Preserving

Lejaren A. Hiller's Flowcharts:
A Case Study

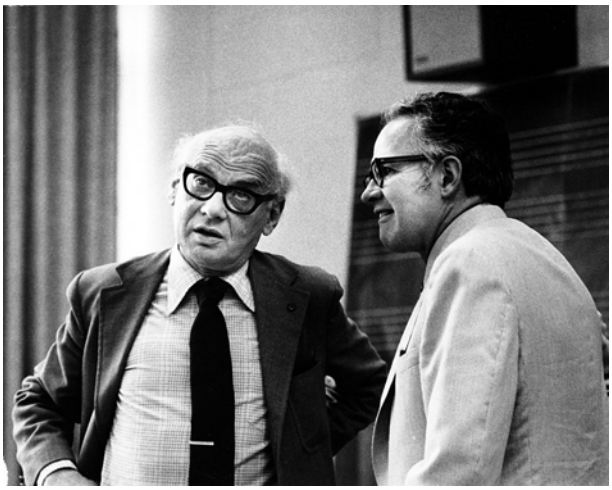
Lejaren A. Hiller
1924-1994

Co-curated by
Judy Jungels
&
John Bewley
with text and advisory support by
Professor Cort Lippe

January 2007

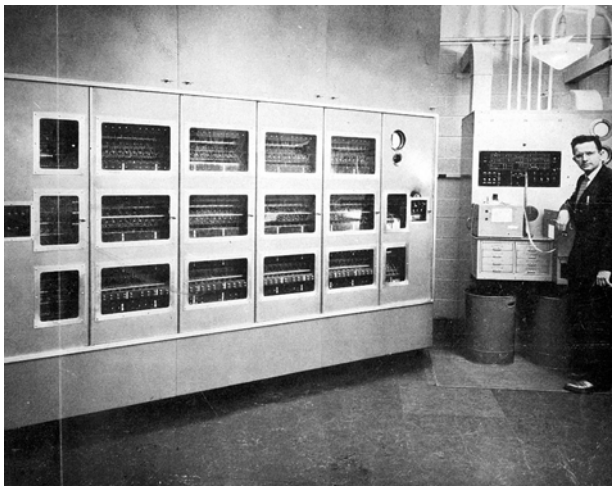
Case 1

Lejaren Arthur Hiller, Jr. was born in New York City in 1924. He started his musical training during his teenage years and continued them while a student at Princeton University where he received his undergraduate degree and Ph.D. in chemistry. His instructors at Princeton included Milton Babbitt and Roger Sessions.

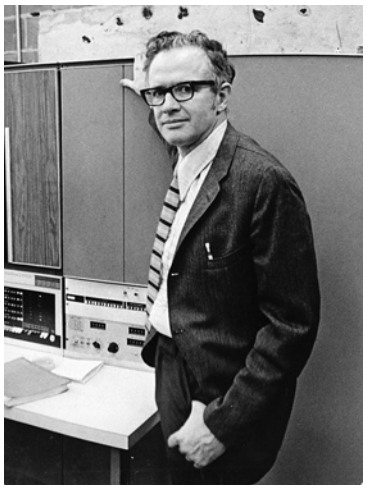


Milton Babbitt and Lejaren Hiller, ca. 1980
Photograph by Irene Haupt

He joined the faculty of the chemistry department at the University of Illinois in 1952. Part of his chemical research required Hiller to perform analyses on the **Illiac** computer at the University. This exposure and access to the computer led Hiller to musical experimentation that resulted in his *Illiac Suite*, composed 1955-57 with Leonard Isaacson, using the Illiac computer. The piece is recognized as being the **first significant computer music composition**.



Lejaren Hiller with Illiac computer,
University of Illinois, ca. 1956



Lejaren Hiller joined the Music Department faculty at the **University at Buffalo** in 1968. He served as Co-Director of the Center for the Creative and Performing Arts 1968-1974. Ill health forced Hiller to retire from the faculty in 1989. He was the author of three books, more than 80 articles on music, electronics, computer applications, and chemistry, and composer of more than 70 scores. Lejaren Hiller died January 26, 1994.

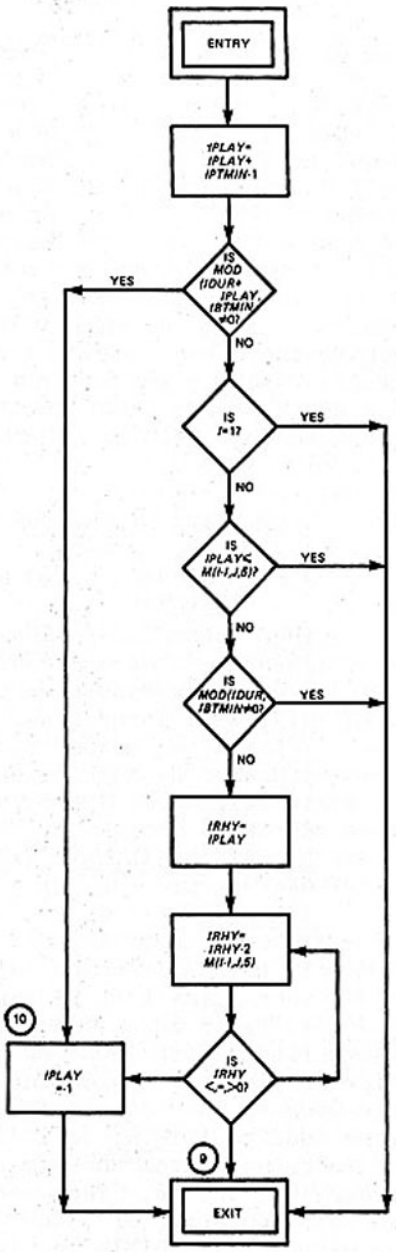
The **flowcharts** you see in this exhibit represent the work of one of the most important pioneers in the field of computer music. Lejaren Hiller's contributions to the field are some of the earliest and most far-reaching. His seminal book, ***Experimental Music*** (1959), describes his early work in the field of algorithmic composition. Hiller proposed breaking down the "rules" of composition to a series of instructions for a digital computer. This experimental approach to music composing anticipated the use of the computer in the creative arts.

[Realia—a copy of Hiller's text, *Experimental Music* is displayed.]

As Hiller states in the introduction to ***Experimental Music*** :

*"**such an undertaking immediately raises fundamental questions concerning the nature of musical communication and its relation to formal musical structures. Moreover, it also raises the question of how far it is possible to express musical and aesthetic principles in forms suitable for computer processing. Lastly, it also brings up the problem of what role automation of the type exemplified by high-speed digital computers can be expected to fulfill in the creative arts."***

As the computer becomes more and more intertwined with all of our daily activities, it is interesting to note that Lejaren Hiller posed these fundamental questions almost fifty years ago. Today, many of us are attempting to answer, in one way or another, these and similar questions. These flowcharts represent a significant effort in this pursuit.



FLOWCHART FOR
SUBROUTINE RHYTHM

Figure 5.12

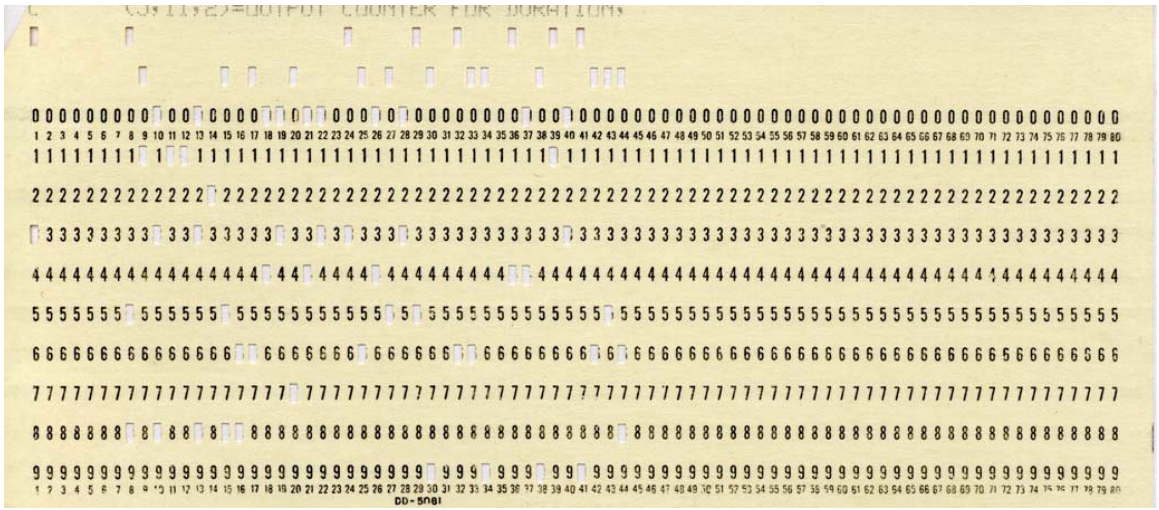
Example of "simple" Hiller flowchart for **Subroutine Rhythm.**
Fig. 5.12 in *Phrase Generation in Computer Music Composition*, Oct. 1978.

Hiller described the purpose of the Subroutine Rhythm as follows:

"What the present subroutine accomplishes is the elimination of a great deal of rhythmic complexity – essentially rhythmic randomness – that not only creates performance difficulties but also is aesthetically limited like any minimally organized structure."

CASE 2

Historically, flowcharts were used to represent the conditional logic of computer programs in what was called "electronic data processing" (better known as computer programming today). In the early days of the "electronic calculating machine", computers were enormous, often taking up entire floors of large buildings, and found only at government research centers, and soon thereafter on university campuses. Programmers punched holes in a set of **stiff manila cards** or onto **paper tape** at a mechanical teletype machine, then submitted the stack of cards to a computer operator, who in turn, read the cards into a reader connected to the computer and ran the programmer's job in a queue.

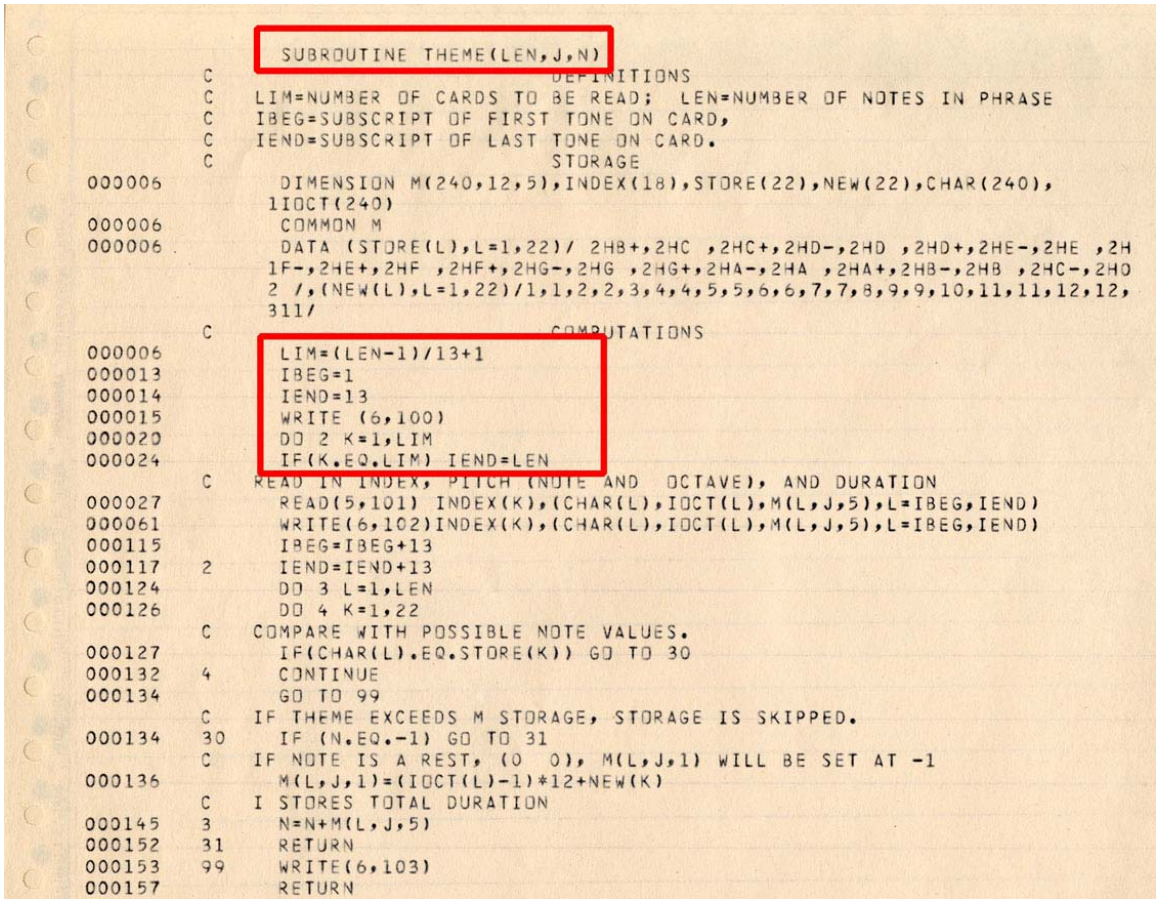


Programs were assigned a priority level; those requiring intensive calculations were queued to be run during the night when computer use was in less demand. Often a programmer came back the following day to pick up the printed results. Since the turnaround time was exceedingly slow, programmers naturally wanted to be sure that their programs were as error-free as possible and executed properly. The **flowchart** was a significant part of the programming procedure. Before the actual software was written, the programmer would sketch out a general scheme of the data flow with a set of visual symbols that represented the basic operations of the computer.

If a computer program can be described as a set of instructions for carrying out a particular task (an algorithm), then the flowchart was a graphical representation of an algorithm. Programmers used flowcharts to work out their ideas, and then pored over the final flowchart they created, making sure that their logic was correct, before writing the actual computer code. For many programmers, the final code was almost an afterthought: **the flowchart was where the bulk of the mental work took place**.

The relationship between a flowchart and a corresponding computer program can be seen in the following examples.

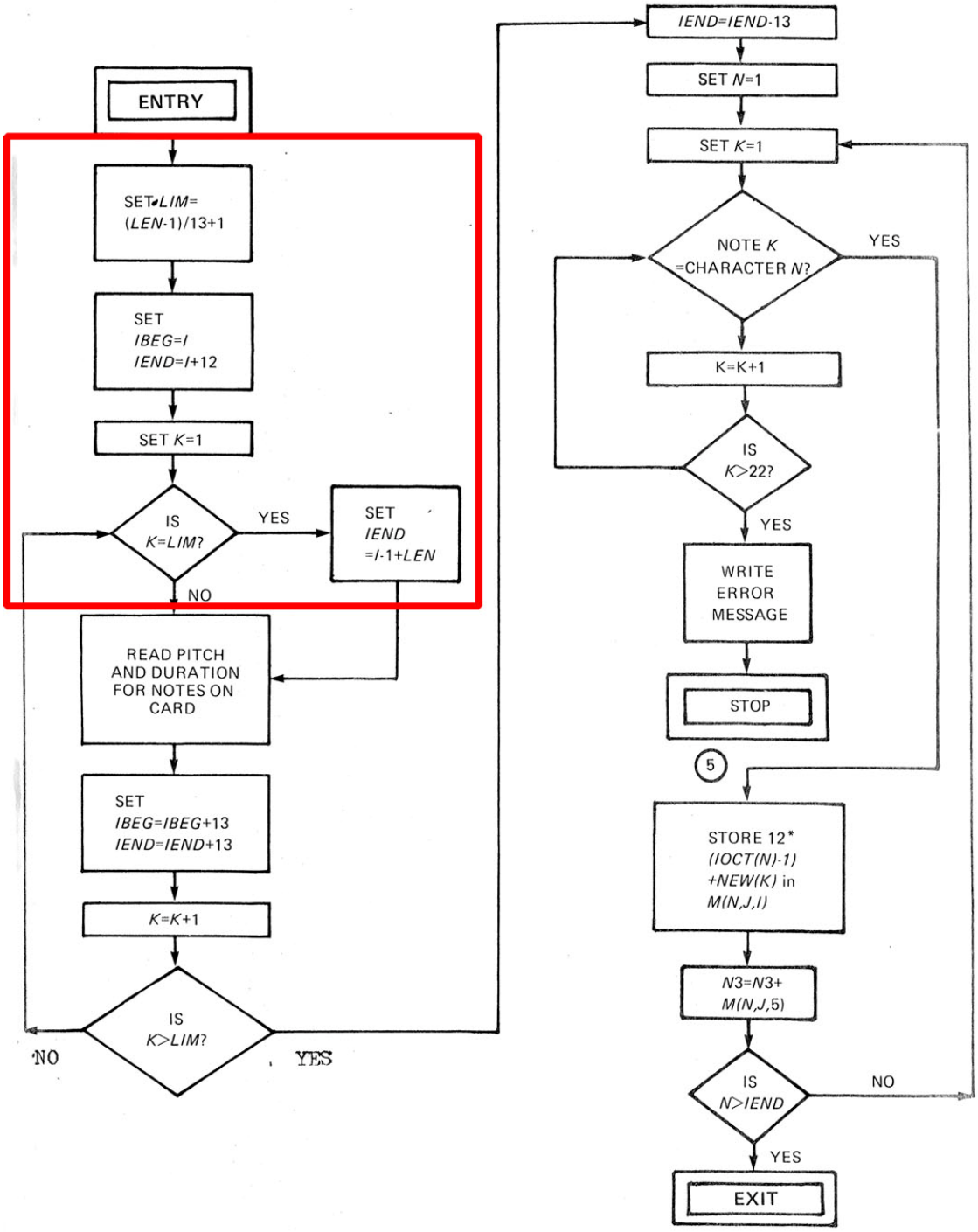
This computer printout was created by Lejaren Hiller October 23, 1976 for the second movement of his composition, ***Algorithms III.*** The printout refers to several subroutines, including one called "Theme". A close comparison of Hiller's flowchart for "Theme" and the computer printout reveals how the flowchart data would have been used to create the input for the computer program.



```
                      SUBROUTINE THEME(LEN,J,N)
            C                                 DEFINITIONS
            C     LIM=NUMBER OF CARDS TO BE READ;  LEN=NUMBER OF NOTES IN PHRASE
            C     IBEG=SUBSCRIPT OF FIRST TONE ON CARD,
            C     IEND=SUBSCRIPT OF LAST TONE ON CARD.
            C                                 STORAGE
000006                DIMENSION M(240,12,5),INDEX(18),STORE(22),NEW(22),CHAR(240),
                     1IOCT(240)
000006                COMMON M
000006                DATA (STORE(L),L=1,22)/ 2HB+,2HC ,2HC+,2HD-,2HD ,2HD+,2HE-,2HE ,2H
                     1F-,2HE+,2HF ,2HF+,2HG-,2HG ,2HG+,2HA-,2HA ,2HA+,2HB-,2HB ,2HC-,2HD
                     2 /,(NEW(L),L=1,22)/1,1,2,2,3,4,4,5,5,6,6,7,7,8,9,9,10,11,11,12,12,
                     311/
            C                                 COMPUTATIONS
000006                LIM=(LEN-1)/13+1
000013                IBEG=1
000014                IEND=13
000015                WRITE (6,100)
000020                DO 2 K=1,LIM
000024                IF(K.EQ.LIM) IEND=LEN
            C     READ IN INDEX, PITCH (NOTE AND  OCTAVE), AND DURATION
000027                READ(5,101) INDEX(K),(CHAR(L),IOCT(L),M(L,J,5),L=IBEG,IEND)
000061                WRITE(6,102)INDEX(K),(CHAR(L),IOCT(L),M(L,J,5),L=IBEG,IEND)
000115                IBEG=IBEG+13
000117      2         IEND=IEND+13
000124                DO 3 L=1,LEN
000126                DO 4 K=1,22
            C     COMPARE WITH POSSIBLE NOTE VALUES.
000127                IF(CHAR(L).EQ.STORE(K)) GO TO 30
000132      4         CONTINUE
000134                GO TO 99
            C     IF THEME EXCEEDS M STORAGE, STORAGE IS SKIPPED.
000134      30        IF (N.EQ.-1) GO TO 31
            C     IF NOTE IS A REST, (0  0), M(L,J,1) WILL BE SET AT -1
000136                M(L,J,1)=(IOCT(L)-1)*12+NEW(K)
            C     I STORES TOTAL DURATION
000145      3         N=N+M(L,J,5)
000152      31        RETURN
000153      99        WRITE(6,103)
000157                RETURN
```

Flowchart for Subroutine "**Theme**" from Lejaren Hiller's ***Algorithms III***, as published in his technical report, *Phrase Generation in Computer Music Generation*, October 1978.

Figure 5.3 FLOWCHART FOR SUBROUTINE "THEME"



Today, the traditional flowchart is no longer a step in computer programming, and has gone the way of the ancient mainframe computer, the punched cards, and the batch jobs queued by a computer operator. Programmers work in an entirely different fashion, using personal desktop machines, and compiling and running their programs immediately. The need for accuracy and error-free programming is no longer critical. Programmers work directly on the code, which looks more and more like a natural language. They draft a rapid prototype, then run the program until the compiler no longer gives errors and the program produces the desired results, all the while making small changes to the program. Programming has become a trial-and-error activity in a real-time feedback loop environment. A **flowchart** seems like an unnecessary, extra step in the process.

Fortunately, most of the flowcharts and diagrams produced by Lejaren Hiller were published as examples in his various texts and articles. These examples were collected and then matched to the remaining flowcharts. Comparison of the originals to the published examples allowed the conservation specialist to identify the missing instruction labels so they could be reattached to their original locations with a stable adhesive. If the **published examples** had not existed it would have been virtually impossible to identify the correct locations of all the scattered fragments.

[Realia – the display contains the original flowchart for "Program Dice Game" and the flowchart as published in Hiller's *Computer Programs Used to Produce the Composition HPSCHD*, 1972. Since the scanned images of the published flowchart and the original flowchart are essentially the same, only one version of the flowchart appears below. It is an excerpt from the center of the flowchart.]

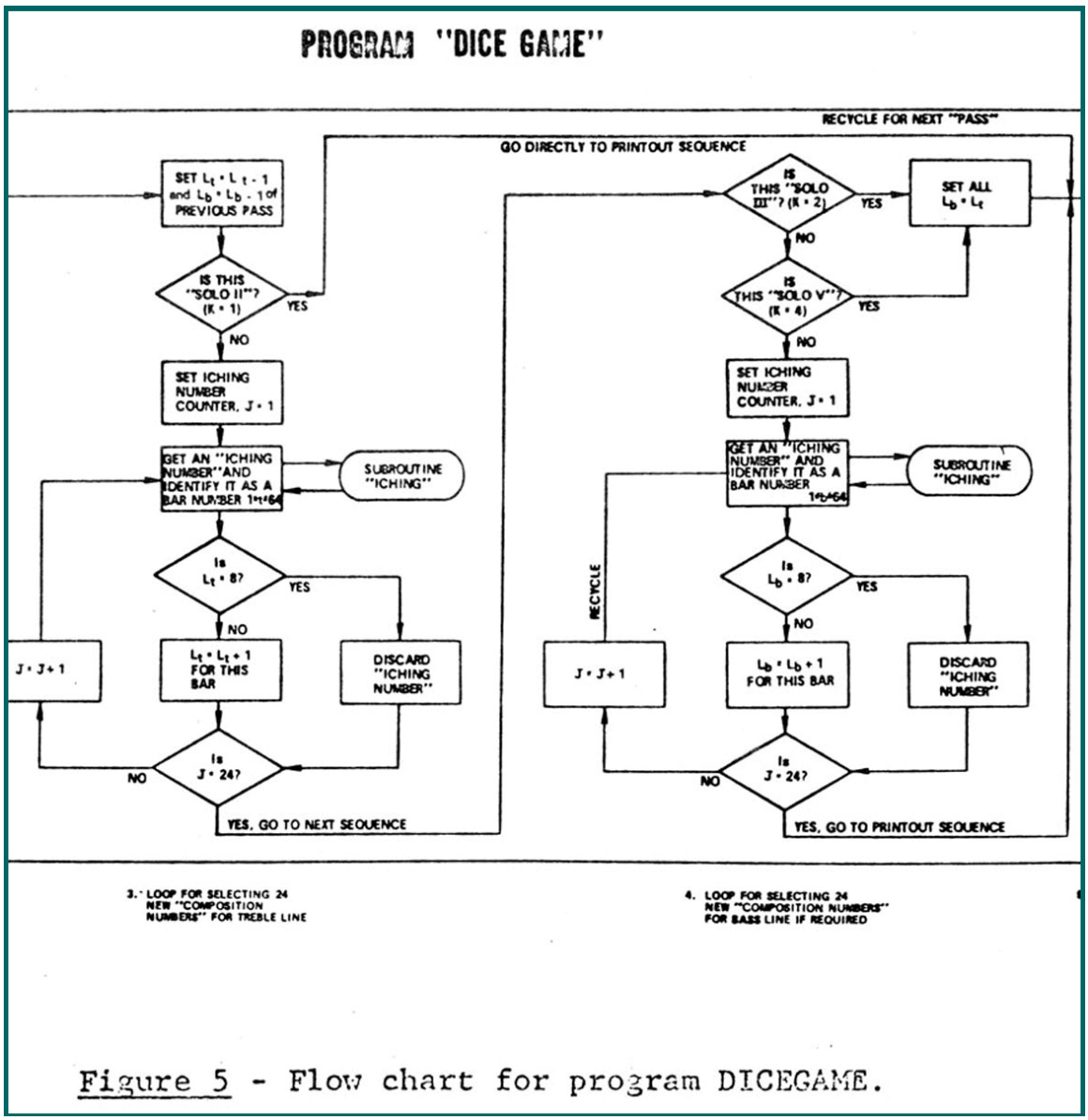


Figure 5 - Flow chart for program DICEGAME.

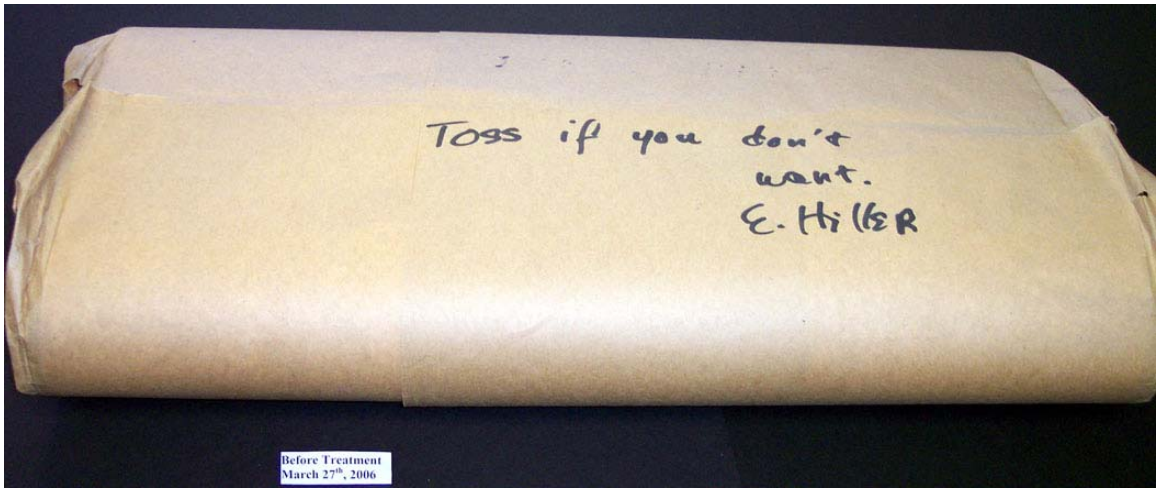
Wolfgang Amadeus Mozart reportedly (the authenticity is questioned in some sources) created his ***Musikalische Würfelspiele*** (Musical Dice Game) in 1787. He composed one-measure fragments that could be assembled to form a 64-measure minuet. The order of the fragments was determined by the roll of dice so that the final work was an example of what later became known as "**chance composition**".

John Cage and Lejaren Hiller computerized the **random process** of Mozart's musical dice game and used the results as a point of departure for five of the seven Soli performed by the harpsichordists in their multimedia work, ***HPSCHD.***

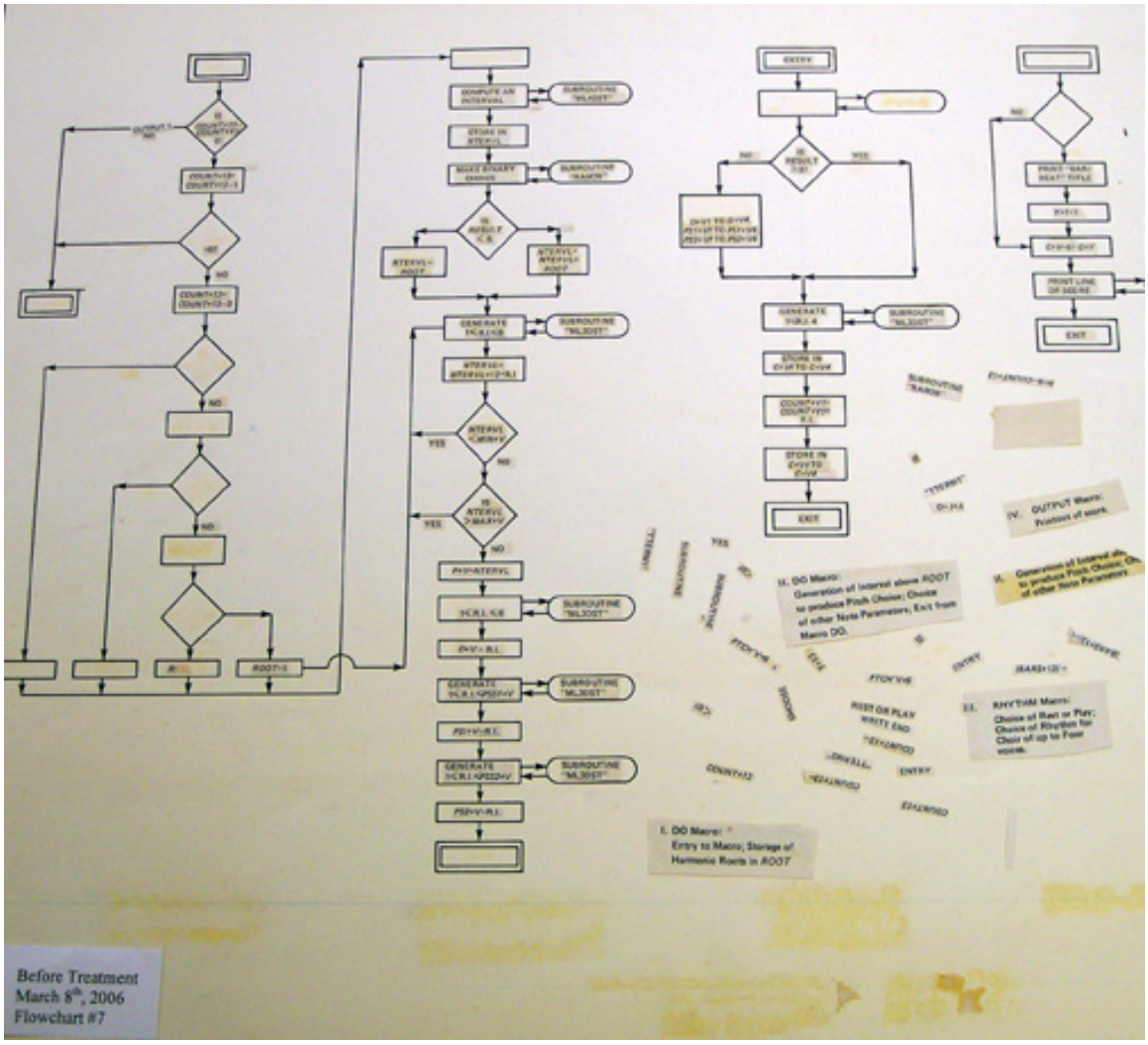
CASE 3

Elizabeth and Lejaren Hiller
Photograph by Irene Haupt

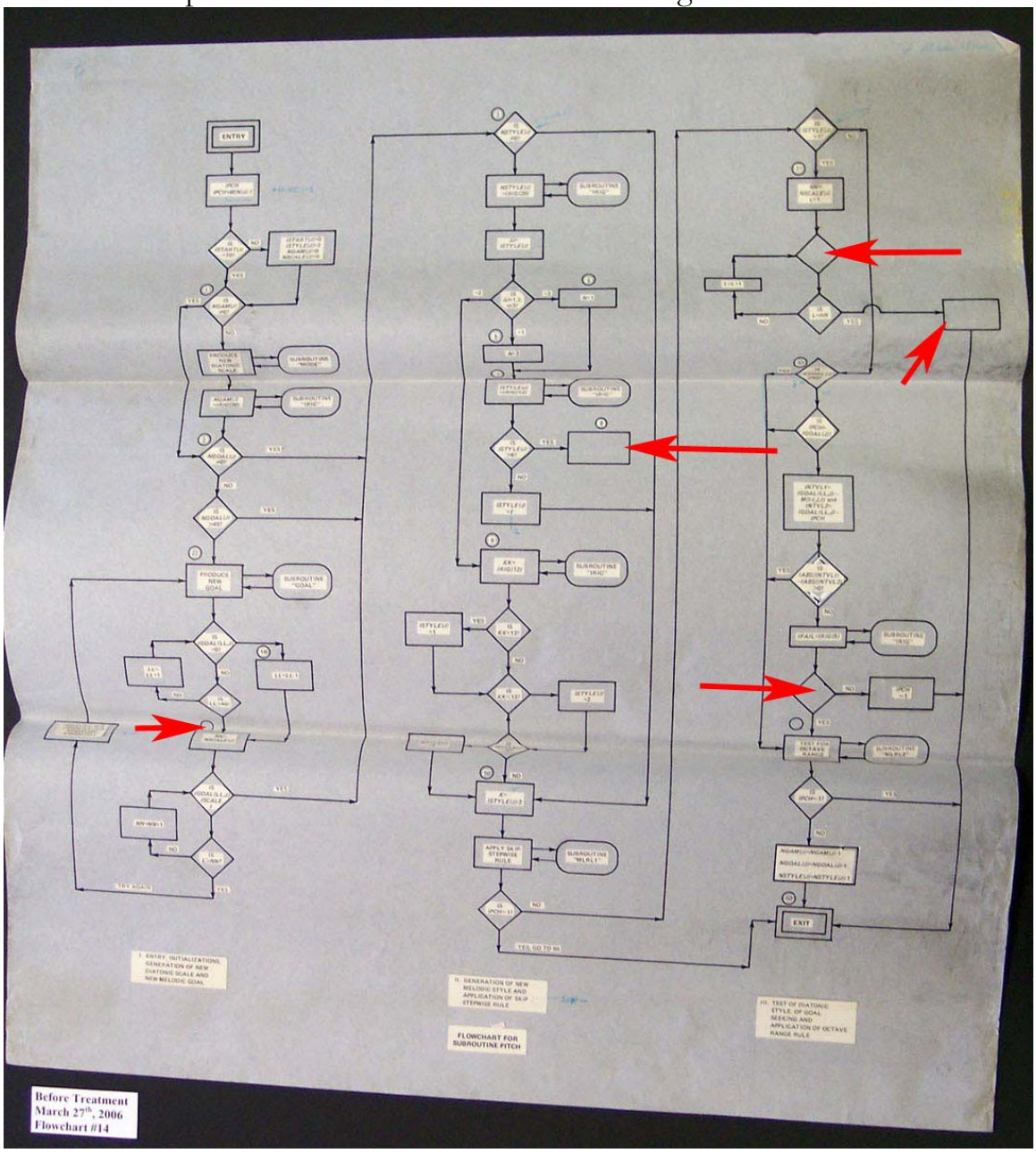
The Hiller flowcharts were donated to the Music Library by Lejaren Hiller's wife, **Elizabeth**. They were folded and rolled within brown wrapping paper in a manner that was surely meant to be temporary. However, as often happens, they were left to sit until a decision could be made about what to do with them. By the time they were rediscovered, they were in such a serious state of deterioration that many of the small **flowchart instructions** that had been glued onto the backing were no longer attached but were separated and lying loose on the backing, or worse, lost among all the other flowcharts. Unrolling the flowcharts or laying them flat caused many more of the fragments to **pop** loose as well, so the flowcharts could not even be inventoried. Conservation and preservation were **imperative** for this valuable material before even more damage occurred or the material was lost forever.



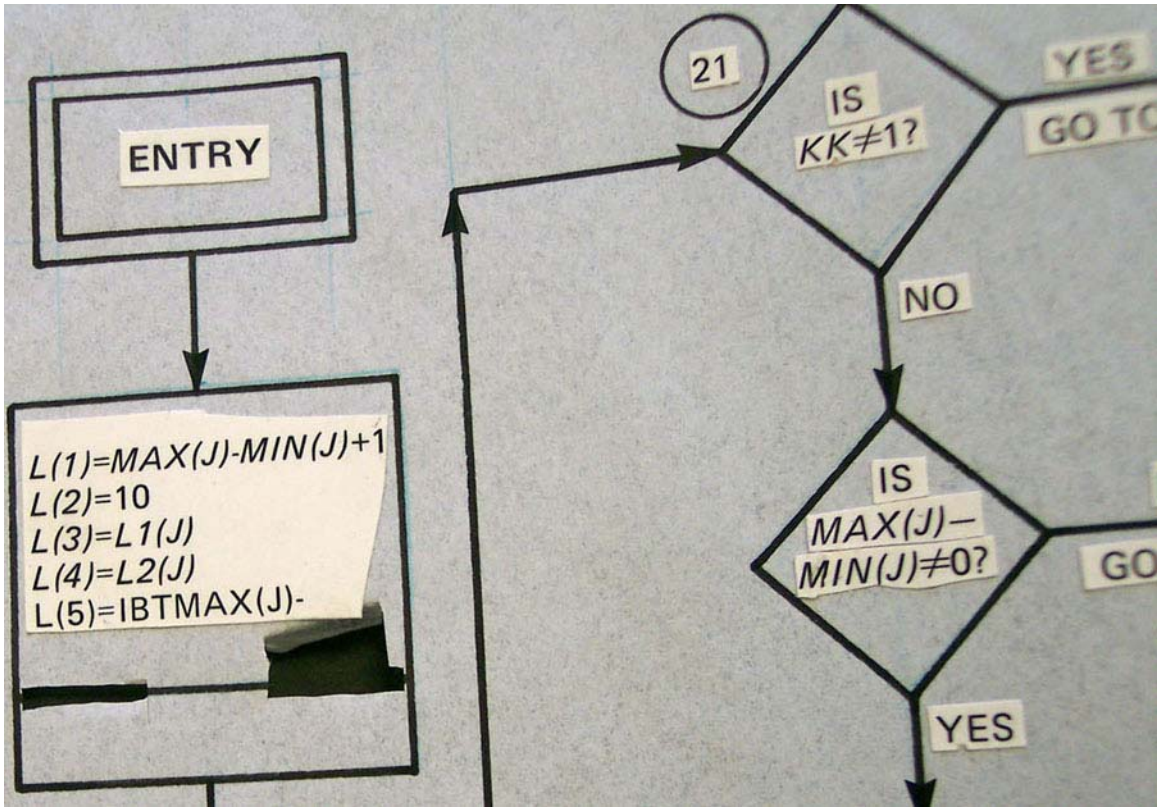
Wrapper in which the flowcharts were received, with Elizabeth Hiller's message.



Example of untreated flowchart with missing instruction labels.



Example of flowchart as it was removed from storage. Note the discoloration, effects of folding, and missing instruction labels (marked by red arrows).



Close-up view of instruction labels attached to vellum backing.

The **treatment** of the flowcharts required several steps:

1. **F**irst the surface was **cleaned** of dirt and grime using vulcanized rubber sponges and erasers.

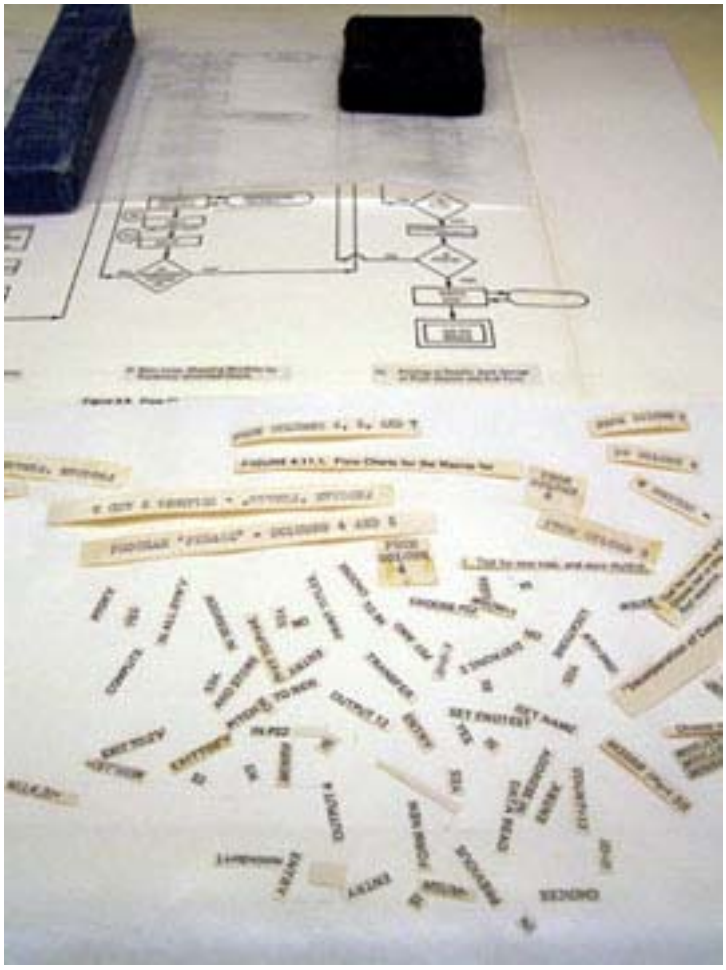
2. **T**he most heavily distorted flowcharts were then humidified and flattened under weights.

3. **A**ll of the flowcharts had to be cleaned of the old deteriorating adhesives, including pieces that were still attached with adhesive that would eventually fail over time.

4. **T**he correct location for the fragments was determined by comparison to published examples. The tiny pieces were then re-attached with a stable adhesive in the proper location.

5. The most prominent stains were reduced either manually or using a chemical treatment.

6. All the flowcharts were encapsulated in polyester Mylar for protection during handling and storage. The smaller flowcharts were also placed in a clamshell enclosure.

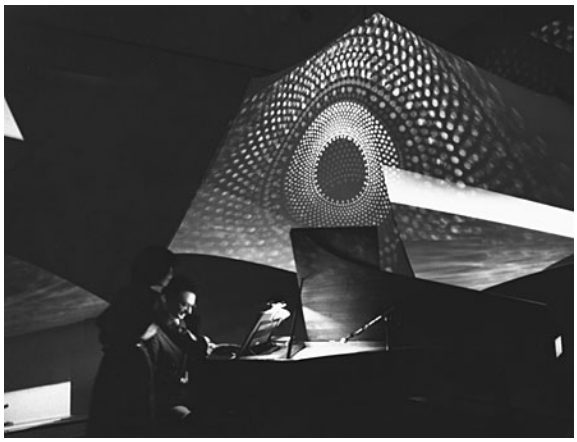


Example of flowchart during treatment: being flattened and labels being prepared to be re-attached.

Vertical cases

Lejaren Hiller and John Cage collaborated on the composition ***HPSCHD*** while Cage was in residence at the **University of Illinois** in 1967. The May 16, 1969 premiere performance of the huge multimedia work required 7 harpsichords, 208 tapes (4 copies of each of the 52), 52 tape-players (13

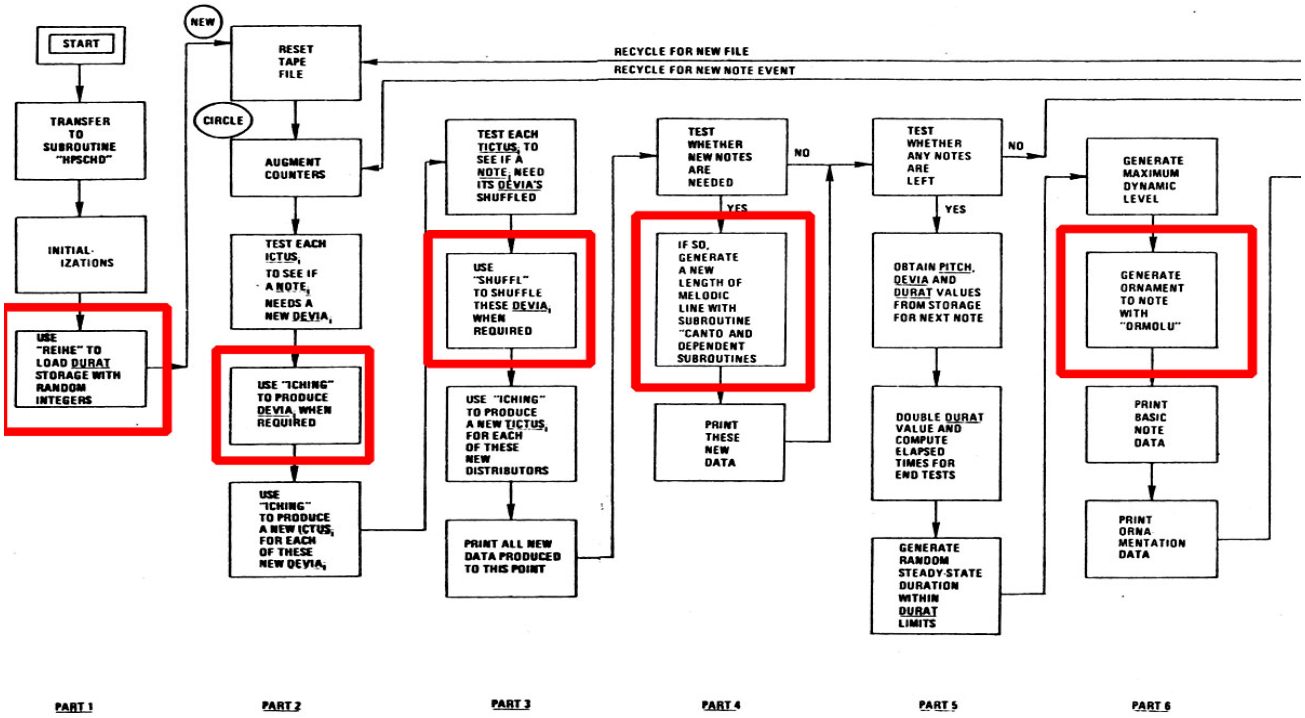
stations with 4 each), 59 amplifiers and loudspeakers, 6,400 slides (5,000 from NASA), 64 slide projectors, 40 films, 8 motion picture projectors, 11 100 x 40 foot silk screens and a 340 foot circumference circular screen made by Calvin Sumsion. It was attended by approximately 8,000 people and lasted nearly 5 hours.



Photographs by Irene Haupt of the 1980 performance of ***HPSCHD*** at the Albright-Knox Art Gallery, Buffalo, N.Y.

The following six flowcharts demonstrate how routines and subroutines were nested within the overall scheme for an entire work. The composition is ***HPSCHD***. The first example is an excerpt from the flowcharts for the simplified block diagram for the main program of ***HPSCHD***. It refers to the subroutines, Reihe, I-Ching, Shuffl, Canto, and Ormolu (marked by red boxes on the main program). Each of the subroutines is then represented by its own flowchart.

Simplified block diagram for main program of *HPSCHD*



Flowchart for subroutine "Reihe"

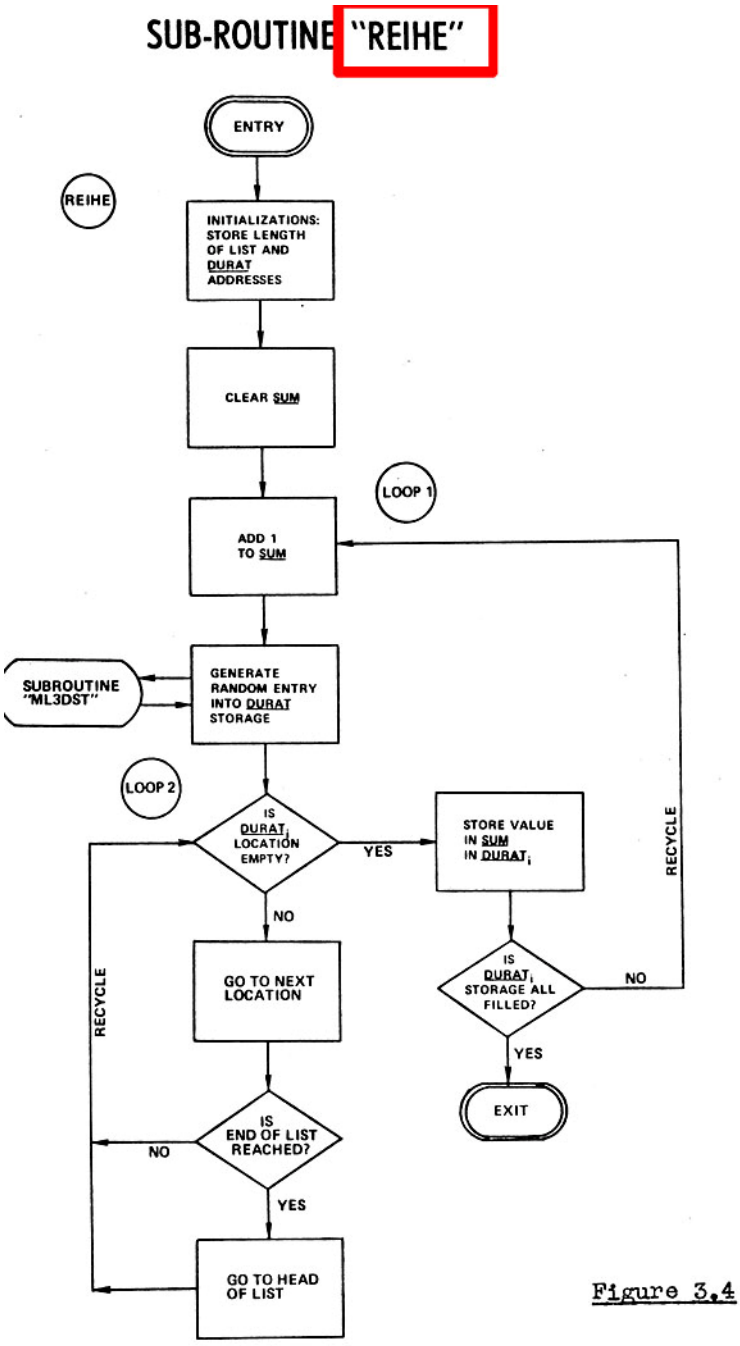
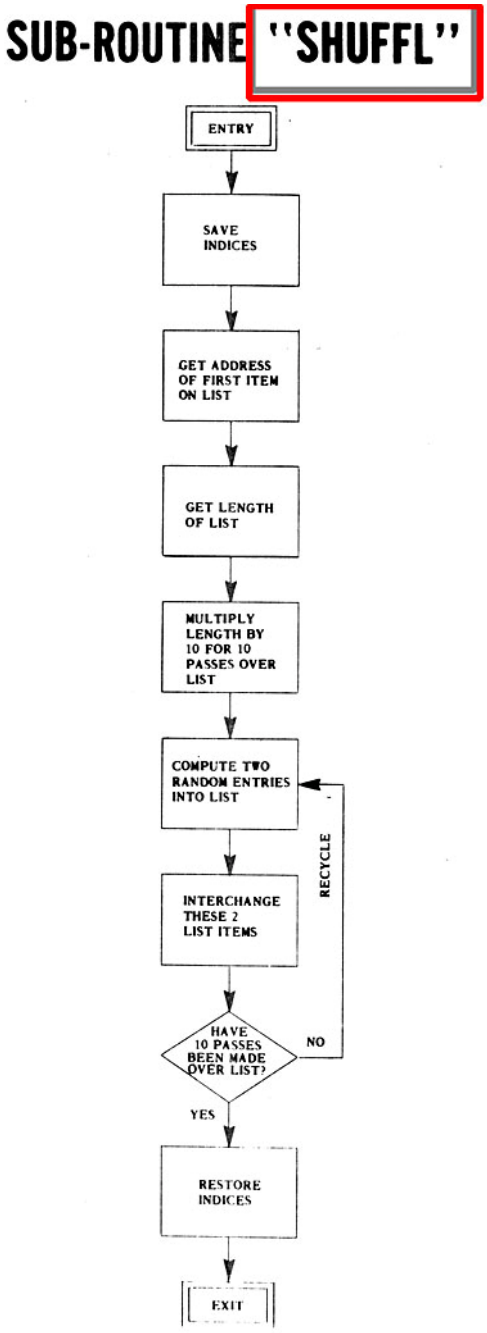


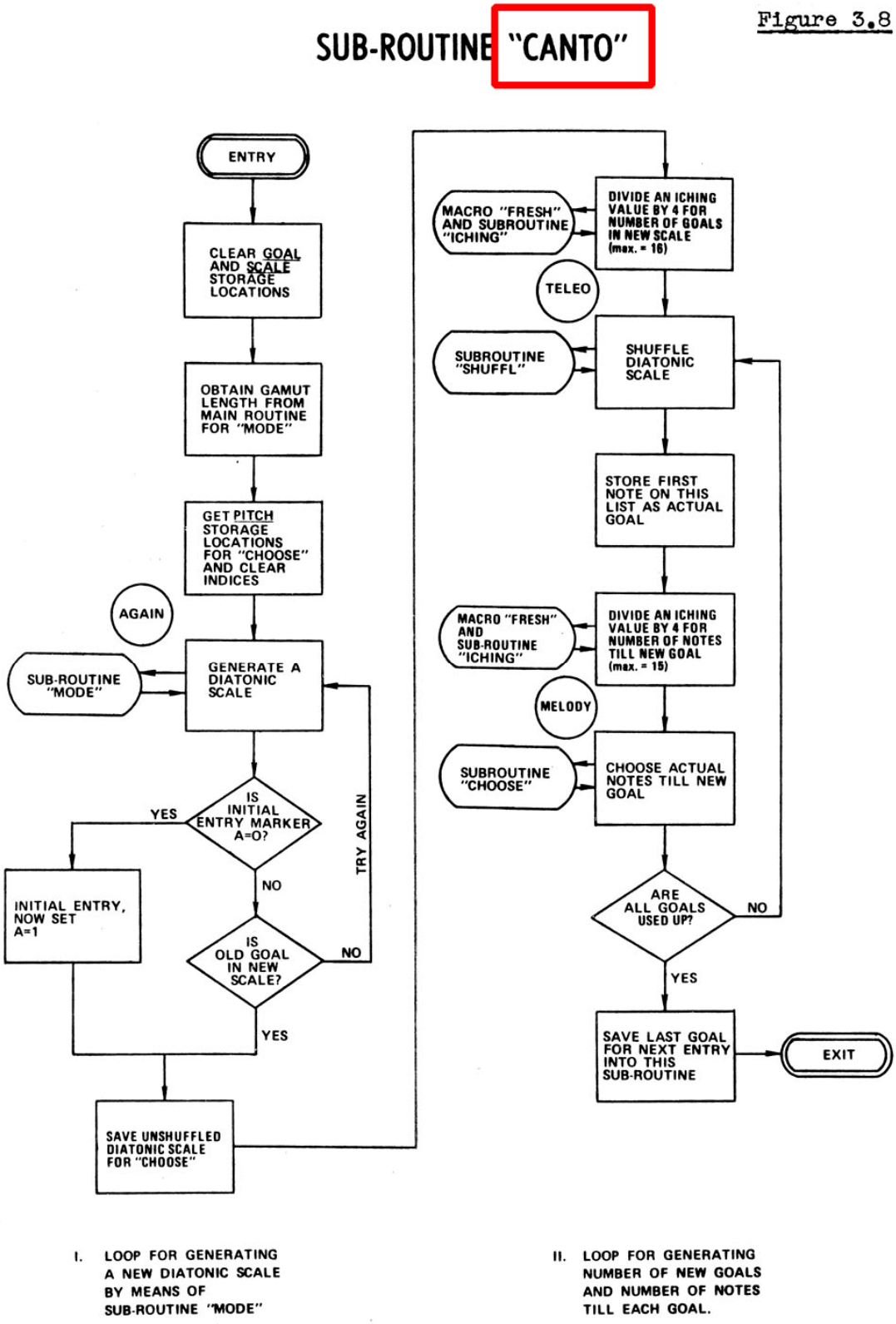
Figure 3.4

Flowchart for Subroutine "Shuffl"



Flowchart for Subroutine "Canto"

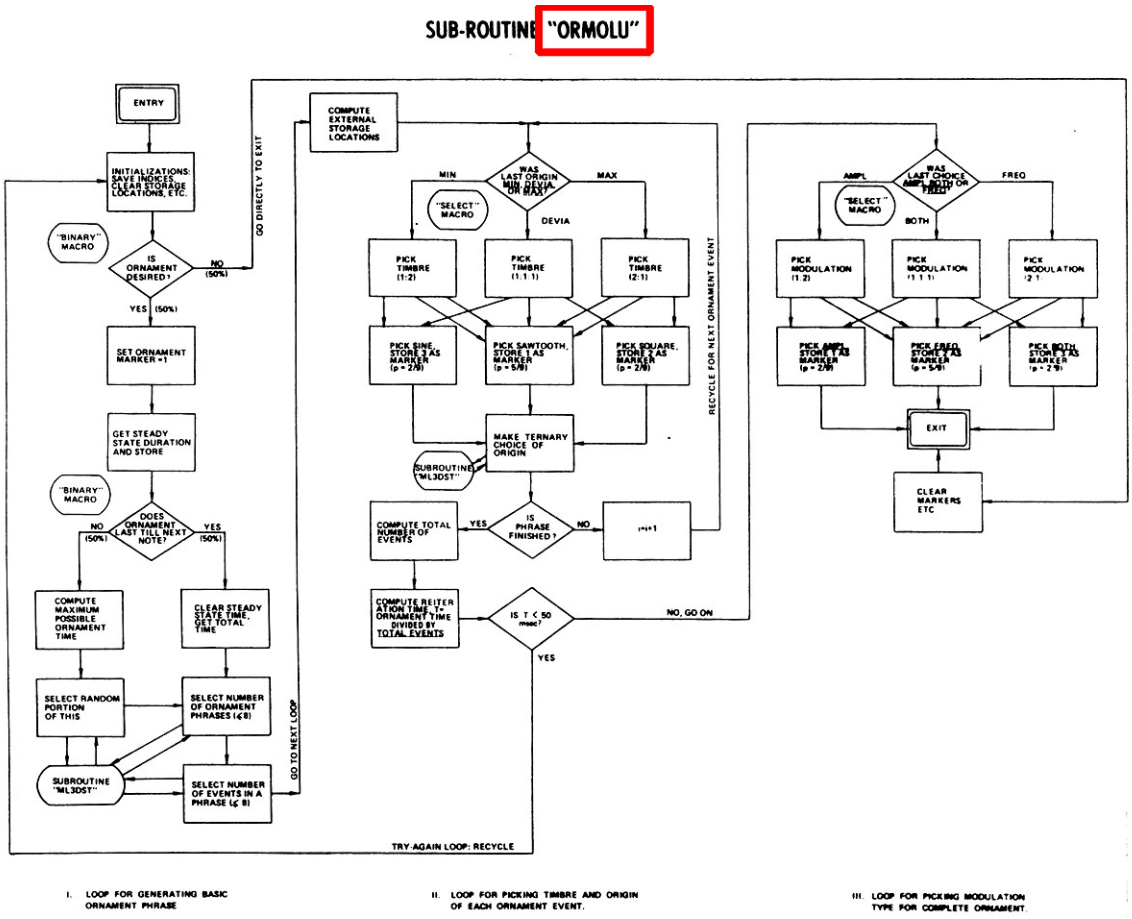
Figure 3.8



I. LOOP FOR GENERATING A NEW DIATONIC SCALE BY MEANS OF SUB-ROUTINE "MODE"

II. LOOP FOR GENERATING NUMBER OF NEW GOALS AND NUMBER OF NOTES TILL EACH GOAL.

Flowchart for Subroutine "Ormalu"



Now that the preservation of the **flowcharts** has been completed, they can be properly inventoried and stored so that they will be available for future research.