137 K3=6 | |
| GO TO 156 | |
| 138 IF (K3-8) | 140,139,140 |
| 139 K3=-6 | |
| GO TO 156 | |
| 140 IF (K3-7) | 142,141,142 |
| 141 K3=5 | |
| GO TO 156 | |
| 142 IF (K3-6) | 144,143,144 |
| 143 K3=-5 | |
| GO TO 156 | |
| 144 IF (K3-5) | 146,145,146 |
| 145 K3=4 | |
| GO TO 156 | |
| 146 IF (K3-4) | 148,147,148 |
| 147 K3=3 | |
| GO TO 156 | |
| 148 IF (K3-3) | 150,149,150 |
| 149 K3=-3 | |
| GO TO 156 | |
| 150 IF (K3-2) | 152,151,152 |
| 151 K3=2 | |
| GO TO 156 | |
| 152 IF (K3-1) | 154,153,154 |

153 K3=-2
GO TO 156
154 IF (K3-0) 156,155,156
155 K3=1
156 IF (K4-12) 161,160,161
160 K4=8
GO TO 185
161 IF (K4-11) 163,162,163
162 K4=7
GO TO 185
163 IF (K4-10) 165,164,165
164 K4=-7
GO TO 185
165 IF (K4-9) 167,166,167
166 K4=6
GO TO 185
167 IF (K4-8) 169,168,169
168 K4=-6
GO TO 185
169 IF (K4-7) 171,170,171
170 K4=5
GO TO 185
171 IF (K4-6) 173,172,173
172 K4=-5
GO TO 185
173 IF (K4-5) 175,174,175
174 K4=4
GO TO 185
175 IF (K4-4) 177,176,177

```

176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    KORD(1,M)=0
    KORD(2,M)=0
    KORD(3,M)=0
    KORD(4,M)=0
40 CONTINUE
    STOP
    END
    END

```

| | | | | |
|----|----|---|---|---|
| 8 | 3 | 0 | 0 | 0 |
| 11 | 9 | 3 | 2 | |
| 12 | 8 | 7 | 0 | |
| 10 | 9 | 4 | 0 | |
| 8 | 6 | 0 | 0 | |
| 11 | 8 | 3 | 1 | |
| 11 | 10 | 5 | 3 | |
| 11 | 10 | 9 | 5 | |
| 7 | 4 | 3 | 0 | |
| 9 | 8 | 1 | 0 | |

| | | | |
|----|----|---|---|
| 7 | 1 | 0 | 0 |
| 10 | 4 | 2 | 0 |
| 11 | 7 | 3 | 1 |
| 12 | 10 | 4 | 0 |

| | | | |
|----|----|---|---|
| 7 | 1 | 0 | 0 |
| 10 | 4 | 2 | 0 |
| 11 | 7 | 3 | 1 |
| 12 | 10 | 4 | 0 |

* WEBER NO. 6 - - CHORD REPETITION - RESULTS

NO REPETITIONS

WEBER - NO. 7 - CHORD PROGRESSION

USE QUINTET PROGRAM VII

1943.64 - 19.7 - 0-10 PROGRESSIVE - 100.00

10 PROGRESSIVE


```

* 74981T ROLLER C      3MIN IC OP
PROGRAM WEBB
DIMENSION LUM(27),KORT(132),KORD(132),MU(2,11),MUS(2,11),
1NO(2,11),NOT(2,11),KL(2,11),
2KLA(2,11),LO(2,11),LON(2,11),MITH(4),MOE(26),JOE(14),KOE(10),
3LOE(2)
1 FORMAT (I4)
2 FORMAT (11(I3,I2))
READ 2, ((MU(I,N),I=1,2),N=1,11)
READ 2, ((MUS(I,N),I=1,2),N=1,11)
READ 2, ((NO(I,N),I=1,2),N=1,11)
READ 2, ((NOT(I,N),I=1,2),N=1,11)
READ 2, ((KL(I,N),I=1,2),N=1,11)
READ 2, ((KLA(I,N),I=1,2),N=1,11)
READ 2, ((LO(I,N),I=1,2),N=1,11)
READ 2, ((LON(I,N),I=1,2),N=1,11)
11 FORMAT (13(I3))
READ 11, (MOE(I),I=1,13)
14 FORMAT (7(I3))
READ 14, (JOE(I),I=1,7)
16 FORMAT (5(I3))
READ 16, (KOE(I),I=1,5)
18 FORMAT (13)
READ 18, (LOE(I),I=1,1)
DO 232 M=1,11
J=1
N=1
I=1
K=7
LUM(N)= MU(I,M)
N=N+1

```

```

LUM(N)=      MUS(I,M)
N=N+1
LUM(N)=      NO(I,M)
N=N+1
LUM(N)=      NOT(I,M)
N=N+1
LUM(N)=      KL(I,M)
N=N+1
LUM(N)=      KLA(I,M)
N=N+1
LUM(N)=      LO(I,M)
N=N+1
LUM(N)=      LON(I,M)
8 DO 7 I=1,K
  IF (LUM(N)) 4,3,4
  4 IF (LUM(I)) 5,3,5
  5 KORD(I) = LUM(I) - LUM(N)
  IF (KORD(I)) 10,7,7
  10 KORD(I)=-1*KORD(I)
      GO TO 7
  3 KORD(I)=0
  7 CONTINUE
  IF (N-2) 9,9,6
  6 K=K-1
  N=N-1
  GO TO 8
  9 N=12
  DO 183 I=1,28
    J=KORD(I)
    IF (J-N) 80,80,86
  80 KORT(I)=KORD(I)
      GO TO 182
  86 K=0
  163 N=N+88
      K=K+88

```

```

      IF (J-N) 164,164,165
164 GO TO 170
165 N=N+12
      K=K+12
      IF (J-N) 166,166,167
166 GO TO 170
167 IF (L-6) 168,170,170
168 L=L+1
      GO TO 163
170 KORT(I)=KORD(I)-K
182 N=12
183 CONTINUE
      L=1
186 LEW=6
      N=8
      K=7
      IT =1
      DO 66 I=1,28
      IF (KORT(I)-MOE(L)) 140,20,140
20 JROOT=KORT(I)
      DO 32 J=1,7
      IF (JROOT-JOE(J)) 32,31,32
31 MITH(1) =LUM(N)
      GO TO 68
32 CONTINUE
      DO 34 J=1,5
      IF (JROOT-KOE(J)) 34,33,34
33 MITH(1) = LUM(IT)
      GO TO 68
34 CONTINUE
      IF (JROOT-LOE(1)) 140,35,140
35 MITH(1)=0
      GO TO 68
140 IF (I-K) 21,22,66
21 IT=IT+1
      GO TO 66

```


RESULTS

-

HINDLEWITH ROOTS

-

NO. 8

-

WEBERN

+

601
 601
 211
 401
 407
 311
 501
 506
 304
 0
 0

```

* 7498IT ROLLER (WEB9)      2MIN IC  OP
PROGRAM WEB9
DIMENSION MUS(3,33),NOTE(3,33),KLANG(3,33),LONG(3,33),KAN(33),
1KUN(33),KEN(33),KIN(33),MITH(33)
2 FORMAT (11(13,11,11))
READ 2, ((MUS(1,N),I=1,3),N=1,33)
READ 2, ((NOTE(1,N),I=1,3),N=1,33)
READ 2, ((KLANG(1,N),I=1,3),N=1,33)
READ 2, ((LONG(1,N),I=1,3),N=1,33)
3 FORMAT (11(14))
READ 3, (MITH(1),I=1,33)
13 FORMAT (1H0,5X,17HMELODIC INTERVALS)
14 FORMAT (1H0,5X,8H1 VIOLIN)
15 FORMAT (1H0,16(14))
16 FORMAT (1H0,5X,8H2 VIOLIN)
17 FORMAT (1H0,5X, 5HVIOLA)
18 FORMAT (1H0,5X,5HCELLO)
19 FORMAT (1H0,5X,15HINDEWITH ROOTS)
IAN=0
MAN=0
DO 8 N=1,33
IAN=MAN
JAN=MUS(1,N)
IF (IAN) 30,20,30
30 IF (JAN) 21,20,21
20 KAN(N)=0
MAN = JAN
GO TO 8
21 KAN(N)=IAN-JAN
MAN=JAN
8 CONTINUE
IAN=0
MAN=0

```



```

DO 9 N=1,33
  IAN=MAN
  JANSNOTE(1,N)
  IF (IAN) 31,24,31
31 IF (JAN) 23,24,23
22 KUN(N)=0
  MAN=JAN
  GO TO 9
23 KUN(N)=IAN-JAN
  MAN=JAN
  9 CONTINUE
  IAN=0
  MAN=0
DO 10 N=1,33
  IAN=MAN
  JANSKLANG(1,N)
  IF (IAN) 32,24,32
32 IF (JAN) 23,24,23
24 KUN(N)=0
  MAN=JAN
  GO TO 10
25 KUN(N)=IAN-JAN
  MAN=JAN
  10 CONTINUE
  IAN=0
  MAN=0
DO 11 N=1,33
  IAN=MAN
  JANSLONG(1,N)
  IF (IAN) 33,26,33
33 IF (JAN) 27,26,27
26 KUN(N)=0
  MAN=JAN

```

```

GO TO 11
27 KEN(N)=IAN-JAN
MAN=JAN
11 CONTINUE
IAN=0
MAN=0
DO 12 N=1,33
IAN=MAN
JAN=MITH(N)
IF (IAN) 34,28,34
34 IF (JAN) 29,28,29
28 KIN(N)=0
MAN=JAN
GO TO 12
29 KIN(N)=IAN-JAN
MAN = JAN
12 CONTINUE
PRINT 13
PRINT 14
PRINT 15, (KAN(1),I=1,33)
PRINT 16
PRINT 15, (KUN(1),I=1,33)
PRINT 17
PRINT 15, (KUN(1),I=1,33)
PRINT 18
PRINT 15, (KEN(1),I=1,33)
PRINT 19
PRINT 15, (KIN(1),I=1,33)
STOP
END
END

```



```

* 74981T KOLLER G          2MIN 1C UP
PROGRAM WEDH10
  DIMENSION KAN(105),MELU(30)
  30 FORMAT (10(14))
  READ 30, (KAN(I),I=1,100)
  105 FORMAT (1H0,4H P1,9H MINUR 2.4H M2.9H MINUR 3.4H M3.4H P4.
    14H A4.4H P5.9H MINUR 6.4H M6.9H MINUR 7.4H M7.4H P8)
  106 FORMAT (1H0,1X,4(13,3X),3X,3(13,1X),13,3(3X,13),4X,13,2X,13)
  107 FORMAT (1H0,17MELODIC INTERVALS)
  111 FORMAT (1H0,8H1 VIOLIN)
  112 FORMAT (1H0,9H11 VIOLIN)
  113 FORMAT (1H0,5HVIOLA)
  114 FORMAT (1H0,5HCELLO)
  115 FORMAT (1H0,15HMINDEMITH ROOTS)
  PRINT 111
  M=1
  N=12
  NU = 1
  NUT=32
  85 IT = 1
  31 DO 29 I= NU,NUT
    L=1
    51 IF (KAN(I)) 40,41,41
    40 GO TO 52
    41 J=KAN(I)
    IF (J-N) 50,50,57
    50 KAN(I)=KAN(I)
    GO TO 76
    57 K=0
    47 N=N+85
    K=K+35

```



```

IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
K=K+12
IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
GO TO 47
70 KAN(1) = KAN(1) - K
76 N=12
GO TO 29
52 J=KAN(1)
IF (J+N) 54,53,53
53 KAN(1) = KAN(1)
GO TO 17
54 K=0
10 N=N+88
K=K+88
IF (J+N) 12,11,11
11 GO TO 16
12 N=N+12
K=K+12
IF (J+N) 14,13,13
13 GO TO 16
14 IF (L-6) 15,16,16
15 L=L+1
GO TO 10
16 KAN(1)=KAN(1)+K
17 N=12
29 CONTINUE
PRINT 107
PRINT 106, (KAN(1),1=NO,NUT)
DO 6 1=1,26

```



```

      MELO(1) = 0
      6 CONTINUE
      100 MOT = 0
      88 DO B3 I= 1,13
         IF (KAN(IT)-MOT) 93,82,93
      82 MELO(1) = MELO(1)+1
      93 MOT = MOT+1
      83 CONTINUE
      MOT = 0
      DO B5 I= 14,26
         IF (KAN(IT) + MOT) 95,84,95
      84 MELO(1) = MELO(1) - 1
      95 MOT = MOT+1
      85 CONTINUE
      IF (IT- MOT) <0,87,87
      20 IT = IT +1
      86 GO TO 100
      87 PRINT 105
      PRINT 106, (MELO(I),I=1,13)
      PRINT 106, (MELO(I),I=14,26)
      IF (M-5) 18,18,110
      18 NO=NO+32
      NOT=NOT+32
      N=12
      M=M+1
      IF (M-3) 116,117,118
      116 PRINT 112
      GO TO 125
      117 PRINT 113
      GO TO 125
      118 IF (M-5) 119,120,110
      119 PRINT 114
      GO TO 125

```


22 0 1 1 1 0 1 0 0 0 0 0 0
 -22 -3 -1 -1 -1 0 0 0 0 0 -1 0 0

CELLO

MELODIC INTERVALS

0 0 4 0 0 0 0 -9 -4 3 -1 -2 -1 -1
 0 0 0 0 0 2 3 0 5 0 0
 0 0 0 0 0 0
 P1 MINOR 2 M2 MINOR 3 M3 P4 A4 P5 MINOR 6 M6 MINOR 7 M7 P8
 22 0 1 1 1 1 0 0 0 0 0
 -22 -3 -1 0 -1 0 0 0 -1 0 0

HINDENITH ROOTS

MELODIC INTERVALS

0 12 -12 10 -10 12 0 -2 2 -12 10 -10 12
 -12 10 -10 12 0 -2 2 -12 10 -10 12 -12 10
 -10 12 0 -2 2 -12
 P1 MINOR 2 M2 MINOR 3 M3 P4 A4 P5 MINOR 6 M6 MINOR 7 M7 P8
 3 3 3 3 3 3 3 3 3 3 3 3
 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 -3

SCHUMANN

```

*   GILBERT MÜLLER - NO. 1 - INTERVALS
    DIMENSION LUM(2,8),KORD(10),KUM(2,120)
    195 FORMAT (1H0,7(14))
    95  FORMAT (7(14))
    20  FORMAT (7(13,12))
    2   FORMAT (8(13,12))
    READ 2,((LUM(1,M),I=1,2),N=1,8)
    J=1
    10 N=8
    I=1
    L=N-7
    3 IF (LUM(1,N)-0) 12,4,12
    4 N=N-1
    IF (N-L) 12,12,11
    11 GO TO 3
    12 L=N-L
    13 IF (LUM(1,1)) 5,7,5
    5 IF (LUM(1,N)-LUM(1,1)) 7,7,6
    6 KE=LUM(1,N)
    KA=LUM(1,1)
    LUM(1,N)=KA
    LUM(1,1)=KE
    7 I=I+1
    L=L-1
    IF (L-0) 9,9,8
    8 GO TO 13
    9 I=1
    91 IF (LUM(1,1)-0) 93,93,92
    92 KORD(1)=LUM(1,1)-LUM(1,N)
    KUM(1,J)=LUM(1,N)
    KUM(2,J)=LUM(2,N)
    GO TO 94

```

the first of these is the fact that the
the second is the fact that the
the third is the fact that the
the fourth is the fact that the
the fifth is the fact that the
the sixth is the fact that the
the seventh is the fact that the
the eighth is the fact that the
the ninth is the fact that the
the tenth is the fact that the
the eleventh is the fact that the
the twelfth is the fact that the
the thirteenth is the fact that the
the fourteenth is the fact that the
the fifteenth is the fact that the
the sixteenth is the fact that the
the seventeenth is the fact that the
the eighteenth is the fact that the
the nineteenth is the fact that the
the twentieth is the fact that the
the twenty-first is the fact that the
the twenty-second is the fact that the
the twenty-third is the fact that the
the twenty-fourth is the fact that the
the twenty-fifth is the fact that the
the twenty-sixth is the fact that the
the twenty-seventh is the fact that the
the twenty-eighth is the fact that the
the twenty-ninth is the fact that the
the thirtieth is the fact that the
the thirty-first is the fact that the
the thirty-second is the fact that the
the thirty-third is the fact that the
the thirty-fourth is the fact that the
the thirty-fifth is the fact that the
the thirty-sixth is the fact that the
the thirty-seventh is the fact that the
the thirty-eighth is the fact that the
the thirty-ninth is the fact that the
the fortieth is the fact that the
the forty-first is the fact that the
the forty-second is the fact that the
the forty-third is the fact that the
the forty-fourth is the fact that the
the forty-fifth is the fact that the
the forty-sixth is the fact that the
the forty-seventh is the fact that the
the forty-eighth is the fact that the
the forty-ninth is the fact that the
the fiftieth is the fact that the
the fifty-first is the fact that the
the fifty-second is the fact that the
the fifty-third is the fact that the
the fifty-fourth is the fact that the
the fifty-fifth is the fact that the
the fifty-sixth is the fact that the
the fifty-seventh is the fact that the
the fifty-eighth is the fact that the
the fifty-ninth is the fact that the
the sixtieth is the fact that the
the sixty-first is the fact that the
the sixty-second is the fact that the
the sixty-third is the fact that the
the sixty-fourth is the fact that the
the sixty-fifth is the fact that the
the sixty-sixth is the fact that the
the sixty-seventh is the fact that the
the sixty-eighth is the fact that the
the sixty-ninth is the fact that the
the seventieth is the fact that the
the seventy-first is the fact that the
the seventy-second is the fact that the
the seventy-third is the fact that the
the seventy-fourth is the fact that the
the seventy-fifth is the fact that the
the seventy-sixth is the fact that the
the seventy-seventh is the fact that the
the seventy-eighth is the fact that the
the seventy-ninth is the fact that the
the eightieth is the fact that the
the eighty-first is the fact that the
the eighty-second is the fact that the
the eighty-third is the fact that the
the eighty-fourth is the fact that the
the eighty-fifth is the fact that the
the eighty-sixth is the fact that the
the eighty-seventh is the fact that the
the eighty-eighth is the fact that the
the eighty-ninth is the fact that the
the ninetieth is the fact that the
the ninety-first is the fact that the
the ninety-second is the fact that the
the ninety-third is the fact that the
the ninety-fourth is the fact that the
the ninety-fifth is the fact that the
the ninety-sixth is the fact that the
the ninety-seventh is the fact that the
the ninety-eighth is the fact that the
the ninety-ninth is the fact that the
the hundredth is the fact that the

```

93 KORD(I)=0
94 IF (I-7) 105,106,106
105 I=I+1
106 GO TO 91
106 PRINT 195, (KORD(I),I=1,7)
107 PUNCH 95, (KORD(I),I=1,7)
98 J=J+1
99 IF (J-16) 102,103,103
102 READ 2,((LUM(I,M),I=1,2),M=1,8)
103 PUNCH 20, ((KUM(I,N),I=1,2),N=1,14)
104 STOP
      END

```

```

6106160562602632102411015000060000700008
6056160262510533023420225000060000700008
60261510525055330534202525000060000700008
5105150552502533103421025000060000700008
50551410424024330234000005000060000700008
50551410424024330234000005000060000700008
50751410424034330334000005000060000700008
50551410424024330234000005000060000700008
5025141042405433123421225000060000700008
5025141042405433103421025000060000700008
5025141042405433093420925000060000700008
5025141042405433053424525000060000700008
6106160562602632102411015000060000700008
6056160262510533023420225000060000700008
60261510525055330534202525000060000700008

```


SCHUMANN - RESULTS

INTERVALS

NO. 1

GILBERT ROLLER -

*

| | | | | | | | |
|-----|-----|-----|-----|---|---|---|---|
| 500 | 495 | 492 | 100 | 0 | 0 | 0 | 0 |
| 403 | 400 | 308 | 100 | 0 | 0 | 0 | 0 |
| 397 | 305 | 300 | 100 | 0 | 0 | 0 | 0 |
| 300 | 295 | 292 | 100 | 0 | 0 | 0 | 0 |
| 203 | 108 | 100 | 0 | 0 | 0 | 0 | 0 |
| 203 | 108 | 100 | 0 | 0 | 0 | 0 | 0 |
| 204 | 107 | 100 | 0 | 0 | 0 | 0 | 0 |
| 203 | 108 | 100 | 0 | 0 | 0 | 0 | 0 |
| 291 | 197 | 193 | 100 | 0 | 0 | 0 | 0 |
| 292 | 200 | 195 | 100 | 0 | 0 | 0 | 0 |
| 293 | 201 | 196 | 100 | 0 | 0 | 0 | 0 |
| 297 | 205 | 205 | 100 | 0 | 0 | 0 | 0 |
| 500 | 495 | 492 | 100 | 0 | 0 | 0 | 0 |
| 403 | 400 | 308 | 100 | 0 | 0 | 0 | 0 |
| 397 | 305 | 300 | 100 | 0 | 0 | 0 | 0 |

```

PROGRAM SCH3
  DIMENSION KORD(105),KORT1(15),KORT2(15),KORT3(15),KORT4(15),
1 KORT5(15),KORT6(15),KORT7(15),KORT8(15)
  1 FORMAT (7(I4))
  READ 1, (KORD(I),I=1,105)
  2 FORMAT (1H0,7(I4))
  M=1
  N=12
  IT = 1
  KT = 99
  5 DO 98 I=M,KT,7
    L=1
    J=KORD(I)
    IF (M-2) 11,13,15
    11 IF (J-N) 12,12,45
    12 KORT1(IT) = KORD(I)
    GO TO 96
    13 IF (J-N) 14,14,45
    14 KORT2(IT) = KORD(I)
    GO TO 96
    15 IF (M-4) 16,18,20
    16 IF (J-N) 17,17,45
    17 KORT3(IT) = KORD(I)
    GO TO 96
    18 IF (J-N) 19,19,45
    19 KORT4(IT) = KORD(I)
    GO TO 96
    20 IF (M-6) 21,23,25
    21 IF (J-N) 22,22,45
    22 KORT5(IT) = KORD(I)
    GO TO 96
    23 IF (J-N) 24,24,45

```

```

24 KORT6(IT) = KORD(I)
   GO TO 96
25 IF (M-8) 26,98,98
26 IF (J-N) 27,27,45
27 KORT7(IT) = KORD(I)
   GO TO 96
45 K=0
47 N=N+88
   K=K+88
   IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 IF (M-2) 71,72,73
71 KORT1(IT) = KORD(I) -K
   GO TO 96
72 KORT2(IT) = KORD(I) -K
73 IF (M-4) 74,75,76
74 KORT3(IT) = KORD(I) -K
   GO TO 96
75 KORT4(IT) = KORD(I) -K
   GO TO 96
76 IF (M-6) 77,78,79
77 KORT5(IT) = KORD(I) -K
   GO TO 96
78 KORT6(IT) = KORD(I) -K
   GO TO 96
79 IF (M-8) 80,96,96
80 KORT7(IT) = KORD(I) -K

```

```

GO TO 96
96 N=12
   IT = IT+1
98 CONTINUE
   IT = 1
   M=M+1
   KT=KT+1
   N=12
   IF (M-8) 99,100,100
99 GO TO 5
100 PRINT 2, (KORT1(I),I=1,15)
    PRINT 2, (KORT2(I),I=1,15)
    PRINT 2, (KORT3(I),I=1,15)
    PRINT 2, (KORT4(I),I=1,15)
    PRINT 2, (KORT5(I),I=1,15)
    PRINT 2, (KORT6(I),I=1,15)
    PRINT 2, (KORT7(I),I=1,15)
    STOP
    END
    END
500 495 492 100 0 0 0 0
403 400 308 100 0 0 0 0
397 305 300 100 0 0 0 0
300 295 292 100 0 0 0 0
203 108 100 0 0 0 0 0
203 108 100 0 0 0 0 0
204 107 100 0 0 0 0 0
203 108 100 0 0 0 0 0
291 197 193 100 0 0 0 0
292 200 195 100 0 0 0 0
293 201 196 100 0 0 0 0
297 205 205 100 0 0 0 0
500 495 492 100 0 0 0 0
403 400 308 100 0 0 0 0
397 305 300 100 0 0 0 0

```



```

* 74981T ROLLER K          1MIN 1C  OP
PROGRAM SCH4
  DIMENSION INTA(15),INTB(15),INTC(15),INTD(15),INTE(15),INTG(15),
  1INTK(15)
  1 FORMAT (7(I4))
  49 FORMAT (1H0,7(I4))
  READ 1, (INTA(I),I=1,14)
  READ 1, (INTB(I),I=1,14)
  READ 1, (INTC(I),I=1,14)
  READ 1, (INTD(I),I=1,14)
  READ 1, (INTE(I),I=1,14)
  READ 1, (INTG(I),I=1,14)
  READ 1, (INTK(I),I=1,14)
  DO 30 I=1,14
    M1 = INTA (I)
    M2 = INTB(I)
    M3 = INTC(I)
    M4 = INTD(I)
    M5 = INTE (I)
    M6 = INTG(I)
    M7 = INTK(I)
    DO 30 K=1,7
      KORTS = 0
      IF (KORTS - M1) 12,13,13
      12 KORTS = M1
      13 IF (KORTS -M2) 14,15,15
      14 KORTS = M2
      15 IF (KORTS -M3) 16,17,17
      16 KORTS = M3
      17 IF (KORTS - M4) 40,41,41
      40 KORTS = M4
      41 IF (KORTS - M5) 50,51,51

```

```

50 KORTS = M5
51 IF (KORTS - M6) 52,53,53
52 KORTS = M6
53 IF (KORTS - M7) 54,55,55
54 KORTS = M7
55 IF (KORTS - M1) 19,18,19
18 M1=0
   GO TO 23
19 IF (KORTS - M2) 21,20,21
20 M2=0
   GO TO 23
21 IF (KORTS-M3) 42,22,42
22 M3=0
   GO TO 23
42 IF (KORTS - M4) 56,43,56
43 M4=0
   GO TO 23
56 IF (KORTS - M5) 58,57,58
57 M5=0
   GO TO 23
58 IF (KORTS - M6) 60,59,60
59 M6=0
   GO TO 23
60 IF (KORTS - M7) 23,61,23
61 M7=0
23 IF (K-1) 25,24,25
24 KO=KORTS
25 IF (K-2) 27,26,27
26 KOR = KORTS
27 IF (K-3) 44,28,44
28 KORT=KORTS
44 IF (K-4) 63,62,63
62 KORTA=KORTS
63 IF (K-5) 65,64,65

```



```

64 KORTB=KORTS
65 IF (K-6) 67,66,67
66 KORTC=KORTS
67 IF (K-7) 30,68,30
68 KORTD=KORTS
PRINT49, KO,KOR,KURT,KORTA,KORTB,KORTC,KORTD
30 CONTINUE
STOP
END
END

```

```

12 3 3 9 12 3 3 4
3 3 4 5 5 12 3
7 12 5 7 8 8 7
8 9 12 1 1 7 12
4 8 12 4 12 12 12
12 5 7 8 8 4 8
12 12 12 12 0 0 0
0 12 12 12 12 12 12
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

```



```

* 74981T ROLLER E      14IN IC  OP
PROGRAM SCHOM
DIMENSION KORD(105),KEEP(60)
1 FORMAT (7(14))
15 FORMAT (4(14))
READ 1, (KORD(1),1=1,105)
J=0
I=1
N=2
K=7
4 IF (KORD(1)-KORD(N)) 6,2,6
2 1=1+2
IF (N-K) 16,14,14
14 J=J+1
KEEP(J)=KORD(N)
GO TO 5
16 IF (KORD(N)-KORD(1)) 12,3,12
3 N=N+2
GO TO 4
6 J=J+1
KEEP(J)=KORD(1)
IF (N-K) 2,17,17
17 J=J+1
KEEP(J)=KORD(N)
GO TO 5
12 J=J+1
KEEP(J)=KORD(N)
GO TO 3
5 IF (J-4) 7,9,9
7 J=J+1
DO 8 1=J,4
KEEP(1)=0
8 CONTINUE

```

```

9 PRINT 15, (KEEP(1),I=1,4)
IF (K-105) 10,20,20
10 I=K+1
N=K+2
K=K+7
J=0
GO TO 4
20 STOP
END
END

```

| | | | | | | | |
|----|----|---|---|---|---|---|---|
| 12 | 12 | 7 | 4 | 0 | 0 | 0 | 0 |
| 12 | 12 | 6 | 3 | 0 | 0 | 0 | 0 |
| 12 | 12 | 9 | 5 | 0 | 0 | 0 | 0 |
| 12 | 12 | 7 | 4 | 0 | 0 | 0 | 0 |
| 12 | 12 | 3 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 3 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 4 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 3 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 5 | 3 | 0 | 0 | 0 | 0 |
| 12 | 12 | 7 | 4 | 0 | 0 | 0 | 0 |
| 12 | 12 | 5 | 1 | 0 | 0 | 0 | 0 |
| 12 | 12 | 5 | 5 | 0 | 0 | 0 | 0 |
| 12 | 12 | 7 | 4 | 0 | 0 | 0 | 0 |
| 12 | 12 | 6 | 3 | 0 | 0 | 0 | 0 |
| 12 | 12 | 9 | 5 | 0 | 0 | 0 | 0 |


```

*      GILBERT MULLER - SCHUMANN - NO.3 - REARRANGE
      DIMENSION KORD(8,15),INTERV(120)
      1 FORMAT (7(14))
      READ 1, ((KORD(I,N),I=1,7),N=1,15)
      2 FORMAT (15(14))
      DO 5 N=1,15
      INTERV(N)=KORD(1,N)
      5 CONTINUE
      N=N+1
      DO 6 J=1,15
      INTERV(N)=KORD(2,J)
      N=N+1
      6 CONTINUE
      DO 7 J=1,15
      INTERV(N)=KORD(3,J)
      N=N+1
      7 CONTINUE
      DO 8 J=1,15
      INTERV(N)=KORD(4,J)
      N=N+1
      8 CONTINUE
      DO 9 J=1,15
      INTERV(N) = KORD(5,J)
      N=N+1
      9 CONTINUE
      DO 10 J=1,15
      INTERV(N)=KORD(6,J)
      N=N+1
      10 CONTINUE
      DO 11 J=1,15
      INTERV(N) = KORD(7,J)
      N=N+1

```



```

11 CONTINUE
DO 12 J=1,15
  INTERV(N)=KORD(B,J)
  N=N+1
12 CONTINUE
PUNCH 2, (INTERV(I), I=1,105)
STOP
END

```

```

500 495 492 100 0 0 0 0
403 400 308 100 0 0 0 0
397 305 300 100 0 0 0 0
300 295 292 100 0 0 0 0
203 108 100 0 0 0 0 0
203 108 100 0 0 0 0 0
204 107 100 0 0 0 0 0
203 108 100 0 0 0 0 0
291 197 193 100 0 0 0 0
292 200 195 100 0 0 0 0
293 201 196 100 0 0 0 0
257 165 160 60 0 0 0 0
500 495 492 100 0 0 0 0
403 400 308 100 0 0 0 0
397 305 300 100 0 0 0 0

```

| * GILBERT KULLER - SCHUMANN - NO.3 - REARRANGE - RESULTS |
|---|
| 500 403 397 300 203 203 204 203 291 292 293 257 500 403 397 |
| 495 400 305 295 108 108 107 108 197 200 201 165 495 400 305 |
| 492 308 300 292 100 100 100 100 193 195 196 160 492 308 300 |
| 100 100 100 100 0 0 0 0 100 100 100 60 100 100 100 |
| 0 0 0 0 0 |
| 0 0 0 0 0 |
| 0 0 0 0 0 |

SCHUMANN - NO. Va - REDUCTION OF LONG
USE QUARTET PROGRAM Va

```

* 74981T ROLLER Z      IMIN IC OP
PROGRAM SCH5
  DIMENSION INFO(2,16),MUS(3,16),NOTE(3,48),LONG(3,16),KORD(64)
  1 FORMAT (16(13,12))
  2 FORMAT (16(13,11,11))
  READ 1, ((INFO(I,N),I=1,2),N=1,16)
  READ 2, ((MUS(I,N),I=1,3),N=1,16)
  READ 2, ((NOTE(I,N),I=1,3),N=1,48)
  READ 2, ((LONG(I,N),I=1,3),N=1,16)
  3 FORMAT (16(14))
  READ 3, (KORD(I),I=1,64)
  20 FORMAT (1H0,17,115,18X,2H0 ,113,110/)
  270 FORMAT (1H0,17,115,18X,2H2-,113,110/)
  271 FORMAT (1H0,17,115,18X,2H2.,113,110/)
  272 FORMAT (1H0,17,115,18X,2H3-,113,110/)
  273 FORMAT (1H0,17,115,18X,2H3.,113,110/)
  274 FORMAT (1H0,17,115,18X,2H4.,113,110/)
  275 FORMAT (1H0,17,115,18X,2H4+,113,110/)
  276 FORMAT (1H0,17,115,18X,2H5.,113,110/)
  277 FORMAT (1H0,17,115,18X,2H6-,113,110/)
  278 FORMAT (1H0,17,115,18X,2H6.,113,110/)
  279 FORMAT (1H0,17,115,18X,2H7-,113,110/)
  280 FORMAT (1H0,17,115,18X,2H7.,113,110/)
  281 FORMAT (1H0,17,115,18X,2H8.,113,110/)
  21 FORMAT (41X,2H0 ,113,110/)
  285 FORMAT (41X,2H2-,113,110/)
  286 FORMAT (41X,2H2.,113,110/)
  287 FORMAT (41X,2H3-,113,110/)
  288 FORMAT (41X,2H3.,113,110/)
  289 FORMAT (41X,2H4.,113,110/)
  290 FORMAT (41X,2H4+,113,110/)
  291 FORMAT (41X,2H5.,113,110/)
  292 FORMAT (41X,2H6-,113,110/)

```



```

293 FORMAT (41X,2H6.,113,110/)
294 FORMAT (41X,2H7-,113,110/)
295 FORMAT (41X,2H7.,113,110/)
296 FORMAT (41X,2H8.,113,110/)
  22 FORMAT (41X,2H0 ,113,110//)
315 FORMAT (41X,2HC ,113,110//)
316 FORMAT (41X,2HC+ ,113,110//)
317 FORMAT (41X,2HD ,113,110//)
318 FORMAT (41X,2HE- ,113,110//)
319 FORMAT (41X,2HE ,113,110//)
320 FORMAT (41X,2HF ,113,110//)
321 FORMAT (41X,2HF+ ,113,110//)
322 FORMAT (41X,2HG ,113,110//)
323 FORMAT (41X,2HA- ,113,110//)
324 FORMAT (41X,2HA ,113,110//)
325 FORMAT (41X,2HB- ,113,110//)
326 FORMAT (41X,2HB ,113,110//)
350 FORMAT (1H0,4HCODE/4X,21H. IS MAJOR OR PERFECT/4X,10H- IS MINOR/
  14X,14H+ IS AUGMENTED)
  PRINT 350
42 FORMAT (1H0,13HMEASURE NO. ,19HRHYTHMIC POSITION ,
  120HHARMONIC INTERVALS ,8HOCTAVE ,12HINSTRUMENT )
41 PRINT 42
  J=64
  DO 302 I=1,16
  12 J=J-63
  N=1
  IF (KORD(J)-0) 4,5,4
  5 PRINT 20,INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
  GO TO 217
  4 IF (KORD(J)-2) 200,201,202
  200 PRINT 270, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)

```



```

GO TO 217
201 PRINT 271, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
202 IF (KORD(J)-4) 203,204,205
203 PRINT 272, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
204 PRINT 273, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
205 IF (KORD(J)-6) 206,207,208
206 PRINT 274, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
207 PRINT 275, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
208 IF (KORD(J)-8) 209,210,211
209 PRINT 276, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
210 PRINT 277, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
211 IF (KORD(J)-10) 212,213,214
212 PRINT 278, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
213 PRINT 279, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
214 IF (KORD(J)-12) 215,216,217
215 PRINT 280, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
216 PRINT 281, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
217 J=J+16
DO 253 L=1,3
9 IF (KORD(J)-0) 6,7,6
7 PRINT 21, NOTE(2,N),NOTE(3,N)
GO TO 235

```



```

6 IF (KORD(J)-2) 218,219,220
218 PRINT 285, NOTE(2,N),NOTE(3,N)
   GO TO 235
219 PRINT 286, NOTE(2,N),NOTE(3,N)
   GO TO 235
220 IF (KORD(J)-4) 221,222,223
221 PRINT 287, NOTE(2,N),NOTE(3,N)
   GO TO 235
222 PRINT 288, NOTE(2,N),NOTE(3,N)
   GO TO 235
223 IF (KORD(J)-6) 224,225,226
224 PRINT 289, NOTE(2,N),NOTE(3,N)
   GO TO 235
225 PRINT 290, NOTE(2,N),NOTE(3,N)
   GO TO 235
226 IF (KORD(J)-8) 227,228,229
227 PRINT 291, NOTE(2,N),NOTE(3,N)
   GO TO 235
228 PRINT 292, NOTE(2,N),NOTE(3,N)
   GO TO 235
229 IF (KORD(J)-10) 230,231,232
230 PRINT 293, NOTE(2,N),NOTE(3,N)
   GO TO 235
231 PRINT 294, NOTE(2,N),NOTE(3,N)
   GO TO 235
232 IF (KORD(J)-12) 233,234,235
233 PRINT 295, NOTE(2,N),NOTE(3,N)
   GO TO 235
234 PRINT 296, NOTE(2,N),NOTE(3,N)
235 J=J+16
      N=N+16
253 CONTINUE

```



```

N=1
  IF (LONG(1,N) -0) 10,11,10
    11 PRINT 22, LONG(2,N),LONG(3,N)
      GO TO 302
    10 IF (LONG(1,N)-2) 255,256,257
      255 PRINT 315, LONG(2,N),LONG(3,N)
        GO TO 302
      256 PRINT 316, LONG(2,N),LONG(3,N)
        GO TO 302
      257 IF (LONG(1,N) -4) 258,259,260
        258 PRINT 317, LONG(2,N),LONG(3,N)
          GO TO 302
        259 PRINT 318, LONG(2,N),LONG(3,N)
          GO TO 302
        260 IF (LONG(1,N) -6) 261,262,263
          261 PRINT 319, LONG(2,N),LONG(3,N)
            GO TO 302
          262 PRINT 320, LONG(2,N),LONG(3,N)
            GO TO 302
          263 IF (LONG(1,N) -8) 264,265,266
            264 PRINT 321, LONG(2,N),LONG(3,N)
              GO TO 302
            265 PRINT 322, LONG(2,N),LONG(3,N)
              GO TO 302
            266 IF (LONG(1,N) -10) 267,268,269
              267 PRINT 323, LONG(2,N),LONG(3,N)
                GO TO 302
              268 PRINT 324, LONG(2,N),LONG(3,N)
                GO TO 302
              269 IF (LONG(1,N)-12) 327,328,302
                327 PRINT 325, LONG(2,N),LONG(3,N)
                  GO TO 302

```


* SCHUMANN - NO. 5 - PRINT OUT - RESULTS

CODE

. IS MAJOR OR PERFECT
- IS MINOR
/ IS AUGMENTED

| MEASURE NO. | RHYTHMIC POSITION | HARMONIC INTERVALS | OCTAVE | INSTRUMENT |
|-------------|-------------------|---------------------------|-----------------------|-----------------------|
| 1 | 1 | 8.
5.
3.
0
A | 0
0
0
0
1 | 1
2
3
4
5 |
| 1 | 2 | 8.
6-
3-
0
C/ | 0
0
0
0
2 | 1
2
3
4
5 |
| 1 | 3 | 8.
6.
4.
0
E | 0
0
0
0
2 | 1
2
3
4
5 |
| 2 | 1 | 8.
5.
3.
0
A | 0
0
0
0
2 | 1
2
3
4
5 |

| | | | | | |
|---|---|-----------|--------|-------|-------|
| 2 | 2 | 8:6-3-07 | 000003 | 12347 | 12347 |
| 2 | 2 | 8:6-3-07 | 000003 | 12347 | 12347 |
| 3 | 3 | 8:5-3-08 | 000003 | 12347 | 12347 |
| 2 | 3 | 8:6-3-07 | 000003 | 12347 | 12347 |
| 3 | 3 | 8:6-4-3-8 | 000002 | 12347 | 12347 |

| | | | | |
|---|---|-----------|-----------|-----------|
| 4 | 1 | 8.5.3.0 A | 0 0 0 0 2 | 1 2 3 4 5 |
| 7 | 2 | 8.6-4.2-A | 0 0 0 0 2 | 1 2 3 4 5 |
| 4 | 3 | 8.6.7.0 B | 0 0 0 0 2 | 1 2 3 4 5 |
| 5 | 1 | 8.5.3.0 A | 0 0 0 0 1 | 1 2 3 4 5 |
| 5 | 2 | 8.6-3.0 C | 0 0 0 0 2 | 1 2 3 4 5 |

1 2 3 4 0 1 2 3 4 0

0 0 0 0 0 0 0 0 0 0

8. 6. 4. 0 0 0 0 0 0 0

3

1

5

6

```

* 74981T ROLLER GA          I MINIC  OP
PROGRAM  SCH6
DIMENSION KORD(4,15)
41  FORMAT (4(I4))
1  FORMAT (4(I4))
1  READ 1, ((KORD(I,N),I=1,4),N=1,15)
J=1
DO 40 N=1,15
K1=KORD(1,N)
K2=KORD(2,N)
K3=KORD(3,N)
K4=KORD(4,N)
J=J+1
DO 40 M=J,15
L1=KORD(1,M)
L2=KORD(2,M)
L3=KORD(3,M)
L4=KORD(4,M)
IF (K1-L1) 31,32,31
31 GO TO 40
32 IF (K2-L2) 31,34,31
34 IF (K3-L3) 31,35,31
35 IF (K4-L4) 31,36,31
36 IF (K1-0) 43,38,43
37 IF (K2-0) 43,38,43
38 IF (K3-0) 43,39,43
39 IF (K4-0) 43,40,43
43 IF (K1-12) 61,60,61
60 K1=8
GO TO 100
61 IF (K1-11) 63,62,63

```


| | |
|---------------|------------|
| 62 K1=7 | |
| GO TO 100 | |
| 63 IF (K1-10) | 65,64,65 |
| 64 K1=7 | |
| GO TO 100 | |
| 65 IF (K1-9) | 67,66,67 |
| 66 K1=6 | |
| GO TO 100 | |
| 67 IF (K1-8) | 69,68,69 |
| 68 K1=5 | |
| GO TO 100 | |
| 69 IF (K1-7) | 71,70,71 |
| 70 K1=5 | |
| GO TO 100 | |
| 71 IF (K1-6) | 73,72,73 |
| 72 K1=5 | |
| GO TO 100 | |
| 73 IF (K1-5) | 75,74,75 |
| 74 K1=4 | |
| GO TO 100 | |
| 75 IF (K1-4) | 77,76,77 |
| 76 K1=3 | |
| GO TO 100 | |
| 77 IF (K1-3) | 79,78,79 |
| 78 K1=3 | |
| GO TO 100 | |
| 79 IF (K1-2) | 81,80,81 |
| 80 K1=2 | |
| GO TO 100 | |
| 81 IF (K1-1) | 83,82,83 |
| 82 K1=2 | |
| GO TO 100 | |
| 83 IF (K1-0) | 100,84,100 |
| 84 K1=1 | |

| | |
|----------------|-------------|
| 100 IF (K2-12) | 102,101,102 |
| 101 K2=8 | |
| GO TO 130 | |
| 102 IF (K2-11) | 104,103,104 |
| 103 K2=7 | |
| GO TO 130 | |
| 104 IF (K2-10) | 106,105,106 |
| 105 K2=7 | |
| GO TO 130 | |
| 106 IF (K2-9) | 108,107,108 |
| 107 K2=6 | |
| GO TO 130 | |
| 108 IF (K2-8) | 110,109,110 |
| 109 K2=6 | |
| GO TO 130 | |
| 110 IF (K2-7) | 112,111,112 |
| 111 K2=5 | |
| GO TO 130 | |
| 112 IF (K2-6) | 114,113,114 |
| 113 K2=5 | |
| GO TO 130 | |
| 114 IF (K2-5) | 116,115,116 |
| 115 K2=4 | |
| GO TO 130 | |
| 116 IF (K2-4) | 118,117,118 |
| 117 K2=3 | |
| GO TO 130 | |
| 118 IF (K2-3) | 120,119,120 |
| 119 K2=3 | |
| GO TO 130 | |
| 120 IF (K2-2) | 122,121,122 |
| 121 K2=2 | |
| GO TO 130 | |
| 122 IF (K2-1) | 124,123,124 |

| | |
|----------------|-------------|
| 123 K2--2 | |
| GO TO 130 | |
| 124 IF (K2=0) | 130,125,130 |
| 125 K2=1 | |
| 130 IF (K3=12) | 132,131,132 |
| 131 K3=8 | |
| GO TO 156 | |
| 132 IF (K3=11) | 134,133,134 |
| 133 K3=7 | |
| GO TO 156 | |
| 134 IF (K3=10) | 136,135,136 |
| 135 K3=-7 | |
| GO TO 156 | |
| 136 IF (K3=9) | 138,137,138 |
| 137 K3=6 | |
| GO TO 156 | |
| 138 IF (K3=8) | 140,139,140 |
| 139 K3=-6 | |
| GO TO 156 | |
| 140 IF (K3=7) | 142,141,142 |
| 141 K3=5 | |
| GO TO 156 | |
| 142 IF (K3=6) | 144,143,144 |
| 143 K3=-5 | |
| GO TO 156 | |
| 144 IF (K3=5) | 146,145,146 |
| 145 K3=4 | |
| GO TO 156 | |
| 146 IF (K3=4) | 148,147,148 |
| 147 K3=3 | |
| GO TO 156 | |
| 148 IF (K3=3) | 150,149,150 |
| 149 K3=-3 | |

THE
JOURNAL
OF
THE
ROYAL
ANTHROPOLOGICAL
INSTITUTE
OF GREAT
BRITAIN
AND IRELAND
VOLUME
LXXV
PART I
1905
LONDON
PUBLISHED BY THE
INSTITUTE
11, BEDFORD SQUARE, W.C.1
1905

GO TO 156
 150 IF (K3-2) 152,151,152
 151 K3=2
 GO TO 156
 152 IF (K3-1) 154,153,154
 153 K3=-2
 GO TO 156
 154 IF (K3-0) 156,155,156
 155 K3=1
 156 IF (K4-12) 161,160,161
 160 K4=8
 GO TO 165
 161 IF (K4-11) 163,162,163
 162 K4=7
 GO TO 165
 163 IF (K4-10) 165,164,165
 164 K4=-7
 GO TO 165
 165 IF (K4-9) 167,166,167
 166 K4=6
 GO TO 165
 167 IF (K4-8) 169,168,169
 168 K4=-6
 GO TO 165
 169 IF (K4-7) 171,170,171
 170 K4=5
 GO TO 165
 171 IF (K4-6) 173,172,173
 172 K4=-5
 GO TO 165
 173 IF (K4-5) 175,174,175
 174 K4=4
 GO TO 165
 175 IF (K4-4) 177,176,177


```

176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    KORD(1,M)=0
    KORD(2,M)=0
    KORD(3,M)=0
    KORD(4,M)=0
40 CONTINUE
STOP
END
12 7 4 0
12 8 3 0
12 9 5 0
12 7 4 0
12 8 3 0
12 8 3 0
12 7 4 0
12 8 3 0
12 9 5 3
12 7 4 0
12 8 5 1
12 9 5 0
12 7 4 0
12 8 3 0
12 9 5 0

```

* SCHUMANN - NO. 6 - CHORD REPETITION - RESULTS

| | | | |
|---|----|----|---|
| 8 | 5 | 3 | 1 |
| 8 | -6 | -3 | 1 |
| 8 | 6 | 4 | 1 |
| 8 | -6 | -3 | 1 |
| 8 | 5 | 3 | 1 |

* SCHEFFA N - NO. 7 - CIRCLED PROGRESSION
SEE QUINCY PROGRAM 7.

SCIENTIFIC - NO. 7 - C. AD. PL. 2. 1917 - 1918

12 7 4 0
12 8 3 0
12 9 5 0

AD

12 7 4 0
12 8 3 0

AD

12 7 4 0
12 8 3 0

AD

```

*      GILBERT ROLLER - STRAVINSKY - INTERVALS - NO.1
      DIMENSION MUSA(2,16),MUSB(2,16),MUSC(2,16),MUSD(2,16),MUSE(2,16),
      1MUSG(2,16),MUSH(2,16),MUSI(2,16),KORA(2,16),KORB(2,16),KORC(2,16),
      3KORE(2,16),NOTA(2,16),NOTB(2,16),NOTC(2,16),NOTR(2,16),NOTP(2,16),
      4LONA(2,16),LONB(2,16),LONC(2,16),LOND(2,16),LONE(2,16),LUM(30),
      5KORD(30)
      20 FORMAT (8(13,13))
      195 FORMAT (1H0,12(14))
      95 FORMAT (8(14))
      2 FORMAT (8(13,13,4X))
      READ 2, ((MUSA(I,N),I=1,2),N=1,16)
      READ 2, ((MUSB(I,N),I=1,2),N=1,16)
      READ 2, ((MUSC(I,N),I=1,2),N=1,16)
      READ 2, ((MUSD(I,N),I=1,2),N=1,16)
      READ 2, ((MUSE(I,N),I=1,2),N=1,16)
      READ 2, ((MUSG(I,N),I=1,2),N=1,16)
      READ 2, ((MUSH(I,N),I=1,2),N=1,16)
      READ 2, ((MUSI(I,N),I=1,2),N=1,16)
      READ 2, ((KORA(I,N),I=1,2),N=1,16)
      READ 2, ((KORB(I,N),I=1,2),N=1,16)
      READ 2, ((KORC(I,N),I=1,2),N=1,16)
      READ 2, ((KORE(I,N),I=1,2),N=1,16)
      READ 2, ((NOTA(I,N),I=1,2),N=1,16)
      READ 2, ((NOTB(I,N),I=1,2),N=1,16)
      READ 2, ((NOTC(I,N),I=1,2),N=1,16)
      READ 2, ((NOTR(I,N),I=1,2),N=1,16)
      READ 2, ((NOTP(I,N),I=1,2),N=1,16)
      READ 2, ((LONA(I,N),I=1,2),N=1,16)
      READ 2, ((LONB(I,N),I=1,2),N=1,16)
      READ 2, ((LONC(I,N),I=1,2),N=1,16)
      READ 2, ((LOND(I,N),I=1,2),N=1,16)
      READ 2, ((LONE(I,N),I=1,2),N=1,16)
      M=0

```



```

J=1
10 M=M+1
N=1
I=1
LUM(N) = MUSA(I,M)
N=N+1
LUM(N) = MUSB(I,M)
N=N+1
LUM(N) = MUSC(I,M)
N=N+1
LUM(N) = MUSD(I,M)
N=N+1
LUM(N) = MUSE(I,M)
N=N+1
LUM(N) = MUSG(I,M)
N=N+1
LUM(N) = MUSH(I,M)
N=N+1
LUM(N) = MUSI(I,M)
N=N+1
LUM(N) = KORA(I,M)
N=N+1
LUM(N) = KORB(I,M)
N=N+1
LUM(N) = KORC(I,M)
N=N+1
LUM(N) = KORE(I,M)
N=N+1
LUM(N) = NOTA(I,M)
N=N+1
LUM(N) = NOTB(I,M)
N=N+1
LUM(N) = NOTC(I,M)
N=N+1

```

```

LUM(N) = NOTR(I,M)
N=N+1
LUM(N) = NOTP(I,M)
N=N+1
LUM(N) = LONA(I,M)
N=N+1
LUM(N) = LONB(I,M)
N=N+1
LUM(N) = LONC(I,M)
N=N+1
LUM(N) = LOND(I,M)
N=N+1
LUM(N) = LONE(I,M)
L=N-21
3 IF (LUM(N)-0) 12,4,12
4 N=N-1
  IF (N-L) 12,12,11
11 GO TO 3
12 L=N-L
13 IF (LUM(I)) 5,7,5
5 IF (LUM(N)-LUM(I)) 7,7,6
6 KE=LUM(N)
  KA=LUM(I)
  LUM(N)=KA
  LUM(I)=KE
7 I=I+1
  L=L-1
  IF (L-0) 9,9,8
8 GO TO 13
9 I=1
91 IF (LUM(I)-0) 93,93,92
92 KORD(I)=LUM(I)-LUM(N)
  LONE(I,M) = LUM(N)
  GO TO 94

```

```

93 KORD(I)=0
94 IF (I-21) 105,106,106
105 I=I+1
    GO TO 91
106 PRINT 195, (KORD(I),I=1,21)
    PUNCH 95, (KORD(I),I=1,21)
98 J=J+1
    IF (J-17) 102,103,103
102 GO TO 10
103 PUNCH 20,((LONE(I,N),I=1,2),N=1,16)
104 STOP
    END

```

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 601601 | 601601 | 508501 | 508501 | 601601 | 601601 | 601601 | 605601 | 605601 |
| 508501 | 605601 | 601601 | 508501 | 605601 | 605601 | 508501 | 508501 | 000001 |
| 501502 | 501502 | 411402 | 411402 | 505502 | 505502 | 505502 | 605602 | 605602 |
| 508502 | 601602 | 505502 | 508502 | 601602 | 601602 | 505502 | 505502 | 2 |
| 605603 | 605603 | 508503 | 508503 | 601603 | 601603 | 601603 | 605603 | 605603 |
| 508503 | 605603 | 508503 | 508503 | 605603 | 605603 | 508503 | 605603 | 3 |
| 601604 | 601604 | 411404 | 411404 | 505504 | 505504 | 505504 | 601604 | 604604 |
| 411404 | 601604 | 505504 | 411404 | 601604 | 601604 | 505504 | 4 | 4 |
| 601605 | 601605 | 508505 | 508505 | 601605 | 601605 | 601605 | 601605 | 601605 |
| 508505 | 601605 | 508505 | 508505 | 601605 | 601605 | 508505 | 508505 | 5 |
| 501506 | 501506 | 411406 | 411406 | 505506 | 505506 | 505506 | 601606 | 601606 |
| 411406 | 505506 | 411406 | 411406 | 505506 | 505506 | 505506 | 505506 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 7 | 304307 | 304307 | 306307 | 306307 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 8 | 8 | 8 | 8 | 304308 | 304308 | 206208 | 206208 | 8 |
| 501509 | 501509 | 408409 | 408409 | 501509 | 501509 | 501509 | 505509 | 505509 |
| 9 | 9 | 9 | 9 | 505509 | 505509 | 9 | 9 | 9 |
| 401410 | 401410 | 406410 | 406410 | 405410 | 405410 | 405410 | 401410 | 401410 |
| 408410 | 10 | 10 | 10 | 405410 | 405410 | 408410 | 408410 | 10 |
| 501511 | 501511 | 408411 | 408411 | 501511 | 501511 | 501511 | 505511 | 505511 |
| 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| 401412 | 401412 | 406412 | 406412 | 405412 | 405412 | 405412 | 401412 | 401412 |

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 408412 | 12 | 408412 | 405412 | 408412 | 408412 | 12 | 12 |
| 605613 | 605613 | 508513 | 601613 | 601613 | 508513 | 508513 | 605613 |
| 508513 | 505513 | 508513 | 605613 | 508513 | 508513 | 508513 | 13 |
| 601614 | 601614 | 411414 | 505514 | 505514 | 411414 | 601614 | 601614 |
| 411414 | 505514 | 508514 | 601614 | 505514 | 508514 | 505514 | 14 |
| 301315 | 301315 | 204215 | 204215 | 204215 | 204215 | 206215 | 206215 |
| 208215 | 208215 | 206215 | 204215 | 206215 | 206215 | 206215 | 15 |
| 601616 | 601616 | 508516 | 601616 | 601616 | 508516 | 505516 | 505516 |
| 505516 | 601616 | 501516 | 601616 | 508516 | 408416 | 508516 | 16 |
| 401417 | 401417 | 304317 | 305317 | 305317 | 304317 | 306317 | 306317 |
| 308317 | 308317 | 306317 | 304317 | 306317 | 306317 | 306317 | 17 |
| 605618 | 605618 | 18 | 18 | 18 | 18 | 605618 | 605618 |
| 18 | 605618 | 18 | 605618 | 508518 | 508518 | 18 | 18 |
| 605619 | 605619 | 19 | 19 | 19 | 19 | 605619 | 605619 |
| 19 | 605619 | 19 | 605619 | 508519 | 508519 | 19 | 19 |
| 501520 | 501520 | 20 | 20 | 20 | 20 | 505520 | 505520 |
| 20 | 505520 | 20 | 505520 | 408420 | 408420 | 20 | 20 |
| 401421 | 401421 | 404421 | 405421 | 405421 | 405421 | 406421 | 406421 |
| 408421 | 408421 | 406421 | 404421 | 406421 | 406421 | 21 | 21 |
| 401422 | 401422 | 404422 | 405422 | 405422 | 405422 | 406422 | 406422 |
| 408422 | 408422 | 406422 | 404422 | 406422 | 406422 | 22 | 22 |

* GILBERT ROLLER - STRAVINSKY - INTERVALS - NO.1 - RESULTS

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 300 | 200 | 304 | 300 | 300 | 200 | 0 | 0 |
| 200 | 100 | 200 | 100 | 304 | 300 | 100 | 300 |
| 100 | 304 | 304 | 200 | 100 | | | |
| 300 | 200 | 304 | 300 | 300 | 200 | 0 | 0 |
| 200 | 100 | 200 | 100 | 304 | 300 | 100 | 300 |
| 100 | 304 | 304 | 200 | 100 | | | |
| 304 | 207 | 304 | 207 | 304 | 207 | 0 | 0 |
| 204 | 202 | 204 | 202 | 304 | 207 | 200 | 304 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 304 | 207 | 304 | 207 | 304 | 207 | 0 | 0 |
| 204 | 202 | 204 | 202 | 304 | 207 | 200 | 304 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 396 | 300 | 396 | 300 | 396 | 300 | 0 | 0 |
| 296 | 200 | 296 | 200 | 396 | 300 | 200 | 396 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 396 | 300 | 396 | 300 | 396 | 300 | 0 | 0 |
| 296 | 200 | 296 | 200 | 396 | 300 | 200 | 396 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 399 | 399 | 399 | 395 | 395 | 395 | 0 | 0 |
| 299 | 200 | 299 | 195 | 399 | 395 | 195 | 299 |
| 100 | 399 | 399 | 299 | 200 | | | |
| 399 | 399 | 399 | 398 | 395 | 395 | 0 | 0 |
| 299 | 200 | 299 | 195 | 399 | 395 | 195 | 299 |
| 100 | 399 | 399 | 299 | 200 | | | |
| 300 | 300 | 300 | 203 | 300 | 203 | 0 | 0 |
| 0 | 200 | 0 | 200 | 300 | 203 | 200 | 297 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 397 | 393 | 397 | 393 | 393 | 297 | 0 | 0 |
| 297 | 0 | 297 | 0 | 297 | 297 | 200 | 393 |
| 100 | 397 | 397 | 297 | 200 | | | |
| 395 | 299 | 395 | 299 | 395 | 299 | 0 | 0 |

| | | | | | | | |
|--|-----|-----|-----|-----|-----|-----|-----|
| 295 | 0 | 295 | 0 | 395 | 295 | 200 | 295 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 302 | 302 | 302 | 205 | 302 | 205 | 0 | 0 |
| 0 | 202 | 0 | 202 | 302 | 302 | 200 | 202 |
| 100 | 0 | 0 | 0 | 200 | | | |
| 401 | 397 | 401 | 397 | 397 | 301 | 200 | 100 |
| 301 | 201 | 301 | 201 | 401 | 397 | 100 | 397 |
| 100 | 401 | 401 | 301 | 200 | | | |
| 302 | 299 | 302 | 299 | 302 | 299 | 200 | 100 |
| 0 | 202 | 0 | 202 | 302 | 299 | 0 | 302 |
| 100 | 302 | 302 | 202 | 200 | | | |
| 302 | 299 | 0 | 0 | 302 | 299 | 100 | 100 |
| 0 | 202 | 0 | 0 | 302 | 299 | 0 | 302 |
| 0 | 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 0 | 0 | | | |
| 3014223014222042220422205422205422206422206422 | | | | | | | |
| 208422208422206422206422206422206422206 | | | | | | | |
| 22 | | | | | | | |
| 0 | | | | | | | |
| 22 | | | | | | | |


```

* 74981T ROLLER                                3MIN IC OP
PROGRAM STR3
  DIMENSION KORD(252),KORT1(12),KORT2(12),KORT3(12),KORT4(12),
  1KORT5(12),KORT6(12),KORT7(12),KORT8(12),KORT9(12),KORT10(12),
  2KORT11(12),KORT12(12),KORT13(12),KORT14(12),KORT15(12),KORT16(12),
  3KORT17(12),KORT18(12),KORT19(12),KORT20(12),KORT21(12)
  2 FORMAT (1H0, 12(14))
  3 FORMAT (8(14))
  6 FORMAT (5(14))
  K=1
  JIM =1
  1 L=K+15
  READ 3, (KORD(I),I=K,L)
  L=L+1
  M=L+4
  READ 6, (KORD(I),I=L,M)
  K=M+1
  IF (JIM-12) 4,7,7
  4 JIM=JIM+1
  GO TO 1
  7 M=1
  N=12
  IT =1
  KT = 232
  5 DO 98 I=M,KT,21
  L=1
  J=KORD(I)
  IF (M-2) 11,13,15
  11 IF (J-N) 12,12,45
  12 KORT1(IT) = KORD(I)
  GO TO 96
  13 IF (J-N) 14,14,45
  14 KORT2(IT) = KORD(I)
  GO TO 96

```



```
15 IF (M-4) 16,18,20
16 IF (J-N) 17,17,45
17 KORT3(IT) = KORD(I)
   GO TO 96
18 IF (J-N) 19,19,45
19 KORT4(IT) = KORD(I)
   GO TO 96
20 IF (M-6) 21,23,25
21 IF (J-N) 22,22,45
22 KORT5(IT) = KORD(I)
   GO TO 96
23 IF (J-N) 24,24,45
24 KORT6(IT) = KORD(I)
   GO TO 96
25 IF (M-8) 26,28,30
26 IF (J-N) 27,27,45
27 KORT7(IT) = KORD(I)
   GO TO 96
28 IF (J-N) 29,29,45
29 KORT8(IT) = KORD(I)
   GO TO 96
30 IF (M-10) 31,33,35
31 IF (J-N) 32,32,45
32 KORT9(IT) = KORD(I)
   GO TO 96
33 IF (J-N) 34,34,45
34 KORT10(IT) = KORD(I)
   GO TO 96
35 IF (M-12) 36,38,110
36 IF (J-N) 37,37,45
37 KORT11(IT) = KORD(I)
   GO TO 96
38 IF (J-N) 39,39,45
39 KORT12(IT) = KORD(I)
```



```

GO TO 96
110 IF (M-14) 111,113,115
111 IF (J-N) 112,112.45
112 KORT13(IT)=KORD(I)
GO TO 96
113 IF (J-N) 117,117.45
114 KORT14(IT)=KORD(I)
GO TO 96
115 IF (M-16) 116,118,120
116 IF (J-N) 117,117.45
117 KORT15(IT)=KORD(I)
GO TO 96
118 IF (J-N) 119,119.45
119 KORT16(IT)=KORD(I)
GO TO 96
120 IF (M-18) 121,123,125
121 IF (J-N) 122,122.45
122 KORT17(IT)=KORD(I)
GO TO 96
123 IF (J-N) 124,124.45
124 KORT18(IT)=KORD(I)
GO TO 96
125 IF (M-20) 126,128,130
126 IF (J-N) 127,127.45
127 KORT19(IT)=KORD(I)
GO TO 96
128 IF (J-N) 129,129.45
129 KORT20(IT)=KORD(I)
GO TO 96
130 IF (M-22) 131,98,98
131 IF (J-N) 132,132.45
132 KORT21(IT)=KORD(I)

```

```

      GO TO 96
45 K=0
47 N=N+BB
      K=K+BB
      IF (J-N) 58,58,59
58 GO TO 70
59 N = N+12
      K=K+12
      IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
      GO TO 47
70 IF (M-2) 71,72,73
71 KORT1(IT) = KORU(1) -K
      GO TO 96
72 KORT2(IT) = KORU(1) -K
73 IF (M-4) 74,75,76
74 KORT3(IT) = KORU(1) -K
      GO TO 96
75 KORT4(IT) = KORU(1) -K
      GO TO 96
76 IF (M-6) 77,78,79
77 KORT5(IT) = KORU(1) -K
      GO TO 96
78 KORT6(IT) = KORU(1) -K
      GO TO 96
79 IF (M-8) 80,81,82
80 KORT7(IT) = KORU(1) -K
      GO TO 96
81 KORT8(IT) = KORU(1) -K
      GO TO 96
82 IF (M-10) 83,84,85

```

```

83 KURT9(IT) = KURD(1) -K
   GO TO 96
84 KURT10(IT)=KURD(1)-K
   GO TO 96
85 IF (M-12) 86,87,88
86 KURT11(IT)=KURD(1)-K
   GO TO 96
87 KURT12(IT)=KURD(1)-K
   GO TO 96
88 IF (M-14) 89,90,91
89 KURT13(IT)=KURD(1) -K
   GO TO 96
90 KURT14(IT)=KURD(1) -K
   GO TO 96
91 IF (M-16) 92,93,94
92 KURT15(IT)=KURD(1) -K
   GO TO 96
93 KURT16(IT)=KURD(1) -K
   GO TO 96
94 IF (M-18) 50,51,52
95 KURT17(IT)=KURD(1) -K
   GO TO 96
96 KURT18(IT)=KURD(1) -K
   GO TO 96
97 IF (M-20) 53,54,55
98 KURT19(IT)=KURD(1) -K
   GO TO 96
99 KURT20(IT)=KURD(1) -K
   GO TO 96
95 IF (M-22) 56,96,96
96 KURT21(IT) = KURD(1) -K
96 N=12
   IT = IT+1

```



```

98 CONTINUE
   IT = 1
   M=M+1
   KT=KT+1
   N=12
   IF (M-22) 99,100,100
99 GO TO 5
100 PRINT 2, (KORT1(IT),IT=1,12)
    PRINT 2, (KORT2(IT),IT=1,12)
    PRINT 2, (KORT3(IT),IT=1,12)
    PRINT 2, (KORT4(IT),IT=1,12)
    PRINT 2, (KORT5(IT),IT=1,12)
    PRINT 2, (KORT6(IT),IT=1,12)
    PRINT 2, (KORT7(IT),IT=1,12)
    PRINT 2, (KORT8(IT),IT=1,12)
    PRINT 2, (KORT9(IT),IT=1,12)
    PRINT 2, (KORT10(IT),IT=1,12)
    PRINT 2, (KORT11(IT),IT=1,12)
    PRINT 2, (KORT12(IT),IT=1,12)
    PRINT 2, (KORT13(IT),IT=1,12)
    PRINT 2, (KORT14(IT),IT=1,12)
    PRINT 2, (KORT15(IT),IT=1,12)
    PRINT 2, (KORT16(IT),IT=1,12)
    PRINT 2, (KORT17(IT),IT=1,12)
    PRINT 2, (KORT18(IT),IT=1,12)
    PRINT 2, (KORT19(IT),IT=1,12)
    PRINT 2, (KORT20(IT),IT=1,12)
    PRINT 2, (KORT21(IT),IT=1,12)
    STOP
    END
    END

```



```

* 74981T ROLLER L          3MIN 1C OP
PROGRAM STR4
  DIMENSION INTA(16),INTB(16),INTC(16),INTD(16),INTE(16),INTG(16),
  1INTH(16),INTI(16),INTJ(16),INTK(16),INTL(16),INTM(16),INTN(16),
  2INTO(16),INTP(16),INTQ(16),INTR(16),INTS(16),INTT(16),INTU(16),
  3INTV(16)
  1 FORMAT (12(I4))
  49 FORMAT (1H0,12(I4))
  READ 1, (INTA(I),I=1,12)
  READ 1, (INTB(I),I=1,12)
  READ 1, (INTC(I),I=1,12)
  READ 1, (INTD(I),I=1,12)
  READ 1, (INTE(I),I=1,12)
  READ 1, (INTG(I),I=1,12)
  READ 1, (INTH(I),I=1,12)
  READ 1, (INTI(I),I=1,12)
  READ 1, (INTJ(I),I=1,12)
  READ 1, (INTK(I),I=1,12)
  READ 1, (INTL(I),I=1,12)
  READ 1, (INTM(I),I=1,12)
  READ 1, (INTN(I),I=1,12)
  READ 1, (INTO(I),I=1,12)
  READ 1, (INTP(I),I=1,12)
  READ 1, (INTQ(I),I=1,12)
  READ 1, (INTR(I),I=1,12)
  READ 1, (INTS(I),I=1,12)
  READ 1, (INTT(I),I=1,12)
  READ 1, (INTU(I),I=1,12)
  READ 1, (INTV(I),I=1,12)
  DO 30 I=1,12
  M1=INTA (I)

```

```

M2=INTB (I)
M3=INTC (I)
M4=INTD (I)
M5=INTE (I)
M6=INTG (I)
M7=INTH (I)
M8=INTI (I)
M9=INTJ (I)
M10=INTK (I)
M11=INTL (I)
M12=INTM (I)
M13=INTN (I)
M14=INTO (I)
M15=INTP (I)
M16=INTQ (I)
M17=INTR (I)
M18=INTS (I)
M19=INTT (I)
M20=INTU (I)
M21=INTV (I)
DO 30 K=1,21
KORTS =0
IF (KORTS - M1) 12,13,13
12 KORTS = M1
13 IF (KORTS -M2) 14,15,15
14 KORTS = M2
15 IF (KORTS -M3) 16,17,17
16 KORTS = M3
17 IF (KORTS - M4) 40,41,41
40 KORTS = M4
41 IF (KORTS - M5) 50,51,51
50 KORTS = M5
51 IF (KORTS - M6) 52,53,53
52 KORTS = M6

```

53 IF (KORTS - M7) 54,55,55
 54 KORTS = M7
 55 IF (KORTS - M8) 70,71,71
 70 KORTS = M8
 71 IF (KORTS - M9) 72,73,73
 72 KORTS = M9
 73 IF (KORTS - M10) 74,75,75
 74 KORTS = M10
 75 IF (KORTS - M11) 76,77,77
 76 KORTS = M11
 77 IF (KORTS - M12) 78,79,79
 78 KORTS = M12
 79 IF (KORTS - M13) 80,81,81
 80 KORTS = M13
 81 IF (KORTS - M14) 82,83,83
 82 KORTS = M14
 83 IF (KORTS - M15) 84,85,85
 84 KORTS = M15
 85 IF (KORTS - M16) 86,87,87
 86 KORTS = M16
 87 IF (KORTS - M17) 88,89,89
 88 KORTS = M17
 89 IF (KORTS - M18) 90,91,91
 90 KORTS = M18
 91 IF (KORTS - M19) 92,93,93
 92 KORTS = M19
 93 IF (KORTS - M20) 94,95,95
 94 KORTS = M20
 95 IF (KORTS - M21) 96,97,97
 96 KORTS = M21
 97 IF (KORTS - M1) 19,18,19
 18 M1=0


```

GO TO 23
19 IF (KORTS - M2) 21,20,21
20 M2=0
GO TO 23
21 IF (KORTS-M3) 42,22,42
22 M3=0
GO TO 23
42 IF (KORTS - M4) 56,43,56
43 M4=0
GO TO 23
56 IF (KORTS - M5) 58,57,58
57 M5=0
GO TO 23
58 IF (KORTS - M6) 60,59,60
59 M6=0
GO TO 23
60 IF (KORTS - M7)127,61,127
61 M7=0
GO TO 23
127 IF (KORTS - M8) 100,101,100
101 M8=0
GO TO 23
100 IF (KORTS - M9) 103,102,103
102 M9=0
GO TO 23
103 IF (KORTS - M10) 105,104,105
104 M10=0
GO TO 23
105 IF (KORTS - M11) 107,106,107
106 M11=0
GO TO 23
107 IF (KORTS - M12) 109,108,109
108 M12=0
GO TO 23
109 IF (KORTS - M13) 111,110,111

```



```

110 M13=0
    GO TO 23
111 IF (KORTS - M14) 113,112,113
112 M14=0
    GO TO 23
113 IF (KORTS - M15) 115,114,115
114 M15=0
    GO TO 23
115 IF (KORTS - M16) 117,116,117
116 M16=0
    GO TO 23
117 IF (KORTS - M17) 119,118,119
118 M17=0
    GO TO 23
119 IF (KORTS - M18) 121,120,121
120 M18=0
    GO TO 23
121 IF (KORTS - M19) 123,122,123
122 M19=0
    GO TO 23
123 IF (KORTS - M20) 125,124,125
124 M20=0
    GO TO 23
125 IF (KORTS - M21) 23,126,23
126 M21=0
23 IF (K-1) 25,24,25
24 KO=KORTS
    GO TO 30
25 IF (K-2) 27,26,27
26 KOR = KORTS
    GO TO 30
27 IF (K-3) 44,28,44

```

28 KORT=KORTS
 GO TO 30
 44 IF (K-4) 63,62,63
 62 KORTA=KORTS
 GO TO 30
 63 IF (K-5) 65,64,65
 64 KORTB=KORTS
 GO TO 30
 65 IF (K-6) 67,66,67
 66 KORTC=KORTS
 GO TO 30
 67 IF (K-7) 131,130,131
 130 KORTD = KORTS
 GO TO 30
 131 IF (K-8) 133,132,133
 132 KORTE = KORTS
 GO TO 30
 133 IF (K-9) 135,134,135
 134 KORTG = KORTS
 GO TO 30
 135 IF (K-10) 137,136,137
 136 KORTH = KORTS
 GO TO 30
 137 IF (K-11) 139,138,139
 138 KORTI = KORTS
 GO TO 30
 139 IF (K-12) 141,140,141
 140 KORTJ = KORTS
 GO TO 30
 141 IF (K-13) 143,142,143
 142 KORTK = KORTS
 GO TO 30

```

143 IF (K-14) 145,144,145
144 KORTL = KORTS
145 GO TO 30
146 IF (K-15) 147,146,147
147 KORTM = KORTS
148 GO TO 30
149 IF (K-16) 149,148,149
150 KORTN = KORTS
151 GO TO 30
152 IF (K-17) 151,150,151
153 KORTO = KORTS
154 GO TO 30
155 IF (K-18) 153,152,153
156 KORTP = KORTS
157 GO TO 30
158 IF (K-19) 155,154,155
159 KORTQ = KORTS
160 GO TO 30
161 IF (K-20) 157,156,157
162 KORTR = KORTS
163 GO TO 30
164 IF (K-21) 30,158,30
165 KORTT = KORTS
166 PRINT 49, KO,KOR,KORT,KORTA,KORTB,KORTC,KORTD,KORTE,KORTG,KORTH,
      1,KORTI,KORTJ,KORTK,KORTL,KORTM,KORTN,KORTO,KORTP,KORTQ,KORTR,KORTT
30 CONTINUE
STOP
END
END

```

| | | | | | | | | | | | |
|----|----|---|---|----|----|----|----|----|---|----|---|
| 12 | 12 | 4 | 4 | 8 | 8 | 11 | 11 | 12 | 9 | 7 | 2 |
| 12 | 12 | 7 | 7 | 12 | 12 | 11 | 11 | 12 | 5 | 11 | 2 |
| 4 | 4 | 4 | 4 | 8 | 8 | 11 | 11 | 12 | 9 | 7 | 2 |
| 12 | 12 | 7 | 7 | 12 | 12 | 7 | 10 | 3 | 5 | 11 | 5 |
| 12 | 12 | 4 | 4 | 8 | 8 | 7 | 7 | 12 | 5 | 7 | 2 |
| 12 | 12 | 7 | 7 | 12 | 12 | 7 | 7 | 3 | 9 | 11 | 5 |


```

* 749817 ROLLER STUM          IMIN  IC  OP
PROGRAM STUM
  DIMENSION KORD(252),KEEP(48)
  1 FORMAT (12(14))
  15 FORMAT (4(14))
  13 FORMAT (9(14))
  K=1
  JIM =1
  23 L=K+11
  READ 1, (KORD(I),I=K,L)
  L=L+1
  M=L+8
  READ 13, (KORD(I),I=L,M)
  K=M+1
  IF (JIM-12) 21,22,22
  21 JIM=JIM+1
  GO TO 23
  22 J=0
  I=1
  N=2
  K=21
  4 IF (KORD(1)-KORD(N)) 6,2,6
  2 I=I+2
  GO TO 16
  14 J=J+1
  KEEP(J)=KORD(1)
  GO TO 5
  16 IF (KORD(N)-KORD(1)) 12,3,12
  3 N=N+2
  IF (N-K) 4,14,14
  6 J=J+1
  KEEP(J)=KORD(1)

```


GILBERT ROLLER • STIIIA

PROGRAM STILLA

```

DIMENSION KORD (4, 16), INTERVAL (64)
FORMAT (4(14))
READ 1, ((KORD (I,N), I=1,4), N=1,16)
J=1
I=1
4 DO 5 N=1,16
  INTERV(J)=KORD(I,U)
  J=J+1
5 CONTINUE
  IF (I-16) 6,7,7
  I=I+1
  GO TO 4
7 PRINT 1, (INTERV(I), I=1,64)
STOP
END
END

```

0 0 2 2 0 0 0 0 2 0 0 0 0 0 0
0 0 4 4 0 0 7 7 3 5 7 2 0 0 0 0
4 4 7 7 8 8 11 11 9 9 11 5 0 0 0 0
12 12 12 12 12 12 12 12 12 12 12 12 0 0 0 0

STRAVINSKY - NO. Va - REDUCTION OF LONG
USE QUARTET PROGRAM Va

```

* 74981T ROLLER X      1MIN IC OP
PROGRAM STR5
DIMENSION INFO(2,16),MUS(3,16),NOTE(3,48),LONG(3,16),KORD(64)
  1 FORMAT (16(13,12))
  2 FORMAT (16(13,11,11))
 30 FORMAT (8(13,11,12))
  READ 1, ((INFO(I,N),I=1,2),N=1,16)
  READ 2, ((MUS(I,N),I=1,3),N=1,16)
  READ 2, ((NOTE(I,N),I=1,3),N=1,48)
  READ30, ((LONG(I,N),I=1,3),N=1,16)
  3 FORMAT (16(14))
  READ 3, (KORD(I),I=1,64)
 20 FORMAT (1H0,17,115,18X,2H0 ,113,110/)
 270 FORMAT (1H0,17,115,18X,2H2-,113,110/)
 271 FORMAT (1H0,17,115,19X,2H2.,113,110/)
 272 FORMAT (1H0,17,115,18X,2H3-,113,110/)
 273 FORMAT (1H0,17,115,18X,2H3.,113,110/)
 274 FORMAT (1H0,17,115,18X,2H4.,113,110/)
 275 FORMAT (1H0,17,115,18X,2H4+,113,110/)
 276 FORMAT (1H0,17,115,18X,2H5.,113,110/)
 277 FORMAT (1H0,17,115,18X,2H6-,113,110/)
 278 FORMAT (1H0,17,115,18X,2H6.,113,110/)
 279 FORMAT (1H0,17,115,18X,2H7-,113,110/)
 280 FORMAT (1H0,17,115,18X,2H7.,113,110/)
 281 FORMAT (1H0,17,115,18X,2H8.,113,110/)
 21 FORMAT (41X,2H0 ,113,110/)
 285 FORMAT (41X,2H2-,113,110/)
 286 FORMAT (41X,2H2.,113,110/)
 287 FORMAT (41X,2H3-,113,110/)
 288 FORMAT (41X,2H3.,113,110/)
 289 FORMAT (41X,2H4.,113,110/)
 290 FORMAT (41X,2H4+,113,110/)

```



```

5 PRINT 20,INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
4 IF (KORD(J)-2) 200,201,202
200 PRINT 270, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
201 PRINT 271, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
202 IF (KORD(J)-4) 203,204,205
203 PRINT 272, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
204 PRINT 273, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
205 IF (KORD(J)-6) 206,207,208
206 PRINT 274, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
207 PRINT 275, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
208 IF (KORD(J)-8) 209,210,211
209 PRINT 276, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
210 PRINT 277, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
211 IF (KORD(J)-10) 212,213,214
212 PRINT 278, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
213 PRINT 279, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
214 IF (KORD(J)-12) 215,216,217
215 PRINT 280, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
GO TO 217
216 PRINT 281, INFO(1,N),INFO(2,N),MUS(2,N),MUS(3,N)
217 J=J+16

```



```

DO 253 L=1,3
  9 IF (KORD(J)-0) 6,7,6
  7 PRINT 21, NOTE(2,N),NOTE(3,N)
    GO TO 235
  6 IF (KORD(J)-2) 218,219,220
  218 PRINT 285, NOTE(2,N),NOTE(3,N)
    GO TO 235
  219 PRINT 286, NOTE(2,N),NOTE(3,N)
    GO TO 235
  220 IF (KORD(J)-4) 221,222,223
  221 PRINT 287, NOTE(2,N),NOTE(3,N)
    GO TO 235
  222 PRINT 288, NOTE(2,N),NOTE(3,N)
    GO TO 235
  223 IF (KORD(J)-6) 224,225,226
  224 PRINT 289, NOTE(2,N),NOTE(3,N)
    GO TO 235
  225 PRINT 290, NOTE(2,N),NOTE(3,N)
    GO TO 235
  226 IF (KORD(J)-8) 227,228,229
  227 PRINT 291, NOTE(2,N),NOTE(3,N)
    GO TO 235
  228 PRINT 292, NOTE(2,N),NOTE(3,N)
    GO TO 235
  229 IF (KORD(J)-10) 230,231,232
  230 PRINT 293, NOTE(2,N),NOTE(3,N)
    GO TO 235
  231 PRINT 294, NOTE(2,N),NOTE(3,N)
    GO TO 235
  232 IF (KORD(J)-12) 233,234,235
  233 PRINT 295, NOTE(2,N),NOTE(3,N)
    GO TO 235

```



```

234 PRINT 296, NOTE(2,N),NOTE(3,N)
235 J=J+16
      N=N+16
253 CONTINUE
      N=1
      IF (LONG(1,N) -0) 10,11,10
11 PRINT 22, LONG(2,N),LONG(3,N)
      GO TO 302
10 IF (LONG(1,N)-2) 255,256,257
255 PRINT 315, LONG(2,N),LONG(3,N)
      GO TO 302
256 PRINT 316, LONG(2,N),LONG(3,N)
      GO TO 302
257 IF (LONG(1,N) -4) 258,259,260
258 PRINT 317, LONG(2,N),LONG(3,N)
      GO TO 302
259 PRINT 318, LONG(2,N),LONG(3,N)
      GO TO 302
260 IF (LONG(1,N) -6) 261,262,263
261 PRINT 319, LONG(2,N),LONG(3,N)
      GO TO 302
262 PRINT 320, LONG(2,N),LONG(3,N)
      GO TO 302
263 IF (LONG (1,N) -8) 264,265,266
264 PRINT 321, LONG(2,N),LONG(3,N)
      GO TO 302
265 PRINT 322, LONG(2,N),LONG(3,N)
      GO TO 302
266 IF (LONG(1,N) -10) 267,268,269
267 PRINT 323, LONG(2,N),LONG(3,N)
      GO TO 302

```


| | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|---|--|---|---|
| 00101001020010300201002020020300301003020030300401004020040300501005020050300601 | 00001000010000100001000010000100001000010000100001000010000100001000010000100001 | 00001000010000100001000010000100001000010000100001000010000100001000010000100001 | 00002000020000200002000020000200002000020000200002000020000200002000020000200002 | 00003000030000300003000030000300003000030000300003000030000300003000030000300003 | 00004000040000400004000040000400004000040000400004000040000400004000040000400004 | 001422001422004422004422005422005422006422006422 | 008422008422006422006422006422006422006422006422 | 12 | 4 4 4 7 7 8 8 11 11 9 9 11 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 0 0 0 4 4 0 0 7 7 3 5 7 2 0 | 0 0 0 2 2 0 0 0 0 0 2 0 |
|--|--|--|--|--|--|--|--|---|--|---|---|

* STRAVINSKY - NO. 5 - PRINT OUT - RESULTS

CODE

. IS MAJOR OR PERFECT
- IS MINOR
/ IS AUGMENTED

| MEASURE NO. | RHYTHMIC POSITION | HARMONIC INTERVALS | OCTAVE | INSTRUMENT |
|-------------|-------------------|----------------------------|-----------------------|------------------------|
| 1 | 1 | 8.
3.
0
0
C | 0
0
0
0
4 | 1
2
3
4
22 |
| 1 | 2 | 8.
3.
0
0
C | 0
0
0
0
4 | 1
2
3
4
22 |
| 1 | 3 | 8.
5.
3.
2.
E- | 0
0
0
0
4 | 1
2
3
4
22 |
| 2 | 1 | 8.
5.
3.
2.
E- | 0
0
0
0
4 | 1
2
3
4
22 |

2

2

8:6-000E

000004

123422

2

3

8:6-000E

000004

123422

3

1

8:7:5:0E

000004

123422

3

2

8:7:5:0E

000004

123422

3

3

8:6:3:2:0

000004

123422

| | | | | |
|---|---|--------------|-----------|------------|
| 4 | 1 | 8.6.4.0 G | 0 0 0 0 4 | 1 2 3 4 22 |
| 4 | 2 | 8.7.5.0 F | 0 0 0 0 4 | 1 2 3 4 22 |
| 4 | 3 | 8.4.2.0 F | 0 0 0 0 4 | 1 2 3 4 22 |
| 5 | 1 | 0 0 0 0 0 E- | 0 0 0 0 4 | 1 2 3 4 22 |
| 5 | 2 | 0 0 0 0 0 F | 0 0 0 0 4 | 1 2 3 4 22 |


```

* 74981T RULLEK QC          1 MINIC  UP
PROGRAM  STR6
DIMENSION KORD(4,12)
41  FORMAT (4(14))
1  FORMAT (4(14))
READ 1, ((KORD(I,N),I=1,4),N=1,12)
J=1
DO 40 N=1,12
K1=KORD(1,N)
K2=KORD(2,N)
K3=KORD(3,N)
K4=KORD(4,N)
J=J+1
DO 40 M=J,12
L1=KORD(1,M)
L2=KORD(2,M)
L3=KORD(3,M)
L4=KORD(4,M)
IF (K1-L1) 31,32,31
31 GO TO 40
32 IF (K2-L2) 31,34,31
34 IF (K3-L3) 31,35,31
35 IF (K4-L4) 31,36,31
36 IF (K1-0) 43,38,43
37 IF (K2-0) 43,38,43
38 IF (K3-0) 43,39,43
39 IF (K4-0) 43,40,43
43 IF (K1-12) 61,60,61
60 K1=8
GO TO 100
61 IF (K1-11) 63,62,63
62 K1=7
GO TO 100

```

| | |
|---------------|------------|
| 62 K1=7 | |
| GO TO 100 | |
| 63 IF (K1-10) | 65.64.65 |
| 64 K1=-7 | |
| GO TO 100 | |
| 65 IF (K1-9) | 67.66.67 |
| 66 K1=6 | |
| GO TO 100 | |
| 67 IF (K1-8) | 69.68.69 |
| 68 K1=-6 | |
| GO TO 100 | |
| 69 IF (K1-7) | 71.70.71 |
| 70 K1=5 | |
| GO TO 100 | |
| 71 IF (K1-6) | 73.72.73 |
| 72 K1=-5 | |
| GO TO 100 | |
| 73 IF (K1-5) | 75.74.75 |
| 74 K1=4 | |
| GO TO 100 | |
| 75 IF (K1-4) | 77.76.77 |
| 76 K1=3 | |
| GO TO 100 | |
| 77 IF (K1-3) | 79.78.79 |
| 78 K1=-3 | |
| GO TO 100 | |
| 79 IF (K1-2) | 81.80.81 |
| 80 K1=2 | |
| GO TO 100 | |
| 81 IF (K1-1) | 83.82.83 |
| 82 K1=-2 | |
| GO TO 100 | |
| 83 IF (K1-0) | 100.84.100 |
| 84 K1=1 | |

| | |
|----------------|-------------|
| 100 IF (K2-12) | 102,101,102 |
| 101 K2=8 | |
| GO TO 130 | |
| 102 IF (K2-11) | 104,103,104 |
| 103 K2=7 | |
| GO TO 130 | |
| 104 IF (K2-10) | 106,105,106 |
| 105 K2=-7 | |
| GO TO 130 | |
| 106 IF (K2-9) | 108,107,108 |
| 107 K2=6 | |
| GO TO 130 | |
| 108 IF (K2-8) | 110,109,110 |
| 109 K2=-6 | |
| GO TO 130 | |
| 110 IF (K2-7) | 112,111,112 |
| 111 K2=5 | |
| GO TO 130 | |
| 112 IF (K2-6) | 114,113,114 |
| 113 K2=-5 | |
| GO TO 130 | |
| 114 IF (K2-5) | 116,115,116 |
| 115 K2=4 | |
| GO TO 130 | |
| 116 IF (K2-4) | 118,117,118 |
| 117 K2=3 | |
| GO TO 130 | |
| 118 IF (K2-3) | 120,119,120 |
| 119 K2=-3 | |
| GO TO 130 | |
| 120 IF (K2-2) | 122,121,122 |
| 121 K2=2 | |
| GO TO 130 | |
| 122 IF (K2-1) | 124,123,124 |

| | |
|----------------|-------------|
| 123 K2--2 | |
| GO TO 130 | |
| 124 IF (K2-0) | 130,125,130 |
| 125 K2=1 | |
| 130 IF (K3-12) | 132,131,132 |
| 131 K3=8 | |
| GO TO 156 | |
| 132 IF (K3-11) | 134,133,134 |
| 133 K3=7 | |
| GO TO 156 | |
| 134 IF (K3-10) | 136,135,136 |
| 135 K3=-7 | |
| GO TO 156 | |
| 136 IF (K3-9) | 138,137,138 |
| 137 K3=6 | |
| GO TO 156 | |
| 138 IF (K3-8) | 140,139,140 |
| 139 K3=-6 | |
| GO TO 156 | |
| 140 IF (K3-7) | 142,141,142 |
| 141 K3=5 | |
| GO TO 156 | |
| 142 IF (K3-6) | 144,143,144 |
| 143 K3=-5 | |
| GO TO 156 | |
| 144 IF (K3-5) | 146,145,146 |
| 145 K3=4 | |
| GO TO 156 | |
| 146 IF (K3-4) | 148,147,148 |
| 147 K3=3 | |
| GO TO 156 | |
| 148 IF (K3-3) | 150,149,150 |
| 149 K3=-3 | |

GO TO 156
 150 IF (K3-2) 152,151,152
 151 K3=2
 GO TO 156
 152 IF (K3-1) 154,153,154
 153 K3=-2
 GO TO 156
 154 IF (K3-0) 156,155,156
 155 K3=1
 156 IF (K4-12) 161,160,161
 160 K4=8
 GO TO 185
 161 IF (K4-11) 163,162,163
 162 K4=7
 GO TO 185
 163 IF (K4-10) 165,164,165
 164 K4=-7
 GO TO 185
 165 IF (K4-9) 167,166,167
 166 K4=6
 GO TO 185
 167 IF (K4-8) 169,168,169
 168 K4=-6
 GO TO 185
 169 IF (K4-7) 171,170,171
 170 K4=5
 GO TO 185
 171 IF (K4-6) 173,172,173
 172 K4=-5
 GO TO 185
 173 IF (K4-5) 175,174,175
 174 K4=4
 GO TO 185
 175 IF (K4-4) 177,176,177

```

176 K4=3
    GO TO 185
177 IF (K4-3) 179,178,179
178 K4=-3
    GO TO 185
179 IF (K4-2) 181,180,181
180 K4=2
    GO TO 185
181 IF (K4-1) 183,182,183
182 K4=-2
    GO TO 185
183 IF (K4-0) 185,184,185
184 K4=1
185 PRINT 41, K1,K2,K3,K4
    KORD(1,M)=0
    KORD(2,M)=0
    KORD(3,M)=0
    KORD(4,M)=0
40 CONTINUE
    STOP
    END
    END

```

```

12 4 0 0 0
12 4 0 0 0
12 7 4 2 2
12 7 4 2 2
12 8 0 0 0
12 8 0 0 0
12 11 7 0 0
12 11 7 0 0
12 9 3 2 2
12 9 5 0 0
12 11 7 0 0
12 5 2 0 0

```

| * STRAVINSKY | - | NO. 6 | - | CHORD PROGRESSION | - | RESULTS |
|--------------|----|-------|---|-------------------|---|---------|
| 8 | 3 | 1 | | | | 1 |
| 8 | 5 | 3 | | | | 2 |
| 8 | -6 | 1 | | | | 1 |
| 8 | 7 | 5 | | | | 1 |

STRAVINSKY - NO. 7 - CHORD PROGRESSION
USE QUINTET PROGRAM VII

STRAVINSKY - NO. 7 - CHORD PROGRESSION - RESULTS

NO PROGRESSIONS

```

* 74981T ROLLER B          1MIN IC OP
PROGRAM STR8
  DIMENSION MUSA(2,16),MUSB(2,16),MUSC(2,16),MUSD(2,16),MUSE(2,16),
  1MUSG(2,16),MUSH(2,16),MUSI(2,16),KORA(2,16),KORB(2,16),KORC(2,16),
  2KORE(2,16),NOTA(2,16),NOTB(2,16),NOTC(2,16),NOTR(2,16),NOTP(2,16),
  3LONA(2,16),LONB(2,16),LONC(2,16),LOND(2,16),LONE(2,16),LUM(27),
  4KORT(231),KORD(231),MOE(26),JOE(14),KOE(10),LOE(2),MITH(4)
  1 FORMAT (14)
  2 FORMAT (8(13,13,4X))
  READ 2, ((MUSA(I,N),I=1,2),N=1,16)
  READ 2, ((MUSB(I,N),I=1,2),N=1,16)
  READ 2, ((MUSC(I,N),I=1,2),N=1,16)
  READ 2, ((MUSD(I,N),I=1,2),N=1,16)
  READ 2, ((MUSE(I,N),I=1,2),N=1,16)
  READ 2, ((MUSG(I,N),I=1,2),N=1,16)
  READ 2, ((MUSH(I,N),I=1,2),N=1,16)
  READ 2, ((MUSI(I,N),I=1,2),N=1,16)
  READ 2, ((KORA(I,N),I=1,2),N=1,16)
  READ 2, ((KORB(I,N),I=1,2),N=1,16)
  READ 2, ((KORC(I,N),I=1,2),N=1,16)
  READ 2, ((KORE(I,N),I=1,2),N=1,16)
  READ 2, ((NOTA(I,N),I=1,2),N=1,16)
  READ 2, ((NOTB(I,N),I=1,2),N=1,16)
  READ 2, ((NOTC(I,N),I=1,2),N=1,16)
  READ 2, ((NOTR(I,N),I=1,2),N=1,16)
  READ 2, ((NOTP(I,N),I=1,2),N=1,16)
  READ 2, ((LONA(I,N),I=1,2),N=1,16)
  READ 2, ((LONB(I,N),I=1,2),N=1,16)
  READ 2, ((LONC(I,N),I=1,2),N=1,16)
  READ 2, ((LOND(I,N),I=1,2),N=1,16)
  READ 2, ((LONE(I,N),I=1,2),N=1,16)
  11 FORMAT (13(13))

```

```

      READ11, (MOE(I),I=1,13)
14  FORMAT (7(I3))
      READ14, (JOE(I),I=1,7)
16  FORMAT (5(I3))
      READ16, (KOE(I),I=1,5)
18  FORMAT (13)
      READ18, (LOE(I),I=1,1)
      DO 232 M=1,16
        J=1
        N=1
        I=1
        K=21
        LUM(N) = MUSA(I,M)
        N=N+1
        LUM(N) = MUSB(I,M)
        N=N+1
        LUM(N) = MUSC(I,M)
        N=N+1
        LUM(N) = MUSD(I,M)
        N=N+1
        LUM(N) = MUSE(I,M)
        N=N+1
        LUM(N) = MUSG(I,M)
        N=N+1
        LUM(N) = MUSH(I,M)
        N=N+1
        LUM(N) = MUSI(I,M)
        N=N+1
        LUM(N) = KORA(I,M)
        N=N+1
        LUM(N) = KOR3(I,M)

```

```

N=N+1
LUM(N) = KORC(I,M)
N=N+1
LUM(N) = KORE(I,M)
N=N+1
LUM(N) = NOTA(I,M)
N=N+1
LUM(N) = NOTB(I,M)
N=N+1
LUM(N) = NOTC(I,M)
N=N+1
LUM(N) = NOTR(I,M)
N=N+1
LUM(N) = NOTP(I,M)
N=N+1
LUM(N) = LONA(I,M)
N=N+1
LUM(N) = LONB(I,M)
N=N+1
LUM(N) = LONC(I,M)
N=N+1
LUM(N) = LOND(I,M)
N=N+1
LUM(N) = LONE(I,M)
8 DO 7 I=1,K
  IF (LUM(N)) 4,3,4
  4 IF (LUM(I)) 5,3,5
  5 KORD(I) = LUM(I) - LUM(N)
  IF (KORD(I)) 10,7,7
  10 KORD(I)=-1*KORD(I)
  GO TO 7
  3 KORD(I)=0

```



```

7 CONTINUE
  IF (M-2) 9,9,6
6 N=N-1
  K=K-1
  GO TO 8
9 N=12
  DO 183 I=1,231
    J=KORD(I)
    IF (J-N) 80,80,86
80 KORT(I)=KORD(I)
    GO TO 182
86 K=0
163 N=N+88
    K=K+88
    IF (J-N) 164,164,165
164 GO TO 170
165 N=N+12
    K=K+12
    IF (J-N) 166,166,167
166 GO TO 170
167 IF (L-6) 168,170,170
168 L=L+1
    GO TO 163
170 KORT(I)=KORD(I)-K
182 N=12
183 CONTINUE
    L=1
186 LEW =20
    N=22
    K=21
    IT =1

```

```

DO 66 I=1,231
  IF (KORT(I)-MOE(L)) 140,20,140
20  JROOT=KORT(I)
  DO 32 J=1,7
    IF (JROOT-JOE(J)) 32,31,32
31  MITH(1) =LUM(N)
    GO TO 68
32  CONTINUE
    DO 34 J=1,5
      IF (JROOT-KOE(J)) 34,33,34
33  MITH(1) = LUM(IT)
      GO TO 68
34  CONTINUE
      IF (JROOT-LOE(1)) 140,35,140
35  MITH(1)=0
      GO TO 68
140 IF (I-K) 21,22,66
21  IT=IT+1
      GO TO 66
22  K=K+LEW
      N=N-1
      IT=IT-LEW
      LEW=LEW-1
66  CONTINUE
67  IF (L-13) 69,68,68
69  L=L+1
      GO TO 186
68  PRINT 1, MITH(1)
232 CONTINUE
      STOP
      END
      END

```


* STRAVINSKY - NO. 8 - HUNDRETH ROOTS - RESULTS

401
401
404
404
601
601
404
406
408
408
406
508
404
406
0
0

```

* 74981T ROLLER (ST9)      2MIN IC  OP
PROGRAM STRA9
DIMENSION MUS(2,16),NOTE(2,16),KLANG(2,16),LONG(2,16),KAN(16),
1 KON(16),KUN(16),KEN(16),KIN(16),MITH(16)
2 FORMAT (8(13,13,4X))
3 FORMAT (8(14))
READ 2, ((MUS(1,N),I=1,2),N=1,16)
PEAD 2, ((NOTE(1,N),I=1,2),N=1,16)
READ 2, ((KLANG(1,N),I=1,2),N=1,16)
READ 2, ((LONG(1,N),I=1,2),N=1,16)
READ 3, (MITH(1),I=1,16)
13 FORMAT (1H0,5X,17HMELODIC INTERVALS)
14 FORMAT (1H0,5X,8H1 VIOLIN)
15 FORMAT (1H0,16(14))
16 FORMAT (1H0,5X,8H2 VIOLIN)
17 FORMAT (1H0,5X, 5HVIOLA)
18 FORMAT (1H0,5X,5HCELLO)
19 FORMAT (1H0,5X,15HHINDEMTITH ROOTS)
IAN=0
MAN=0
DO 8 N=1,16
IAN=MAN
JAN=MUS(1,N)
IF (IAN) 30,20,30
30 IF (JAN) 21,20,21
20 KAN(N)=0
MAN = JAN
GO TO 8
21 KAN(N)=IAN-JAN
MAN=JAN
8 CONTINUE
IAN=0
MAN=0

```



```

DO 9 N=1,16
  IAN=MAN
  JAN=NOTE(1,N)
  IF (IAN) 31,22,31
31 IF (JAN) 23,22,23
22 KON(N)=0
  MAN=JAN
  GO TO 9
23 KON(N)=IAN-JAN
  MAN=JAN
  9 CONTINUE
  IAN=0
  MAN=0
  DO 10 N=1,16
    IAN=MAN
    JAN=KLANG(1,N)
    IF (IAN) 32,24,32
32 IF (JAN) 25,24,25
24 KUN(N)=0
  MAN=JAN
  GO TO 10
25 KUN(N)=IAN-JAN
  MAN=JAN
  10 CONTINUE
  IAN=0
  MAN=0
  DO 11 N=1,16
    IAN=MAN
    JAN=LONG(1,N)
    IF (IAN) 33,26,33
33 IF (JAN) 27,26,27
26 KEN(N)=0
  MAN=JAN

```



```

GO TO 11
27 KEN(N)=IAN-JAN
MAN=JAN
11 CONTINUE
IAN=0
MAN=0
DO 12 N=1,16
IAN=MAN
JAN=MITH(N)
IF (IAN) 34,28,34
34 IF (JAN) 29,28,29
28 KIN(N)=0
MAN=JAN
GO TO 12
29 KIN(N)=IAN-JAN
MAN = JAN
12 CONTINUE
PRINT 13
PRINT 14
PRINT 15, (KAN(I),I=1,16)
PRINT 16
PRINT 15, (KON(I),I=1,16)
PRINT 17
PRINT 15, (KUN(I),I=1,16)
PRINT 18
PRINT 15, (KEN(I),I=1,16)
PRINT 19
PRINT 15, (KIN(I),I=1,16)
STOP
END
END

```


| | | | | | | |
|---------|---------|---------|---------|--------|--------|--------|
| 601601 | 601601 | 508501 | 601601 | 601601 | 601601 | 605601 |
| 508501 | 605601 | 508501 | 605601 | 605601 | 508501 | 000001 |
| 501502 | 501502 | 411402 | 505502 | 505502 | 605602 | 605602 |
| 508502 | 601602 | 508502 | 601602 | 505502 | 505502 | 2 |
| 605603 | 605603 | 508503 | 601603 | 601603 | 605603 | 605603 |
| 508503 | 605603 | 508503 | 605603 | 506503 | 3 | 3 |
| 601604 | 601604 | 411404 | 505504 | 505504 | 601604 | 604604 |
| 411404 | 601604 | 505504 | 601604 | 505504 | 4 | 4 |
| 401 401 | 404 404 | 601 406 | 406 406 | | | |
| 403 408 | 406 508 | 404 406 | 000 000 | | | |

```

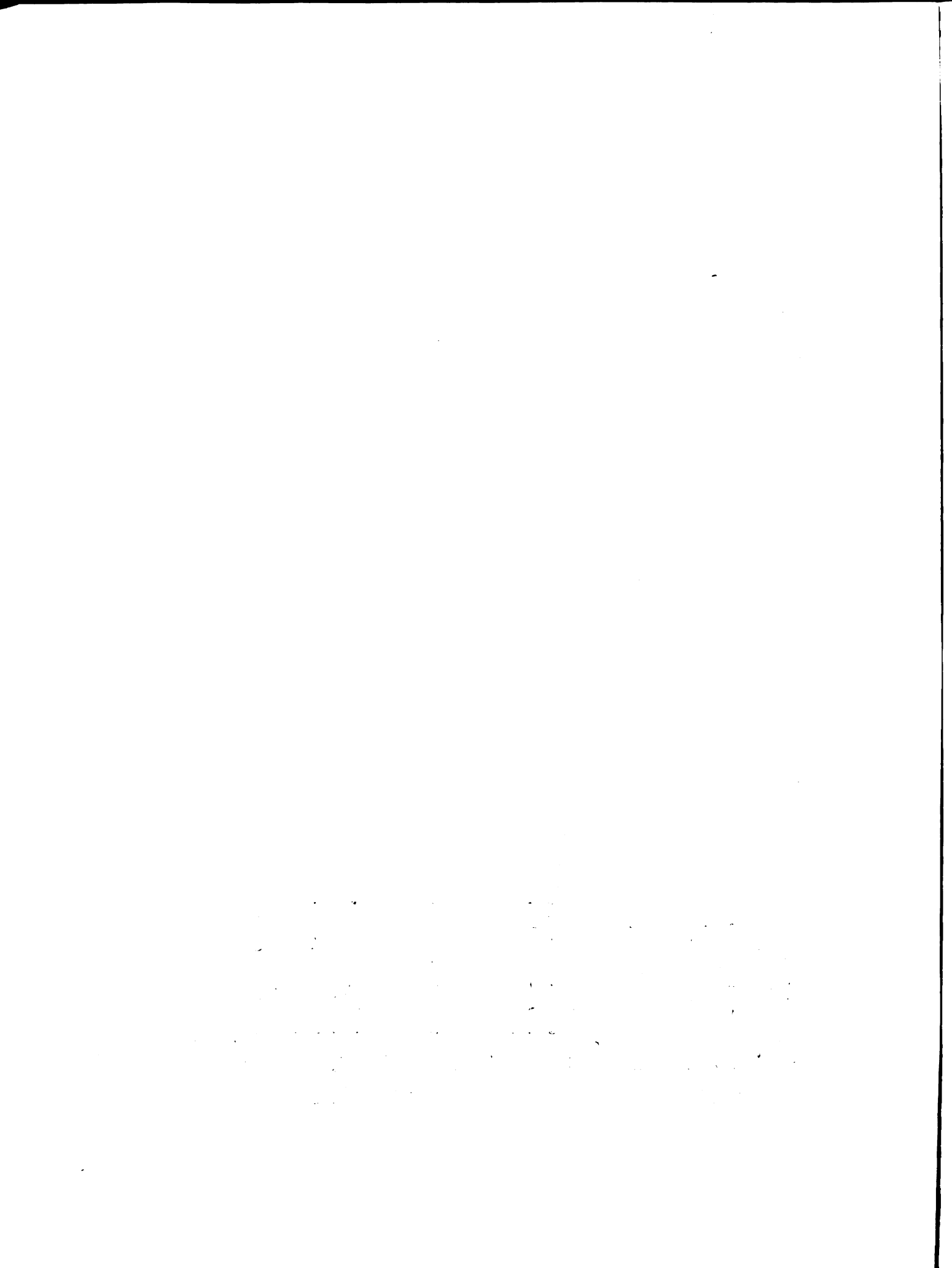
* 74981T ROLLER H          2MIN 1C  OP
PROGRAM STRA10
DIMENSION KAN(80),MELO(30)
30 FORMAT (16(14))
READ 30, (KAN(I),I=1,80)
105 FORMAT (1H0.4H P1.9H MINOR 2.4H M2.9H MINOR 3.4H M3.4H P4,
14H A4.4H P5.9H MINOR 6.4H M6.9H MINOR 7.4H M7.4H P8)
106 FORMAT (1H0.1X.4(13.3X).3X.3(13.1X).13.3(3X.13).4X.13.2X.13)
107 FORMAT (1H0.17HMELODIC INTERVALS)
111 FORMAT (1H0.8H1 VIOLIN)
112 FORMAT (1H0.9H11 VIOLIN)
113 FORMAT (1H0.5HVIOLA)
114 FORMAT (1H0.5HCELO)
115 FORMAT (1H0.15HHINDEWITH ROOTS)
PRINT 111
M=1
N=12
NO = 1
NOT=16
86 IT = 1
31 DO 29 I= NO,NOT
L=1
51 IF (KAN(1)) 40,41,41
40 GO TO 52
41 J=KAN(1)
50 IF (J-N) 50,50,57
50 KAN(1)=KAN(1)
GO TO 76
57 K=0
47 N=N+88
K=K+88
IF (J-N) 58,58,59

```

```

58 GO TO 70
59 N = N+12
   K=K+12
   IF (J-N) 60,60,65
60 GO TO 70
65 IF (L-6) 66,70,70
66 L=L+1
   GO TO 47
70 KAN(I) = KAN(I) - K
76 N=12
   GO TO 29
52 J=KAN(I)
   IF (J+N) 54,53,53
53 KAN(I) = KAN(I)
   GO TO 17
54 K=0
10 N=N+88
   K=K+88
   IF (J+N) 12,11,11
11 GO TO 16
12 N=N+12
   K=K+12
   IF (J+N) 14,13,13
13 GO TO 16
14 IF (L-6) 15,16,16
15 L=L+1
   GO TO 10
16 KAN(I)=KAN(I)+K
17 N=12
29 CONTINUE
   PRINT 107

```



```

PRINT 106, (KAN(I),I=NO,NOT)
DO 6 I=1,26
  MELO(I) = 0
  6 CONTINUE
100 MOT = 0
88 DO 83 I= 1,13
  IF (KAN(I)-MOT) 93,82,93
82 MELO(I) = MELO(I)+1
93 MOT = MOT+1
83 CONTINUE
MOT = 0
DO 85 I= 14,26
  IF (KAN(I) + MOT) 95,84,95
84 MELO(I) = MELO(I) - 1
95 MOT = MOT+1
85 CONTINUE
  IF (IT- NOT) 20,87,87
20 IT = IT +1
  GO TO 100
87 PRINT 105
PRINT 106, (MELO(I),I=1,13)
PRINT 106, (MELO(I),I=14,26)
  IF (M-5) 18,18,110
18 NO=NO+16
  NOT=NOT+16
  N=12
  M=M+1
  IF (M-3) 116,117,118
116 PRINT 112
  GO TO 125
117 PRINT 113
  GO TO 125

```

```

118 IF (M-5) 119,120,110
119 PRINT 114
    GO TO 125
120 PRINT 115
125 GO TO 31
110 GO TO 121
121 STOP
    END
    END

```

| | | | | | | | | | | | | | | | |
|---|---|-----|---|-----|---|------|----|-----|------|----|----|------|----|---|---|
| 0 | 0 | 93 | 0 | -93 | 0 | -4 | 0 | 97 | -97 | 4 | 93 | -97 | 97 | 0 | 0 |
| 0 | 0 | 90 | 0 | -94 | 0 | -100 | 0 | 97 | -93 | 96 | -3 | -93 | 96 | 0 | 0 |
| 0 | 0 | 97 | 0 | -93 | 0 | -4 | 0 | 97 | -97 | 4 | 93 | -97 | 97 | 0 | 0 |
| 0 | 0 | 190 | 0 | -94 | 0 | -96 | -3 | -93 | -190 | 96 | 94 | -190 | 96 | 0 | 0 |
| 0 | 0 | -3 | 0 | -1 | 0 | -1 | 0 | -2 | 0 | 2 | 0 | 2 | -2 | 0 | 0 |

STRAVINSKY - NO.10 - MELODIC INTERVALS AND ROOT MOVEMENTS - RESULTS

I VIOLIN

MELODIC INTERVALS

[illegible]

II VIOLIN

MELODIC INTERVALS

| | | | | | | | | | | | | |
|----|---------|----|---------|----|----|-----|----|---------|----|---------|----|----|
| 0 | 0 | 2 | 0 | -6 | 0 | -12 | 0 | 9 | -5 | 8 | -3 | -5 |
| 8 | 0 | 0 | | | | | | | | | | |
| P1 | MINOR 2 | M2 | MINOR 3 | M3 | P4 | A4 | P5 | MINOR 6 | M6 | MINOR 7 | M7 | P8 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 |
| -8 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | -2 | 0 | 0 | 0 |

VIOLA

MELODIC INTERVALS

| | P1 | MINOR 2 | M2 | MINOR 3 | M3 | P4 | A4 | P5 | MINOR 6 | M6 | MINOR 7 | M7 | P8 |
|----|----|---------|----|---------|----|----|----|----|---------|----|---------|----|----|
| 0 | 0 | 0 | 0 | -5 | 0 | -4 | 0 | 9 | -9 | 4 | 5 | -9 | |
| 9 | 0 | 0 | | | | | | | | | | | |
| 8 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| -8 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | -2 | 0 | 0 | 0 |

CELO

MELODIC INTERVALS


```

*      GILBERT ROLLER -N56T
1  FORMAT (6(I4))
      NO1=1
      DO 5 I = 2,8
      NO2=1
      L=I+1
      DO 5 J= L,9
      NO3=J
      K=J+1
      DO 5 M=K,10
      NO4=M
      N=M+1
      DO 5 I1=N,11
      NO5=I1
      JJ=I1+1
      DO 5 KK = JJ,12
      NO6=KK
      PRINT 1, NO1,NO2,NO3,NO4,NO5,NO6
      PUNCH 1, NO1,NO2,NO3,NO4,NO5,NO6
5  CONTINUE
      STOP
      END

```

* GILBERT ROLLER -N56T - RESULTS

| | | | | | |
|---|---|---|---|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 2 | 3 | 4 | 5 | 7 |
| 1 | 2 | 3 | 4 | 5 | 8 |
| 1 | 2 | 3 | 4 | 5 | 9 |
| 1 | 2 | 3 | 4 | 5 | 10 |
| 1 | 2 | 3 | 4 | 5 | 11 |
| 1 | 2 | 3 | 4 | 5 | 12 |
| 1 | 2 | 3 | 4 | 6 | 7 |
| 1 | 2 | 3 | 4 | 6 | 8 |
| 1 | 2 | 3 | 4 | 6 | 9 |
| 1 | 2 | 3 | 4 | 6 | 10 |
| 1 | 2 | 3 | 4 | 6 | 11 |
| 1 | 2 | 3 | 4 | 6 | 12 |
| 1 | 2 | 3 | 4 | 7 | 8 |
| 1 | 2 | 3 | 4 | 7 | 9 |
| 1 | 2 | 3 | 4 | 7 | 10 |
| 1 | 2 | 3 | 4 | 7 | 11 |
| 1 | 2 | 3 | 4 | 7 | 12 |
| 1 | 2 | 3 | 4 | 8 | 9 |
| 1 | 2 | 3 | 4 | 8 | 10 |
| 1 | 2 | 3 | 4 | 8 | 11 |
| 1 | 2 | 3 | 4 | 8 | 12 |
| 1 | 2 | 3 | 4 | 9 | 10 |
| 1 | 2 | 3 | 4 | 9 | 11 |
| 1 | 2 | 3 | 4 | 9 | 12 |
| 1 | 2 | 3 | 4 | 10 | 11 |
| 1 | 2 | 3 | 4 | 10 | 12 |
| 1 | 2 | 3 | 4 | 11 | 12 |
| 1 | 2 | 3 | 5 | 6 | 7 |
| 1 | 2 | 3 | 5 | 6 | 8 |

| | | | | | |
|---|---|---|----|----|----|
| 1 | 4 | 6 | 7 | 10 | 11 |
| 1 | 4 | 6 | 7 | 10 | 12 |
| 1 | 4 | 6 | 7 | 11 | 12 |
| 1 | 4 | 6 | 8 | 9 | 10 |
| 1 | 4 | 6 | 8 | 9 | 11 |
| 1 | 4 | 6 | 8 | 9 | 12 |
| 1 | 4 | 6 | 8 | 10 | 11 |
| 1 | 4 | 6 | 8 | 10 | 12 |
| 1 | 4 | 6 | 8 | 11 | 12 |
| 1 | 4 | 6 | 8 | 10 | 11 |
| 1 | 4 | 6 | 9 | 10 | 12 |
| 1 | 4 | 6 | 9 | 11 | 12 |
| 1 | 4 | 6 | 10 | 11 | 12 |
| 1 | 4 | 7 | 8 | 9 | 10 |
| 1 | 4 | 7 | 8 | 9 | 11 |
| 1 | 4 | 7 | 8 | 9 | 12 |
| 1 | 4 | 7 | 8 | 10 | 11 |
| 1 | 4 | 7 | 8 | 10 | 12 |
| 1 | 4 | 7 | 8 | 11 | 12 |
| 1 | 4 | 7 | 9 | 10 | 11 |
| 1 | 4 | 7 | 9 | 10 | 12 |
| 1 | 4 | 7 | 9 | 11 | 12 |
| 1 | 4 | 7 | 10 | 11 | 12 |
| 1 | 4 | 8 | 9 | 10 | 11 |
| 1 | 4 | 8 | 9 | 10 | 12 |
| 1 | 4 | 8 | 9 | 11 | 12 |
| 1 | 4 | 8 | 10 | 11 | 12 |
| 1 | 4 | 9 | 10 | 11 | 12 |
| 1 | 5 | 6 | 7 | 8 | 9 |
| 1 | 5 | 6 | 7 | 8 | 10 |
| 1 | 5 | 6 | 7 | 8 | 11 |
| 1 | 5 | 6 | 7 | 8 | 12 |
| 1 | 5 | 6 | 7 | 9 | 10 |
| 1 | 5 | 6 | 7 | 9 | 11 |


```

*      GILBERT ROLLER - RETRO H
      DIMENSION  NOTE(6),LPETRO (12)
2  FORMAT (6(I3))
      READ 2,(NOTE (I) , I=1,6)
      MOT =1
      N=1
      DO 8 M=1,12
3  J=0
      DO 6 I= 1,6
      IF (MOT-NOTE(I)) 6,4,6
4  J=J+1
      6 CONTINUE
      IF (J-1) 7,8,8
7  LPETRO(N)=MOT
      N=N+1
      8 MOT = MOT+1
      5 FORMAT (1H0,6(I3))
      PRINT5, (LPETRO(I), I=1,6)
      STOP
      END

```

```

1 2 3 4 5 6

```

• GILBERT ROLLER - RETRO H - RESULTS

7 8 9 10 11 12

```

*      GILBERT ROLLER - RETRO (A)
      DIMENSION NOTE(6), KOY(5),LOY(8),MUS(48)
42  FORMAT (6(I3))
      READ 42, (NOTE(I),I=1,6)
43  FORMAT (5(I3))
      READ 43, (KOY(I),I=1,5)
44  FORMAT (20(I3))
      READ 44, (LOY(I),I=1,8)
      JJ=9
      LO=0
      KO=3
      DO 71 J=1,5
      KK= KOY(J)
      LO=LO+1
      KO=KO+1
      DO 71 K=LO,KO
      LL= LOY(K)
      L=1
50  N=1
      DO 60 I=1,6
      IF (JJ-NOTE(I))52,51,52
51  GO TO 60
52  IF (KK-NOTE(I)) 54,53,54
53  GO TO 60
54  IF (LL-NOTE(I)) 56,55,56
55  GO TO 60
56  IF (N-2) 57,58,59
57  INC1=NOTE(I)
      N=N+1
      GO TO 60
58  INC2=NOTE(I)
      N=N+1
      GO TO 60

```

```
59 INC3=NOTE(I)
   N=N+1
60 CONTINUE
   IF (L-2) 61,62,63
61 MM=INC1
   NN=INC2
   II=INC3
   GO TO 70
62 MM=INC2
   NN=INC3
   II=INC1
   GO TO 70
63 IF (L-4) 64,65,66
64 MM=INC3
   NN=INC1
   II=INC2
   GO TO 70
65 MM=INC3
   NN=INC2
   II=INC1
   GO TO 70
66 IF (L-6) 67,68,70
67 MM=INC2
   NN=INC1
   II=INC3
   GO TO 70
68 MM=INC1
   NN=INC3
   II=INC2
70 DO 108 M=1,6
   IF (M-2) 100,101,102
100 MUS(M) = JJ
   GO TO 108
```

```

101 MUS(M)=KK
    GO TO 108
102 IF (M-4) 103,104,105
103 MUS(M)=LL
    GO TO 108
104 MUS(M)=MM
    GO TO 108
105 IF (M-6) 106,107,108
106 MUS(M)=NN
    GO TO 108
107 MUS(M)=11
108 CONTINUE
    45 FORMAT ( 1H0, 6(I4))
    PRINT 45, (MUS(I),I=1,6)
    46 FORMAT (6(I4))
    PUNCH 46, (MUS(I),I=1,6)
    IF (L-6) 109,71,71
109 L=L+1
    GO TO 50
71 CONTINUE
    STOP
    END

```

```

9 1 4 5 8 11
1 4 5 8 11
4 5 8 11 1 4 5 8

```

* GILBERT ROLLER - RETRO (A) - RESULTS

| | | | | | |
|---|---|----|----|----|----|
| 9 | 1 | 4 | 5 | 8 | 11 |
| 9 | 1 | 4 | 8 | 11 | 5 |
| 9 | 1 | 4 | 11 | 5 | 8 |
| 9 | 1 | 4 | 11 | 8 | 5 |
| 9 | 1 | 4 | 8 | 5 | 11 |
| 9 | 1 | 4 | 5 | 11 | 8 |
| 9 | 1 | 5 | 4 | 8 | 11 |
| 9 | 1 | 5 | 8 | 11 | 4 |
| 9 | 1 | 5 | 11 | 4 | 8 |
| 9 | 1 | 5 | 11 | 8 | 4 |
| 9 | 1 | 5 | 8 | 4 | 11 |
| 9 | 1 | 5 | 4 | 11 | 8 |
| 9 | 1 | 8 | 4 | 5 | 11 |
| 9 | 1 | 8 | 5 | 11 | 4 |
| 9 | 1 | 8 | 11 | 4 | 5 |
| 9 | 1 | 8 | 11 | 5 | 4 |
| 9 | 1 | 8 | 4 | 11 | 5 |
| 9 | 1 | 11 | 4 | 5 | 8 |
| 9 | 1 | 11 | 5 | 8 | 4 |
| 9 | 1 | 11 | 8 | 4 | 5 |
| 9 | 1 | 11 | 8 | 5 | 4 |
| 9 | 1 | 11 | 5 | 4 | 8 |
| 9 | 1 | 11 | 4 | 8 | 5 |
| 9 | 4 | 5 | 1 | 8 | 11 |
| 9 | 4 | 5 | 8 | 11 | 1 |
| 9 | 4 | 5 | 11 | 1 | 8 |
| 9 | 4 | 5 | 11 | 8 | 1 |
| 9 | 4 | 5 | 8 | 1 | 11 |
| 9 | 4 | 5 | 1 | 11 | 8 |
| 9 | 4 | 8 | 1 | 5 | 11 |
| 9 | 4 | 8 | 5 | 11 | 1 |
| 9 | 4 | 8 | 11 | 1 | 5 |
| 9 | 4 | 8 | 11 | 5 | 1 |

[illegible]

8 4 5 5 4 8 1 5 1 8 5 8 1 4 1 8 4 5 1 4 1 5 4
5 8 4 8 5 5 3 1 5 1 3 4 3 1 4 1 3 4 5 1 4 1 5
4 5 8 4 8 1 5 8 8 5 1 1 4 3 8 4 1 1 4 3 5 4 1
1 1 1 1 1 4 4 4 4 4 4 5 5 5 5 5 5 3 8 3 8 8 8
1
9 9

```

*      GILBERT ROLLER - RETRO - KOMPAR
      DIMENSION LRETRO (720),KAN(600),INT(5),KOMPAR(6)
      3 FORMAT (1H0,6(14))
      1 FORMAT (6(14))
      READ 1, (LRETRO(I),I=1,720)
      2 FORMAT (5(14))
      READ 2, (INT(I), I=1,5)
      K=1
      J=1
      N=1
      I=1
      30 KAN(K)=LRETRO(J)-LRETRO(J+1)
      IF (KAN(K)) 4,35,35
      4 KAN(K) = KAN(K)+12
      35 IF (I-5) 5,6,6
      5 I=I+1
      K=K+1
      J=J+1
      GO TO 30
      6 J=J+2
      K=K+1
      I=1
      IF (N-120) 8,31,31
      8 N=N+1
      GO TO 30
      31 L=1
      J=1
      M=1
      7 N=0
      DO 10 I=1,5
      IF (INT(I)-KAN(M)) 11,9,11
      9 N=N+1
      M=M+1
      10 CONTINUE

```

```

11 IF (N-1) 25,26,27
25 M=M+5
   GO TO 14
26 M=M+4
   GO TO 14
27 IF (N-3) 28,29,12
28 M=M+3
   GO TO 14
29 M=M+2
   GO TO 14
30 IF (N-5) 40,41,41
40 M=M+1
   GO TO 14
41 DO 13 K=1,6
   KOMP(K)=LRETRO(J)
   J=J+1
13 CONTINUE
   PRINT 3,(KOMP(K),K=1,6)
   L=L+6
   GO TO 20
14 DO 15 K=1,6
   KOMP(K)=0
   LRETRO(L)=0
   L=L+1
15 CONTINUE
   J=J+6
20 IF (J-720) 16,17,17
16 GO TO 7
17 GO TO 18
18 STOP
   END

```


[illegible]

[illegible]

* GILBERT ROLLER - RETRO-ACTIVAR - , RESULTS

9 8 5 4 1 11

BIBLIOGRAPHY

Books

- Alt, Franz L. (ed.). Advances in Computers. Vol. I. New York: Academic Press, 1960.
- Alt, Franz L. and Rubinoﬀ, M. Advances in Computers. Vol. III. New York: Academic Press, 1962.
- Andersen, A. O. Lessons in Harmony, Book 3. Boston: C. C. Birchard and Company, 1938.
- Apel, Willi (ed.). Harvard Dictionary of Music. Cambridge: Harvard University Press, 1956.
- Arnold, F. T. The Art of Accompaniment from a Thoroughbass. London: Oxford University Press, 1931.
- Berkeley, Edmund C. Giant Brains or Machines that Think. New York: John Wiley & Sons, Inc., 1949.
- Berkeley, Edmund C. and Wainwright, Lawrence. Computers, Their Operation and Applications. New York: Reinhold Publishing Corporation, 1956.
- Blom, Eric (ed.). Grove's Dictionary of Music and Musicians. New York: St. Martin's Press, Inc., 1954.
- Booth, A. D. and K. H. V. Automatic Digital Calculators. New York: Academic Press, 1956.
- Bukofzer, Manfred F. Music in the Baroque Era. New York: W. W. Norton and Company, 1947.
- Bullis, Carleton. Harmonic Forms. Cleveland: Clifton Press, 1933.
- Burrowes, J. F. Thorough-Bass Primer. Boston: Oliver Ditson and Company, 1874.
- Carpenter, Nan Cooke. Music in the Medieval and Renaissance University. Norman: University of Oklahoma Press, 1958.

- Chailley, Jacques and Leduc, Alphonse. Traite Historique d'Analyse Musicale. Paris, 1947.
- Charlier, Henri. Jean-Philippe Rameau. Lyon, 1955.
- Chorafas, D. N. Programming Systems for Electronic Computers. London: Butterworths, 1962.
- Colman, H. L. Smallwood, C., and Brown, G. W. Computer Language. New York: McGraw-Hill Book Company, 1962.
- Combarieu, Jules. Music, Its Laws and Evolution. New York: D. Appleton and Company, 1910.
- Control Data Corporation. Fortran System. Minneapolis: Control Data Corporation, 1961.
- Cutter, Benjamin. Harmonic Analysis. Philadelphia: Oliver Ditson Company, 1930.
- Einstein, Alfred. A Short History of Music. New York: Alfred A. Knopff, 1954.
- Fahnestock, James D. Computers and How They Work. New York: Ziff-Davis Publishing Company, 1959.
- Foote, Arthur and Spalding, Walter. Modern Harmony. New York: Arthur P. Schmidt Company, 1926.
- Girdleston, Cuthbert. Jean-Philippe Rameau. London: Cassell and Company, Ltd., 1957.
- Goodrich, A. J. Analytical Harmony. New York: John Church Company, 1893.
- Grimm, Carl W. Modern Harmony. Cincinnati: George B. Jennings Company, 1900.
- Hansen, Peter S. An Introduction to Twentieth Century Music. Boston: Allyn and Bacon, Inc., 1961.
- Helmholtz, Hermann. On the Sensations of Tone. New York: Longmans, Green and Company, 1895.
- Hiller, Lejaren A. and Isaacson, Leonard M. Experimental Music. New York: McGraw-Hill Book Company, 1959.
- Hindemith, Paul. Traditional Harmony. New York: Associated Music Publishers, 1944.
- Inall, T. E. (ed.). Electronic Computers. London: Iliffe and Sons, Ltd., 1956.

- Jonas, Oswald (ed.). Heinrich Schenker: Harmony. Chicago: University of Chicago Press, 1954.
- Lang, Paul Henry. Music in Western Civilization. New York: W. W. Norton and Company, 1941.
- Ledley, Robert Steven. Programming and Utilizing Digital Computers. New York: McGraw-Hill Book Company, 1962.
- Lehmann, F. J. Harmonic Analysis. Oberlin, Ohio: A. G. Comings and Son, 1910.
- MacFarren, G. A. Treatise on Harmony. London: Harrison & Sons, 1885.
- Machlis, Joseph. Introduction to Contemporary Music. New York: W. W. Norton and Company, 1961.
- McCoy, William J. Cumulative Harmony. New York: Ginn and Company, 1916.
- McHose, Allen Irvine. The Contrapuntal Harmonic Technique of the Eighteenth Century. New York: Appleton-Century-Crofts, Inc., 1947.
- Mulder, Ernest W. Harmonie. New York: W. De Haan, 1947.
- Piston, Walter. Principles of Harmonic Analysis. Boston: E. C. Schirmer Company, 1933.
- Postley, John A. Computers and People. New York: McGraw-Hill Book Company, 1960.
- Rameau, Jean-Philippe. A Treatise on Music. London: Fielding and Walker, 1737.
- Riemann, Hugo. Handbuch Der Harmonielehre. Leipzig: Breitkopf und Hartel, 1912.
- _____. Harmonielehre. Leipzig: Breitkopf und Hartel, 1880.
- Salazar, Adolfo. Music in Our Time. New York: W. W. Norton and Company, 1946.
- Schillinger, Joseph. Schillinger System of Musical Composition. New York: Carl Fischer, Inc., 1946.
- Shephard, F. H. Harmony Simplified. New York: G. Schirmer Company, 1896.
- Siegel, Paul. Understanding Digital Computers. New York: John Wiley and Sons, 1961.

- Slominsky, Nicholas (ed.). Baker's Biographical Dictionary of Musicians. New York: G. Schirmer Company, 1958.
- Strunk, Oliver. Source Readings in Music History. New York: W. W. Norton and Company, 1950.
- Tiersch, Otto. Harmonielehre. Leipzig: Breitkopf und Hartel, 1868.
- Von Handel, Paul (ed.). Electronic Computers. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1961.
- Woodward, William H. Studies in Education During the Age of the Renaissance. London: Cambridge University Press, 1906.

Articles and Periodicals

- Berkeley, Edmund. "Braille Translations," Computers and Automation, VIII (October, 1959), 12.
- _____. "Computer Art," Computers and Automation, XII (January, 1963), 8.
- Cone, Edward T. "Analysis Today," Musical Quarterly, XLVI (April, 1960), 174.
- Eckert, Wallace J. "Calculating Machines," Encyclopedia Americana, V (1957), 161-162d.
- Higgins, Phyllis. "Three-Part Music with a Computer as One Part," Computers and Automation, VII (March, 1958), 8.
- Hiller, Lejaren A. "Computer Music," Scientific American, CCI, No. 6 (December, 1959), 109-120.
- Hindemith, Paul. "Methods of Music Theory," Musical Quarterly, XXX (January, 1944), 22-23.
- Katz, Adele T. "Heinrich Schenker's Method of Analysis," Musical Quarterly, XXI, No. 3 (July, 1935), 311-329.
- MacDonald, Neil. "Music by Automatic Computers," Computers and Automation, VII (March, 1958), 8.
- _____. "Over 500 Areas of Application of Computers," Computers and Automation, X (June, 1961), 133-137.
- Papworth, D. G. "Computers and Change Ringing," Computer Journal, III (April, 1960), 47.
- Slominsky, Nicholas. Book Review of the Schillinger System of Musical Composition, Musical Quarterly, XXXII (July, 1956), 464-470.

Von Glasefield, E., Perschke, S., and Samet, E. "Human Translation and Translation by Machine," Engineer, CCXII (September, 1961), 442-443.

Unpublished Material

McLean, Barton. "Evolution of Music Theory from 850 to 1960." Paper prepared for a class in Theory 422-423 at Eastman School of Music, Rochester, New York, July, 1962.

ROOM USE ONLY

~~SEP 8 1968~~

~~JUL 20 1967~~

~~JUL 20 1967~~

~~JUL 20 1967~~

~~JUN 12 1971~~

~~JUL 17 1970~~

111V
88

MICHIGAN STATE UNIV. LIBRARIES



31293010590101