W.R. Weissenberger

~~325~~
BNWL 700

BNWL-506

AEC
RESEARCH
and
DEVELOPMENT
REPORT

ON-LINE COMPUTER STORAGE
AND RETRIEVAL OF PROCESSED
GAMMA SPECTRA DATA

L. D. Coffin

July 1967

BATTELLE-NORTHWEST

BATTELLE MEMORIAL INSTITUTE / PACIFIC NORTHWEST LABORATORY

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

operated by

BATTELLE MEMORIAL INSTITUTE

for the

UNITED STATES ATOMIC ENERGY COMMISSION UNDER CONTRACT AT(45-1)-1830

ON-LINE COMPUTER STORAGE AND RETRIEVAL
OF PROCESSED GAMMA SPECTRA DATA

By

L. D. Coffin

Mathematical Analysis Section
Mathematics Department

FIRST UNRESTRICTED
DISTRIBUTION MADE

AUG 11 '67

July 1967

ON-LINE COMPUTER STORAGE AND RETRIEVAL
OF PROCESSED GAMMA SPECTRA DATA

ABSTRACT

Included in this report are: (1) Descriptions of digital computer programs which allow the on-line storage of processed gamma spectra data and the retrieval of this data for on-line reports. (2) Program corrections, extensions, and clarifications pertaining to the on-line analysis of gamma spectra--described in the document, BNWL-CC-700, July 15, 1966.

TABLE OF CONTENTS

	<u>PAGE</u>
ABSTRACT	ii
I. INTRODUCTION	1
II. STORAGE TAPE INITIALIZATION (FIX).....	4
III. STORAGE ROUTINE (LINK 6).....	5
IV. RETRIEVAL PROGRAM (GET).....	7
V. PROTOTYPE REPORT PROGRAM.....	11
VI. CSE2 PROGRAM CORRECTIONS.....	13
REFERENCES	

LIST OF TABLES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
1	ON-LINE ANALYSIS PRINT OUT (CSE2).....	17
2	TAPE #3 BLOCK STORAGE ALLOCATION.....	18
3	BLOCK 5 AFTER EXECUTION OF FIX.....	19
4	TAPE STORED DATA ARRAY.....	20
5	TYPICAL REPORT INPUT AND OUTPUT.....	21
6	FIX PROGRAM LISTING.....	22
7	LINK 6 PROGRAM LISTING.....	26
8	REPORT PROGRAM LISTING.....	32
9	GET PROGRAM LISTING.....	35
10	CSE2 CORRECTED LISTINGS.....	47
11	PSEUDO FORTRAN FOR VROUT.....	76
12	LINK 5 LISTING.....	77

LIST OF FIGURES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
1	RETRIEVAL ("GET") FLOW DIAGRAM.....	83
APPENDIX A		
	REPORT PROGRAM CHANGES NEEDED AFTER COMPILATION.....	85A

I. INTRODUCTION

The Containment Systems Experiment program, undertaken by BNW in 1963, had two major objectives: (1) To measure the effectiveness and engineering safeguards in limiting the release of fission products from nuclear reactor accidents to the environment. (2) To investigate the aspects of a contained coolant loss accident, including the effects of a rupture in high temperature coolant piping and the leakage characteristics.

Following a radioactive aerosol release to the containment systems structure, numerous radioactive samples will be collected and analyzed for specific radionuclide content. The sample counting is accomplished by means of four multichannel analyzers interfaced with a small digital computer.⁽¹⁾ The analyzers operate through a data break into core memory. The digital computer is a PDP-7 with a 8,192 word core (18 bits/word). Interfaced with the PDP-7 are two KSR 33 teletypes, two Dectape dual transports, a Dectype 34 oscilloscope display unit and an IBM Selectric typer with an extra long carriage, which allows reporting on the standard 14 3/4 inch line printer paper.

The quantitative analysis of individual radionuclides for a given sample is accomplished immediately, by program, after a prescribed live count time has elapsed. Thus, the on-line program can operate on the stored raw data without any of the conventional tape inputs. The analysis uses a weighted least squares method which takes into account variation in the gross sample spectrum and counting instrument variability.

The on-line type out, following a sample analysis, includes the following information for an analyzer: (1) run number and sample identification. (2) Time sample was collected and the live time counting interval for the sample. (3) Nuclide, analyzer counts, relative standard error and disintegrations per minute (corrected to a reference time) for each nuclide--up to 8 nuclides allowed. (4) A chi square number, giving an indication of the model fit. (5) Degrees of freedom--equal to the number of channel groups less the number of isotopes being analyzed for in the sample.

This preliminary print out, (See Table 1) allows the experimenter to determine whether the current analysis is within reason and if not, he may make corrections to the analyzer instrumentation or erroneous input information (via the teletype), and reprocess the sample immediately. The experimenter will examine hundreds of sample print outs, extract the data, and sort or classify the data into various sub-sets for further analysis.

Sampling devices will collect fractional amounts of the total airborne and liquid phase content of the containment vessel as a function of time and deposition coupons will measure sorption of input activity on materials of construction. Samples of a given type will also be taken at numerous locations within the containment structure to insure valid interpretation of collected data. Approximately 1,000 individual samples will be processed for a single test and tests will be run about once a month. The possible different sample types and the associated six character sample identification can be illustrated by an example--using an airborne concentration sampler cluster, having 12 individual samplers with each sampler containing 4 aerosol fractionating components. The six character sample

identification is represented by,

$$M\alpha X\beta$$

where M represents airborne sampler, and

Sampler location = α = 01, 02,30

Time = β = 01, 02,12

Sampler component = X = A, B, C, D.

A typical six letter sample identification would be M15A07.

The experimenter may wish to examine the concentrations of a single isotope in micrograms/cubic meter of containment volume, at a fixed β time, for a given component and for all α locations. To perform this and other similar tasks efficiently, storage of all raw analytical data, for any test, is required and retrieval of that data is required for reprocessing and on-line reporting.

Accordingly, three computer programs were written. The first program, FIX, is an independent off-line routine which is used to initialize a master block of a magnetic storage tape--preceding a run or test. The second program is an extension of the automatic portion of the CSE2 code.⁽¹⁾ That portion consisted of several program links which grouped the channel count and background, generated the weighting factors, the group probabilities, carried out the matrix inversion and operated on the solution vector. LINK 6 was added and it allowed the sequential storage of each sample analysis on the magnetic tape after the print out. This addition necessitated refinements in the other links. The third program, GET, is a subroutine to retrieve the analytical data for various report generating codes.

The data storage and retrieval are, in general, adapted to the

characteristics of the PDP-7 magnetic tapes. Each tape has 576 addressable blocks and each block has 256 eighteen bit words. Tape read-write routines must search for a prescribed block and transfer data by one word at a time. A section of blocks on the storage tape will be reserved as an index "file" to the data storage, and another section will be used for the actual data storage. Each storage block will have four, 64 word sets--each set corresponding to one analysis or data array. It is impossible to compute a unique block number for all sample identifications--partly because of the 576 block limitation. Consequently, the data is stored sequentially and file block searching is done when reporting data. The tape block allocation is outlined in Table 2.

II. STORAGE TAPE INITIALIZATION (FIX)

FIX is used only once--preceding the sampling and analysis for a monthly run or test. The only input (via the teletype) is a run number which identifies a particular tape. FIX initializes the master block five by storing the run number in word one of a tape dialed in as unit three. (Hereafter, this storage tape will be called #3.) Word two of block 5 is set to zero. This word is the BD index which points to the current data storage block sub-section (4 sections) where the current data array is stored. Thus, for example, when the first analysis is stored, BD = 1 and the first 64 words of the first data block are filled. Word 3 is initially set equal to 36_8 . This is the BS index which points to the current data storage block. The fourth word, BF, is set equal to 6. BF is the index for the current file block. Any given file block has up to 85 three word sets which uniquely identify a sample analysis and

index the storage block and its sub-section. The fifth word, W, is the word pointer for the current three-word entry in the file block. See Table 3 for the master block layout.

III. STORAGE ROUTINE (LINK 6)

This program is called into core from tape 4 upon receipt of a "yes" command after LINK 5 has completed the analysis print out. (LINK 6 is not executed when a "no" command is received). Master block 5 is read and the run number is checked--if there is no match, an error message is written. Otherwise, the BD index is incremented and if $BD > 4$, the storage block index, BS, is incremented and BD is set to zero. A warning message is written when $BS > 539$. In a similar manner, the word index, W, is incremented by three and if > 255 , a new file block is needed--then BF is incremented, and if $BF > 29$, another error message is written. Otherwise, master block 5 is updated with all new indices after all data has been stored.

The current file block is read in and the two 18 bit sample identification words plus the current BD and BS indices are stored in the n, n + 1, and n + 2 locations of the current file block. The updated file block is written back on tape 3 only after all data has been successfully stored.¹ Note that one file block can reference 85 sets of analyses and with 24 file blocks, 2040 sets of data can be stored. Examination of blocks 5 and 6 after storage of the first data set and the second data set will illustrate the pattern. Suppose the run number is 999, the first sample identification is M11A12 and the second sample identification is M06B30.

¹If there is a tape error while writing data, the reference file must not be updated.

AFTER DATA SET 1

Block 5				Block 6			
9	9	9	RUN	M	1	1	
		1	BD	A	1	2	
	3	6	BS	1	3	6	(BD, BS) *
		6	BF	⋮			
		-1	W				
⋮							
⋮							

*BD is in bits 0-2, BS in bits 12-17

AFTER DATA SET 2

9	9	9	RUN	M	1	1	
		2	BD	A	1	2	
	3	6	BS	1	3	6	(BD, BS)
		6	BF	M	0	6	
		2	W	B	3	0	
				2	3	6	(BD, BS) *
				⋮			

At this point block 36 would be half full of data (2 sets x 64 words = 128).

Next, the proper storage block is read into the buffer and the program stores the data generated by LINK 4 and LINK 5 into the proper sub-section of the block and writes the updated block back on tape 3. Finally, the updated master and file block are written on tape. Examination of Table 4 shows the contents of the stored data for any one given sample analysis. The "program names" given are those actually used in the CSE2 code. The convention used in subscripting the arrays assumes an integer array--even though floating numbers are stored with the exponent and part of the fraction in word one and the rest of the fraction in word two. The IDW subscript in the data array denotes one of the four channel analyzer regions.

IV. RETRIEVAL SUBROUTINE (GET)

The GET subroutine, written in assembly language, is compatible with the PDP-7 Fortran operating system. The various Fortran report programs will use the GET subroutine for retrieval of stored analyses sets. The calling program supplies information to GET through the arguments of the call list and GET places the desired data in a special common area starting at octal 30. The argument list is as follows:

CALL GET (ID, M, IL, IT1, IT2, IM9, I1, I2, IDENT, ITOS, ID3)

ID = First three alpha-numeric characters of the six character sample identification.

M = Second three alpha-numeric characters of sample identification--only if IM9 = 0. If IM9 $\neq$ 0, M will take on values 1, 2,...12 and the subroutine will construct the second part of the sample identification using the value of M and another argument, IL.

IL = 1, 2,...12, indicating an A, B,...L. Again this interpretation is valid only if IM9 $\neq$ 0, otherwise, the argument IL is not used.

IT1 = first three alpha-numeric characters of the isotope identification.

IT2 = second three characters of the six character isotope identification.

IM9 = flag indicator for the output device--either the KSR-33 teletype (IM9 = 0) or the IBM Selectric (IM9 $\neq$ 0).

I1 = a GET return argument which is zero (I1 = 0) only if no data set is found on tape 3 for the input sample identification.

I2 = return argument which is zero (I2 = 0) if the input isotope is not found.

IDENT = flag indicator used by GET. If IDENT $\neq$ 0, GET initializes its search loop and sets IDENT = 0.

ITOS = Integer time of sample value. This is data retrieved from tape 3 (word 6 of the data array--Table 4) any may be used by the calling programs.

ID3 = return argument for deposition samples. If search is successful, ID3 = second three characters of sample identification.

On the formal report programs, using the IBM typer, the calling program will call for the second three characters of the sample identification from a nested DO loop: α = A, B,...L and β = 1, 2,...12. All these samples may not exist on tape. If a sample is found, ID3 = $\alpha\beta$ and if not ID3 = XXX (XXX is set in the calling program).

A typical input argument list (using the teletype) would be:

```
ID = M12
                        Sample identification M12C04
M = C04

IT1 =  I-
                        Isotope I-131
IT2 = 131
IM9 = IL = 0             indicates space or blank

IDENT = 1
```

If the IBM typer were being used,

```
M = 4

IL = 3      resulting in C04

IM9 > 0
```

The advantage gained by the calling program when it sets IM9 $\neq$ 0 can be illustrated by a typical nested DO loop in a report program.

```
B   ID = M12           /Boolean statement

DO  30  IL = 1, 4
    :
DO  20  M = 1, 12
    :
CALL GET (ID, M, IL,.....,ID3)

20 CONTINUE

30 CONTINUE
```

This will save the experimenter the chore of having to type in forty-eight (48) three character identifications each time he decides to change the last three characters of the sample identification.

Since the experimenter will be interested in reporting up to forty-eight sample concentrations for one isotope at a time and since the concentrations of eight isotopes are stored in the same data set (being in the same block and block sub-section on tape 3), a special procedure

was devised which cuts down the tape search time by a factor of eight. Thus, GET, can be called 48 times for isotope number one which necessitates 48 file block searches preceding the actual data block searches. As soon as isotope number two is requested for the same sample, another 48 searches would be required. Consequently, when IDENT $\neq$ 0, GET initializes sets IDENT = 0 and assumes the first of a series of isotopes. As long as the isotope identification (IT1, IT2) does not change, GET, will upon each entry, find the sample block and the sample block section and save these in two separate integer arrays--RS(J) and RD(J). When IT1 and IT2 change and as long as IDENT = 0, these two arrays are examined and all file block searching is by-passed. For example, suppose the following calling program DO loops are repeated for isotopes 1 and 2.

```
      DO 30  IL = 1, 2
          :
          :
      DO 20  M = 1, 1
          :
          :
      CALL GET (ID, M, IL,.....)

20 CONTINUE

30 CONTINUE
```

On the first time through both loops, suppose GET finds block 40 and section 3 (IL = 1, M = 1) and block 72 and section 4 (IL = 2, M = 1) then the two arrays will appear as,

```
      RS(1) = 40          RD(1) = 3
      RS(2) = 72          RD(2) = 4
```

on the second time through both loops (the isotope identification has

changed and IDENT = 0), the RS array is examined and the blocks are immediately read into core. The RD array entries are used to calculate the correct address in order to transfer the desired data. As a result, only one file block search, instead of two, is required. The DATA array (not to be confused with the DATA (IDW, J) array listed on Table 4) which GET returns is, at present, composed as follows:

DATA (1) = Disintegrations/minute for jth isotope

DATA (2) = Relative standard error ($\sigma/\bar{x}$)

DATA (3) = Not being used

DATA (4) = Pressure (if airborne sample)

DATA (5) = Rotometer reading (if airborne sample)

DATA (6) = Sample collection time

These values, plus report program inputs, are used to calculate isotopic concentrations in micrograms/cubic meter of containment volume for airborne samples and various other outputs for liquid and deposition samples. This DATA array is placed in a special common area, beginning at octal location 30. Because of the length of the Fortran report programs and since PDP-7 Fortran tape statements do not work, GET uses its own tape routines and a one block buffer area, beginning at 10400₈. For a description of preparing report programs to insure compatibility with the subroutine, GET, see Appendix A.

V. PROTOTYPE REPORT PROGRAM

This Fortran program was written specifically to test the retrieval program, GET, and to allow the experimenter to print out data for individual samples. The program is also useful if the experimenter merely wants to recall and examine a series of samples without waiting for a long

formal report. The program uses the teletype and not the IBM Selectric.

The following inputs are typed in by the operator:

- (1) IRUN - the three digit run number which must match that on tape #3.
- (2) NOI - number of isotopes
- (3) ID1(I) and ID2(I) for I = 1, NOI. These are respectively, the first and second three character isotope identifiers.
- (4) A(I) for I = 1, NOI - the specific activity of each isotope.
- (5) ISD1 and ISD2 - the six letter sample identification. For example, M05A02.
- (6) KFAC - the search factor associated with the sample identification as listed above. For example, if KFAC = 3, GET would be automatically called three times for one isotope and data would be supplied for type out information for the individual samples, M05A02, M05A03, M05A04. The maximum KFAC would be 12 if the sample identification began at one = say M05B01 or M06C01.

If liquid or deposition sample analysis are reported, the amount of the radionuclide is calculated. Thus, GET, returns the disintegrations/min = DATA (1) and Report does the following calculation:

$$\text{Micrograms} = A(J) \times \text{DATA (1)}$$

where $A(J) = \mu\text{g/d/m}$ (Specific activity for the j-th isotope)

If an airborne analysis is reported, the desired output is micrograms per cubic meter of containment vessel. This calculation is dependent upon

cubic meters of vessel, pressure corrections, flow meter readings and the collection time or the duration of sampling. The following calculation is performed.

$$\mu\text{g}/\text{meter}^3 = (x_1 x_2 x_3 x_4 P) (.008)$$

where,

$$x_1 = d/m = \text{DATA (1)}$$

$$x_2 = \mu\text{g}/d/m = A(I)$$

$$x_3 = \text{Rotometer reading} = \text{DATA (5)}$$

$$x_4 = \text{Collection time} = \text{DATA (6)}$$

$$P = \sqrt{P' + 14.7}$$

$$.008 = \text{Conversion constant}$$

$$P' = \text{DATA (4) if DATA (4) > 0 (lbs/in}^2\text{)}$$

$$P' = (.4909)(\text{DATA (4)}) \text{ if DATA (4) < 0 (inches Hg)}$$

Table 5 shows a typical input and output page from the teletype.

VI. CSE2 PROGRAM CORRECTIONS

Three basic changes, affecting several program links, were made. The first major revision was made to give the experimenter the option of asking for a variable lower limit on the first channel grouping-- thus allowing for the elimination of low energy channels. Secondly, the need for immediate storage on on-line analysis necessitated the complete revision of Link 4 and Link 5 and the addition of a Link 6 or storage program. Finally, the geometry factor, partially as a result of the first change, had to be computed, stored and used differently. Other small changes were made and can be noted by examining the up-dated program revisions for each of the basic changes.

VARIABLE LOWER LIMIT ON FIRST GROUP

- (1) VROUT portion of the main program CSE2 was rewritten to compute the normalized group probabilities. A pseudo Fortran listing for VROUT is given in Table 11. Now the programs use the normalization factor,
- $$\sum_{i=j}^k C_i,$$

where $1 \leq j \leq 45$ and $k = \text{LU}(\text{NG})$, the upper limit of the n -th group. See pages 57 and 58¹ and pages 48, 49, and 50 in this document for actual listing.

- (2) DG routine was rewritten to allow the experimenter to type in the lowest limit of group one and then the upper limits of groups 1, 2,...NG. See page 62¹ and pages 51, and 52 of this document.
- (3) An insertion was needed for the output heading in the DG routine and the three instruction dimension sequence for the lowest limit for all 4 regions. See page 65b and 66¹ and pages 54, 55, and 56, this document.
- (4) In Link 1 where the channel grouping is done for $B(i)$, the background rate and $Y(i)$, the observed total count for the i -th group, several changes were made so that,

$$B(1) = \sum_{i=\text{LLG}}^{\text{LU}(1)} C_i$$

where LLG may be >1 . The same applies to $Y(1)$. The same kind of change is inherent in the computation of the group probabilities denoted by $G(J,I)$. The same thing is done as in VROUT except that the sum of the standard channels, between the lower and upper limits, is calculated for isotopes 1 through j , where $1 \leq j \leq 8$,

and subsequently stored in a HT(j) array for later use in LINK 4. The old listing has changes on pages 94, 96, 98, 99, 100, 101, and 102, including LINK 2 changes.

STORAGE OF ON-LINE ANALYSIS

The storage of the number of disintegrations and the relative standard error, as well as other things associated with a given isotope, necessitated the re-arrangement and extension of common storage so LINK 5 and LINK 6 would have information generated LINK 4. The changes made to LINK 5 allow the experimenter to save the analysis on tape by typing in a simple "YES" command. The changes are so extensive to LINK 4 and LINK 5 that complete new listings are included in this document in Table 10, pages 70 to 75, and Table 12, pages 77 to 82.

GEOMFTRY FACTOR

In the old XLINK or standard link the geometry factor, G, was calculated for the j-th isotope, as follows:

$$G_j = C_j / D_j \exp (-.693 T/t)$$

$$\text{where } C_j = \sum_{i=1}^{256} (S_i / \text{Stime} - b_i / \text{Btime})$$

$$S_i = \text{Standard count}$$

$$b_i = \text{background count}$$

$$D_j = \text{disintegration/sec.}$$

$$t = \text{Half life.}$$

$$T = \text{Elapsed time between standard preparation and current count time.}$$

In the old, LINK 4, this factor was divided into the decay corrected estimated counts. Note that the C_j in the geometry factor weighted all 256 channels, even though the analysis may have considered a grouping, say, from 1 to 150.

In the current XLINK, the quantity, $G_j = D_j \exp(-.693T/t)$ is stored and C_j is not computed; only the spectrum is stored. The current LINK 1 calls in the stored standard spectrum (as well as the above quantity, G_j), calculates a C_j which is a sum between lower and upper markers, and saves for LINK 4. Total estimated disintegrations, for the j-th isotope are then computed (in Link 4) as:

$$\text{Disintegrations} = [A_{ij} G_j / C_j] \times \text{decay factor}$$

where A_{ij} = estimated counts.

Changes made to XLINK are on page 77 on the old CSE2 listing. The new listing is page 57, this document.

REFERENCES

1. N. P. Wilburn and L. D. Coffin. The Combination of On-Line Analysis With Collection of Multicomponent Spectra in a PDP-7, BNWL-CC-700, July 15, 1966.
2. Digital Equipment Corporation. PDP-7 Users Handbook (F-75).

TABLE 1

ON-LINE ANALYSIS PRINTOUT (CSE2)

A1
AUTOMATIC
SAMPLE ID =
M10A01
TYPE OF SAMPLE =
F1
SHELF NO. =
1

ANALYZER #. 1.

RUN NO.	SAMPLE ID	SAMPLE TIME (MIN)	LIVE TIME (MIN)
001	M10A01	1	1

NUCLIDE	COUNTS	REL STD ERROR	DISINTEGRATIONS
I-131	27711.	.024	.120707 06
RU103	23100.	.026	.139763 06
CS134	21469.	.033	.473743 05
BA140	36367.	.028	.193421 06

CHI**2 = -2.714

DEG. F. = 16

SAVE ANALYSIS, Y(YES), N(NO)

Y

TABLE 2

TAPE #3 BLOCK STORAGE ALLOCATION

<u>OCTAL BLOCKS</u>	<u>TOTAL BLOCKS</u>	<u>FUNCTION</u>
1 - 4	4	Never used
5	1	Master block
6 - 35	30 ₈	Storage Index File blocks
36 - 1033	776 ₈	Data Storage blocks

TABLE 3
BLOCK 5 AFTER EXECUTION OF FIX

<u>WORD</u>	<u>BLOCK 5</u>	<u>PROGRAM NAME</u>
1	1 1 0 7 0 1	IRUN (3 Character) (Example: IRUN = 971)
2	0 0 0 0 0 0	BD = 0
3	0 0 0 0 3 6	BS = 36
4	0 0 0 0 0 6	BF = 6
5	7 7 7 7 7 3	W = -4
	:	Words (6-256) never used
	:	
	:	
256		

TABLE 4

TAPE STORED DATA ARRAY			
BLOCK WORD	DESCRIPTION	PROGRAM NAME	
1	Sample ID (A3 FORMAT)	DATA (IDW,1)	
2	Sample ID (2nd part)	DATA (IDW,2)	
3	Aliquot / area ⁻¹ (Floating pt)	DATA (IDW,3)	
4	Aliquot / area ⁻¹ (Floating pt)	DATA (IDW,4)	
5	Rotometer #(I.R.)	DATA (IDW,5)	
6	Time of sample	DATA (IDW,6)	
7	Pressure (Floating pt.)	DATA (IDW,7)	
8	Pressure (Floating pt.)	DATA (IDW,8)	
9	Rotometer reading	DATA (IDW,9)	
10	Rotometer reading	DATA (IDW,10)	
11	Collection time	DATA (IDW,11)	
12	Collection time	DATA (IDW,12)	
13	Degrees of freedom	IG or SUB + 10	
14	χ^2 (Floating pt.)	CHI or SUB + 12	
15	χ^2 (Floating pt.)	CHI or SUB + 13	
16	Isotope #1 {	Isotope ident	IDENT (1)
17		Isotope ident	IDENT (2)
18		Calculated disintegrations (F.P.)	HT (1)
19		Calculated disintegrations (F.P.)	HT (2)
20		Relative std. error (F.P.)	HL (1)
21		Relative std. error (F.P.)	HL (2)
.	.	.	
.	.	.	
.	.	.	
.	.	.	
.	.	.	
58	Isotope #8 {	SAME AS ABOVE	
59			
60			
61			
62		SPARE	
63			
64			

*Assume Section 1 of 256 word block.

TABLE 5
TYPICAL REPORT INPUT AND OUTPUT

RUN NO

001

NO. OF ISOTOPES

2

6 LETTER ID AND SPEC. ACTIVITY

I-131		
.100E-06	ID= I-131	SPECIFIC ACT.= .999-07
CS134		
.100E-05	ID= CS134	SPECIFIC ACT.= .999-06

SAMPLE ID

M10A01

IF SEARCH, TYPE NUMBER FROM 1 TO 12

1

IRUN = 001 ISOTOPE = I-131

SAMPLE	D/M	MICRO GM/M**3	ER	CHI**2
--------	-----	---------------	----	--------

M10A01	.1207 06	.377-00	.020	-2.710
--------	----------	---------	------	--------

IRUN = 001 ISOTOPE = CS134

SAMPLE	D/M	MICRO GM/M**3	ER	CHI**2
--------	-----	---------------	----	--------

M10A01	.4737 05	.148 01	.033	-2.710
--------	----------	---------	------	--------

NEW ID TYPE 1, NO, TYPE 0

0

TABLE 6
FIX PROGRAM LISTING

XYZ	HLT	
100/	CAF	
HERE,	LAC N322	/R
	TLS	
	LAC N325	/U
	TSF	
	JMP --1	
	TLS	
	LAC N316	/N
	TSF	
	JMP --1	
	TLS	
	LAC N277	
	TSF	
	JMP --1	
	TLS	
	LAC N240	
	TSF	
	JMP --1	
	TLS	
	KSF	
	JMP --1	
	KRB	
	SAD N260	
	JMP A1	
	AND N17	
	JMP BT1	
A1,	LAC N20	
BT1,	DAC X	
	KSF	
	JMP --1	
	KRB	
	SAD N260	
	JMP A2	
	AND N17	
	JMP BT2	
A2,	LAC N20	
BT2,	DAC Y	
	KSF	
	JMP --1	
	KRB	
	SAD N260	
	JMP A3	
	AND N17	
	JMP BT3	
A3,	LAC N20	
BT3,	DAC Z	
	LAC X	

```

      CLL
      RTL
      RTL
      RTL
      TAD Y
      RTL
      RTL
      RTL
      TAD Z
      DAC 500
      CLA
      DAC 501
      LAC N36
      DAC 502
      LAC N6
      DAC 503
      LAM -3
      DAC 504
      CLA
      MMSE
      JMS DELA
      LAC CON
I.5,  MMSE
      MMLC
      MMEF
      JMP --1
      MMRS
      AND BIT3
      SZA
      JMP I.6
      LAW 60
      JMP I.5+1
I.6,  LAW 41
      MMLC
      JMS RED
      AND MSK777
      SAD BLOCK
      SKP
      JMP --4
/SET UP LOOP
      LAC CORE
      DAC 17
      LAM -377
      DAC COUNS
      CLC
      DAC CHECK
      LAW 43
      MMLC
      LAC I 17
      JMS WIT
      /WRITE FORWARD

```

B3,

```

        ADD CHECK
        DAC CHECK
        ISZ COUNTS
        JMP B3
/DEPOSIT LAST CHECK SUM
        MMBF
        JMP .-1
        CMA
        MMWR
/WRITE LAST CHECKSUM
        MMDF
        JMP .-1
        CLA
        MMLC
        CLC
        HLT
RED,    0
        MMEF
        SKP
        JMP RONG
        MMDF
        JMP .-4
        MMRD
        JMP I RED
WIT,    0
        MMEF
        SKP
        JMP RONG
        MMBF
        JMP DATA
        DAC TX
        LAC CHECK
        CMA
        MMWR
        CLC
        DAC CHECK
        LAC TX
DATA,   MMDF
        JMP WIT+1
        MMWR
        JMP I WIT
RONG,   LAC SEVEND
        HLT
DELA,   0
        LAC TC
        DAC TX
        ISZ TX
        JMP .-1
        JMP I DELA
/CONSTANTS
BIT3,   040000
```

TC, 754000
CON, 030060
BLOCK, 5
MSK777, 777
CORE, 477
COUNTS, 0
CHECK, 0
SEVEN0, 707070
TX, 0
V17, 17
V260, 260
V20, 20
V36, 36
V6, 6
V322, 322
V325, 325
V316, 316
V240, 240
V277, 277
X, 0
Y, 0
Z, 0
JMP HERE
START HERE

TABLE 7

LINK 6 LISTING

/LINK6, STORED ON TAPE 4 (BLOCKS 121- 124)
/STORES ON-LINE ANALYSIS

FIODEC
OCTAL
LINKER=305
IRUNN.=22
DATA.=144
A17=17
SUB=2605
ERROR=1422
IDW.=70
ISOC.=141
RITE=XCT LINKER+10
PUT=XCT LINKER+11
RUST=XCT LINKER+3
IBUFF.=150

1600/

IDENT, JMS CALSB
LAW 1
•IDDX

HT, JMS CALSB
LAW 2
•HHT

HL, JMS CALSB
LAW 2
•AAN

ASS, CALST
SET2W

JMS DELA
LAC TWORD
PUT
LAC IBUFF.+2
DAC L.
DAC A17
RUST
JMS DELA
LAC I L.
SAD IRUNN.+1

/READ UNIT 3, BLOCK 5

SKP
JMP .710

/WRONG TAPE ON3

DAC SUB
LAC I A17
DAC SUB+1 /BD - BLOCK SECTION
LAC I A17
DAC SUB+2 /BS - BLOCK STORAGE
LAC I A17
DAC SUB+3 /BF - BLOCK FILE
LAC I A17


```
DAC SUB+4    /W- FILE BLOCK WORD POINTER
ISZ SUB+1    /BD =BD+1
LAC SUB+1
DAC SUB0
SAD N5                      /IS IT TIME FOR NEW STORE BLK.
SKP
JMP .607                      /NO
/YES, RESET BD=1, LOCATE STORAGE POINTER TO SECTION 1 AND INCREMENT BS
    LAC N1
    DAC SUB0
    DAC SUB+1
    ISZ SUB+2
    LAC SUB+4
    TAD N3
    DAC SUB+4    /W=W+3
    DAC SUB3
    SAD N376    /IS IT TIME FOR NEW FILE BLOCK
    SKP
    JMP .610                      /NO
    LAM 0
    DAC SUB+4
    DAC SUB3
    ISZ SUB+3    /BF=BF+1
    LAC SUB+3
    SAD N35    /IS FILE ALMOST FULL
    JMS ER    /YES, WARN OPERATOR
/READ IN FILE BLOCK AND STORE FILE DATA
    LAC TBASIC
    TAD BIT0
    TAD SUB+3
    DAC TEM
    JMS INOUT
/NOW, CURRENT FILE BLOCK IN BUFFER AREA, STORE FILE
    LAC L.
    TAD SUB3    /WORD STORE INDEX-1
    DAC A17
    CAL DATA.
    ARX IDW.
    ARX N1
    LAC I TEMAD                      /GET SAMPLE ID(1)
    DAC I A17
    LAC TEMAD
    DAC NGG                      /SAVE ADDRESS
    ISZ TEMAD
    LAC I TEMAD                      /SAMPLE ID(2)
    DAC I A17
/NOW FIX BD IN BITS 0,1,2 AND BS IN BITS 8-17 FOR RETRIEVAL PROGRAM
    LAC SUB+1
    CLL
    RTL
```

```

RTL
RTL
RTL
RTL
RTL
RTL
RCL
TAD SUB+2    /ADD IN BS
DAC I A17
/WRITE OUT TO FILE BLOCK
LAC TEM
XOR BIT0
JMS INOUT
/NOW, READ IN STORAGE BLOCK, BS INTO Ibuff
LAC TBASIC
TAD BIT0
TAD SUB+2    /READ 3, BLOCK BS
DAC TEM
JMS INOUT
/FILL UP BUFFER WITH DATA ARRAY, FIRST FIX ADDRESS
/FOR SECTION 1, 2, 3 OR 4
.612,        LAM -0
TAD SUB0                      /BD-1
CLL
RTL
RTL
RTL
TAD L.
TAD .612                      /(BD-1)*64+L-1
DAC A17
LAM -11
DAC J.
KKK.,        LAC I NGG    /GET DATA(IDW,I)
DAC I A17
ISZ NGG
ISZ J.
JMP KKK.
LAC SUB+10    /DEGREES OF FREEDOM
DAC I A17
LAC SUB+12    /CHI (PART 1)
DAC I A17
LAC SUB+13    /PART 2
DAC I A17
CAL ISOC.
ARX IDW.
LAC I TEMAD                      /GET NO. OF ISOTOPES
CMA
TAD N1
DAC J.        /SET UP LOOP COUNT
CAL IDENT

```

```

        ARX N1
        LAC TEMAD
        DAC TEM+1    /ADDRESS OF ISOTOPE IDENT
        CAL HT
        ARX N1
        LAC TEMAD
        DAC TEM+2    /ADDRESS OF DISINTEGRATIONS
        CAL HL
        ARX N1
        LAC TEMAD
        DAC TEM+3    /ADDRESS OF ERROR
.620,   LAC I TEM+1
        DAC I A17    /STORE IDENT
        ISZ TEM+1
        LAC I TEM+1
        DAC I A17
        ISZ TEM+1
        LAC I TEM+2 /STORE DISINTEGRATIONS
        DAC I A17
        ISZ TEM+2
        LAC I TEM+2
        DAC I A17
        ISZ TEM+2
        LAC I TEM+3          /STORE ERROR
        DAC I A17
        ISZ TEM+3
        LAC I TEM+3
        DAC I A17
        ISZ TEM+3
        ISZ J.
        JMP .620
/END OF LOOP - DATA ARRAY MADE UP - STORE IT ON TAPE
        LAC TEM
        XOR BIT0
        JMS INOUT
/WRITE BLOCK 5 OUT
        LAC TWOR
        PUT
        LAC S2605
        RUST
        JMS DELA
.650,   WRITE    /PUT OUT DASHES
        RITE
        2
        FOR .FAA
        ENDIO
/RESET CHAIN TO BRING IN 5 BLOCKS OF MAIN
        LAC N3
        ADD LINKER
        DAC TEM

```

```

                LAM -4
                DAC I TEM
                JMP ERROR+1
.710,          WRITE      /WRONG TAPE
                RITE
                2
                FOR .772
                ENDIO
                JMP .650

```

/SUBROUTINES

```

INOUT,        0
                PUT
                LAC L.
                RUST
                JMS DELA
                JMP I INOUT

```

```

DELA,         0
                CLA
                MMLC
                JMS OLC
                CLA
                MMSE
                JMS OLC
                JMP I DELA

```

```

OLC,          0
                LAC TTC
                DAC OLC2
                ISZ OLC2
                JMP .-1
                JMP I OLC

```

```

OLC2,        0
/ERROR SUB.
ER,          0
                WRITE
                RITE
                2
                FOR .773
                ENDIO
                JMP I ER

```

/FORMATS

```

                OCTAL
.772,        320024
                TEXT

```

```

WRONG TAPE ON UNIT 3
                600000

```

```

.773,        320030
                TEXT

```

```

FILE STORAGE ALMOST FULL
                600000

```

• FAA, 320010
TEXT

* * * * *

600000

/ CONSTANTS

TWORD, 430005

L. 2 9

SUBO, 0

N5, 5

N 1, 1

N3, 3

SUB3, 0

N376, 376

N35, 35

TWOR, 030005

S2605, 2605

TBASIC, 030000

BIT0, 4000000

TEM, Ø

Q

⑧

9

9

NGG, Ø

J., 9

TTC, 754777

/LOWER COMMON

SUB/ .AAC, /G(8,20)

DECIMAL

• +320/ • AAB, /A(8,9)

•+144/ BTIM., /BKGD

• +2/ • GGS, /GS(8,17)

• +272/ • HHT, /HT(8), DISINTEGRATIONS

/UPPER COMMON

OCTAL

15675/ • AA0, /GG(8)

DECIMAL

.+16/ .AAN, /HL(8), ERROR

• +16/ • AAG, /B(20)

• +40/ • AAD, /Y(20)

• +40/ • AAH, /S(20)

•+40/ •IDDX, /IDENT

/IDENT ENDS AT 16145

START

TABLE 8

REPORT PROGRAM LISTING

```

/REPORT-A PROGRAM WHICH CALLS THE RETREIVAL PROG.-GET
2WORD
DIMENSIONID1(8),ID2(8),DATA(15),A(8)
C ASK FOR RUN NO. AND NO. OF ISOTOPES
4 WRITE 2,900
  READ 1, 906, IRUN
  WRITE2,935
  WRITE 2, 930
  READ 1, 902, NOI
  WRITE 2,904
  WRITE2,935
  DO 10 I=1,NOI
  READ1,906,ID1(I),ID2(I)
  READ1,907,A(I)
  WRITE2,950,ID1(I),ID2(I),A(I)
10 CONTINUE
C ID1 AND ID2 ARE 2A3 IDS, A IS SPECIFIC ACTIVITY
  WRITE2,935
20 WRITE 2,908
  READ 1,910,ISD1,ISD2
  DENT=5.
C ASK IF WE WANT A SEARCH ON ALL MAYPACKS
  WRITE 2,935
  WRITE 2,912
  READ 1,905,KFAC
C K=SEARCH FLAG, IF DESIRED.
C K=NO. OF SEARCHES
C SET UP BIG ISOTOPE LOOP
C SET IM9=0 TO INDICATE TO GET THAT TELETYPE IS BEING
C USED. IF IM9 NOT 0, GET, WILL USE IBM TYPER
  IM9= 0
  Y= .008067
  ID= ISD1
  DO 51 I=1,NOI
  M= ISD2
  IT1= ID1(I)
  IT2= ID2(I)
  WRITE2,914,IRUN,ID1(I),ID2(I)
  WRITE 2,916
  DO 30 L=1,KFAC
  IL=0
  CALL GET (ID,M,IL,IT1,IT2,IM9,I1,I2,ITOS,DENT)
  KK=I1*I2
  IF(KK)47,70,47
C OBTAIN MICRO GRAMS PER METER**3
C B=DISINT. * SPECIFIC ACTIVITY
47 CONTINUE
  R=DATA(1)
  RA=A(I)
  B=R*RA
  R8=B
  KT=ISD1
  OCTAL
  AND (770000
  SAD (430000
  JMP .62
  SAD (640000
  JMP .62
  S DECIMAL

```

```

C      GET CUBIC METERS FROM PRESSURE, ROTD
C      READING AND COLLECTION TIME (DURATION
C      TIME)
      IF(DATA(4))50,60,60
50     DATA(4)=DATA(4)*.4909+14.7
      GO TO 61
60     DATA(4)=DATA(4)+14.7
61     DATA4=DATA(4)
      R5=SQRTF(DATA4)
      R6=DATA(5)*DATA(6)*R5*Y
      R8=B/R6
62     DATA1=DATA(1)
      DATA2=DATA(2)
      DATA7=DATA(7)
      WRITE 2,920,1D,M,DATA1,R8,DATA2,DATA7
920    FORMAT(2A3,3X,E8.4,6X,E8.3,4X,F6.3,3X,F6.3/)
70     CONTINUE
      M=M+1
S      OCTAL
S      AND (77
S      DAC SPAR
S      SAD (12
S      SKP
S      JMP .+10
S      LAC (000120
S      DAC SPAR
S      LAC M.
S      AND (770000
S      TAD SPAR
S      DAC M.
S      JMP .30
S      SAD (21
S      SKP
S      JMP .+3
S      LAC (000101
S      JMP .-12
S      LAC M.
S      AND (777700
S      TAD SPAR
S      DAC M.
S      DECIMAL
30     CONTINUE
51     CONTINUE
      WRITE2,922
      READ1,902,1G
      IF(1G)20,550,20
550    CONTINUE
S      HLT
600    GO TO 4
900    FORMAT(6HRUN NO//)
902    FORMAT(2I3)
904    FORMAT(30H1 LETTER ID AND SPEC. ACTIVITY//)
905    FORMAT(15)
906    FORMAT(2A3)

```

```
907      FORMAT(E8.3)
908      FORMAT(9HSAMPLE ID//)
909      FORMAT(I5)
910      FORMAT(2A3//)
912      FORMAT(34HIF SEARCH,TYPE NUMBER FROM 1 TO 12//)
914      FORMAT (7HIRUN = A3,10X,10HISOTOPE = 2A3//)
916      FORMAT(6HSAMPLE,6X,3HD/M,7X,13HMICRO GM/M**3,4X,
$        2HER,4X,6HCHI**2//)
922      FORMAT(26HNEW ID  TYPE 1, NO, TYPE 0)
930      FORMAT (15HNO. OF ISOTOPES//)
935      FORMAT(1H ///)
950      FORMAT(10X,4HID= ,2A3,16H SPECIFIC ACT.= ,E8.3)
955      FORMAT(8E8.4)
954      FORMAT(57HDATA1  DATA2  DATA3  DATA4  DATA5
$        DATA6  DATA7  B)
980      FORMAT(3E8.3)
      END
```


TABLE 9

GET PROGRAM LISTING

```

/MAYSUB = TAPE RETRIEVAL SUBROUTINE
/CALL ARGUMENTS ARE.
/IDENT NOT 0 RESULTS IN GET INITIALIZATION - SWITCH A SET
/ID = FIRST THREE ALPH-NUMERIC CHARACTERS FOR DATA IDENTIFICATION
/IL = 1 TO 12 INDICATING AN A THROUGH L WHICH FORMS THE FIRST
/CHARACTER OF 2ND DATA IDENT. WORD
/M = 1 - 12 (2ND AND 3RD CHARACTER OF 2ND DATA WORD)
/IT1 = ALPH-NUMERIC ISOTOPE ID
/IT2 = 2ND PORTION OF ISOTOPE ID
/IM9=0 -TELETYPE REPORT, IM9 NOT 0, IBM TYPER REPORT
/RETURN ARGUMENTS
/ID3 = SECOND DATA IDENTIFICATION WORD ( GENERATED BY GET )
/I2 = 0 NO ISOTOPE FOUND
/I1 = 0 NO DATA FOUND
/DATA ARRAY IN COMMON FROM OCTAL 30 THRU 46
/ITOS = FIXED POINT TIME OF SAMPLE
/ VEC CONTAINS JMS .I02 OR JMS .I09( THE LATTER FOR IBM)
/

```

FIODEC

```

/PARAMETER ASSIGNMENTS
EXTERNAL .I02

```

BAR 2

```

OCTAL
.DDD=30
IRUN=26
.BFF=10400
.RDD=17100
.RSS=17160
DECIMAL
INTERNAL GET.

```

```

GET., JMS GTARG
      JMP .AAA

```

```

ID., 0
M., 0
IL., 0
IT1., 0
IT2., 0
IM9., 0
I1., 0
I2., 0

```

```

ITOS., 0 /TIME OF SAMPLE (INTEGER)
IDENT., 0

```

```

ID3., 0

```

```

.AAA., JMP M10
BUF., JMS CALSB

```

LAW 1

.BFF

```

DATA., JMS CALSB

```

LAW 2

.DDD

```

RD., JMS CALSB

```

LAW 1

.RDD

```

RS., JMS CALSB

```

```

LAW 1
.RSS
M10, NOP
LFM
CLA /DESELECT ALL TAPES
MMLC
MMSE
JMS OLC
LAC I IDENT /SEE IF ID HAS CHANGED
SNA
JMP M11 /NO
LAW A /YES, RESET SWITCH
DAC VAR
M11, JMP I VAR /GO TO A,B,C
A, LAC U35
JMS MTAPE /READ IN BLOCK 5
LAC BUF+2 /ARGUMENT FOR MTAPE (NOT EXECUTED)
LAC .BFF
SAD IRUN /HAVE WE THE PROPER TAPE
JMP M20 /YES
/NO, INFORM THE OPERATOR
WRITE
JMS .IO2
2
FOR .800
ENDIO HLT JMP A
/OK, GO AHEAD
M20, LAC .BFF+3 /GET FILE BLOCK FROM MASTER TAPE
CMA
DAC BF / -BF ( USED IN SEARCH)
LAW B
DAC VAR /SET NEXT ENTRY
LAC (1
DAC J
LAC I IT1
DAC IS11 /SAVE ISOTOPE ID PART 1
LAC I IT2
DAC IS22 /SAVE PART 2
DZM I IDENT
/FIX 2-ND PART OF ID AS A FORMAT IF IBM TYPER IS BEING USED
A2, LAC I IM9
SNA
JMP A3.20 /TELETYPE BEING USED
LAC I IL /IL = 1,2,3 .....12
SAD (10
JMP A2.8
SAD (11
JMP A2.8
SAD (12
JMP A2.8
TAD N60
JMP A2.9
A2.8, TAD N27
A2.9, CLL
JMS ALS12 /SHIFT LEFT 12
DAC TEM
LAC I M /M= 1,2,3, ..... 12

```

```

AND N70
SZA
JMP A3.5
/M = 7 OR LESS
A3.1, LAC BIT7
      TAD I M
A3.2, TAD TEM
A3.3, DAC I ID3
      LAC (6
      DAC FB      /SET FILE BLOCK = 6
      JMP A4
A3.5, LAC I M      /M= 8,9,10,11,12
      SAD (8
      JMP A3.1
      SAD (9
      JMP A3.1
      SAD (10      LAC (80      SAD (11
      LAC (65      SAD (12
      LAC (66
      JMP A3.2
/TELETYPE BEING USED
A3.20, LAC I M      /ID ALREADY IN A FORMAT
      JMP A3.3
/READ INTO BUFFER - THE CURRENT FILE BLOCK
A4, TAD U3      /U3 = 030000
      JMS MTAPE
      LAC BUF+2      /ADDRESS OF BUF
      LAC (1
      DAC K      /K=1
A5, CAL BUF
      ARX K
      LAC I TEMAD
      ISZ TEMAD
      SAD I ID      /IS BUF(K) = ID
      SKP      /YES
      JMP A7      /NO
      LAC I TEMAD
      DAC ID2
      LAC I ID3
      SAD ID2      /IS BUF(K+1)=ID2
      JMP P      /YES

```

A7, LAC K
 TAD C3
 DAC K
 SAD C256 /THRU WITH ONE BLOCK
 SKP
 JMP A5 /KEEP SEARCHING

A10, LAC FB
 TAD C1 /FB=FB+1
 DAC FB
 TAD BF /FB-BF
 SMA
 JMP A12 /NEVER FOUND
 LAC FB
 SAD C30
 SKP /NEVER FOUND
 JMP A4 /CONTINUE SEARCH

A12, DZM I 12
 DZM I 11
 RETUR
 HLT

/SAVE BLOCK SECTION, BD AND STORAGE BLOCK, BS
 /IN RD AND RS ARRAYS, RESPECTIVELY

P, ISZ TEMAD
 LAC I TEMAD
 DAC TEM

/TEM CONTAINS BD IN BITS 0, 1, 2 AND BS IN BITS 6 - 17, RIGHT
 /ADJUSTED

RTL
 RTL
 AND C7
 DAC TEM+1 /BD
 LAC TEM
 OCTAL
 AND C7777
 DAC TEM /BS
 DECIMA

CAL RD /SAVE BD IN RD(J) ARRAY
 ARX J

LAC TEM+1
DAC I TEMAD

CAL RS /SAVE BS IN RS ARRAY
ARX J
LAC TEM
DAC I TEMAD

/READ IN BLOCK RS(J)

P1, CAL RS
ARX J
LAC I TEMAD
TAD U3
JMS MTAPE
LAC BUF+2

/GET PROPER SECTION 1, 2, 3 OR 4
/ADDRESS WILL BE FIRST WORD IN BLOCK,
/65TH, 129TH OR 193RD WORD.
/SUBSCRIPT N (BUF(N)) = 64 * (BD - 1) + 1

CAL RD
ARX J
LAC I TEMAD
TAD MN1
AND (7
CLL
RTL
RTL
RTL /64 * (BD - 1)
TAD (1
DAC N
TAD (15
DAC TEM
ISZ J /J= J+1

/CHECK TO SEE IF ISOTOPE IDENT MATCHES

DZM I I1
DZM I I2
LAC (1
DAC L /DD P10 L = 1,8

P5, CAL BUF
ARX TEM
LAC I TEMAD
SAD IS11 /FIRST PART OF ID AGREES
SKP /YES
JMP P7 /NO
ISZ TEMAD

```
LAC I TEMAD
SAD IS22          /DOES 2ND PART COMPARE
JMP P20          /YES

P7,      ISZ L          /NO

LAC L
SAD C9
JMP A12          /NEVER FOUND
LAC TEM
TAD C6
DAC TEM
P10,     JMP P5          /GO SEARCH IN NEXT GROUP

/ISOTOPE FOUND, INDICATE BY SETTING
/I2 NOT ZERO
/START TRANSFER OF DATA TO ARGUMENT LIST

P20,     DAC I I2
DAC I I1
ISZ TEMAD
LAC TEMAD          /ADDRESS OF DISINTEGRATIONS
DAC TEM
CAL DATA
ARX C1

/DATA(1) = DISINTEGRATIONS

EFM
LAC I TEM
DAC I TEMAD
LFM
ISZ TEM
ISZ TEM
ISZ TEMAD
ISZ TEMAD

/DATA(2) = REL. STD. ERROR

EFM
LAC I TEM
DAC I TEMAD
LFM

/NOW FILL UP REST OF DATA ARRAY

LAC N
TAD C2
DAC TEM
ISZ TEMAD
```

ISZ TEMAD
LAC TEMAD
DAC TEM+1
CAL BUF
ARX TEM

/DATA(3) ADDRESS

/DATA (3) = ALIQUOT

EFM
LAC I TEMAD
DAC I TEM+1
LFM

ISZ TEM+1
ISZ TEM+1
LAC TEMAD
TAD (4

/DATA (4) = PRESSURE

EFM
LAC I TEMAD
DAC I TEM+1
LFM

ISZ TEMAD
ISZ TEMAD
ISZ TEM+1
ISZ TEM+1
EFM

/DATA(5) = ROTO READING

LAC I TEMAD
DAC I TEM+1
LFM
ISZ TEMAD
ISZ TEMAD
ISZ TEM+1
ISZ TEM+1
EFM

/DATA(6) = COLLECTION TIME

LAC I TEMAD
DAC I TEM+1
LFM
ISZ TEMAD
ISZ TEMAD
ISZ TEMAD
ISZ TEM+1
ISZ TEM+1

```
EFM
LAC I TEMAD          /CHI**2 INTO DATA(7)
DAC I TEM+1
LFM
/ NOW, GET TIME OF SAMPLE

LAC N
TAD (5
DAC TEM
CAL BUF
ARX TEM
LAC I TEMAD
DAC I ITOS

/ DATA ARRAY COMPLETE

P60,      RETUR
          HLT

/ B SWITCH - ENTERED AFTER FIRST TIME THRU GET SUBBY
/ IS THERE A NEW ISOTOPE REQUEST

B,        JMS REQ
          SNA
          JMP A2          /NO

/ NEW ISOTOPE REQUEST, SET SWITCH C

B1,       LAW C
          DAC VAR

B2,       LAM -0
          TAD J
          CMA
          DAC JAX      /JAX = -( J-1)
          LAC (1
          DAC J          /J = 1
          LAC I IT1
          DAC IS11
          LAC I IT2
          DAC IS22
          JMP P1

/ SWITCH C

C,        LAC J
          TAD JAX
          SMA          /IS THE RS ARRAY EXCEEDED
          JMP A12      /YES
          JMS REQ      /NO, IS THERE A NEW ISOTOPE REQUEST
          SNA
          JMP P1          /NO
          JMP B2          /YES

REQ,      0
          LAC I IT1
          SAD IS11      /NEW REQUEST
```


SKP	/NO
JMP I REQ	/YES

/COMPARE SECOND WORD

LAC I IT2
SAD IS22
CLA
JMP I REQ

/CONSTANTS AND FORMATS

	OCTAL
VAR,	CAL A
U35,	030005
.800,	320037
	TEXT

WRONG TAPE ON 3, CHANGE TAPE, THEN PRESS CONTINUE
600000

J, 0
IS11, 0
IS22, 0
N60, 60
N27, 27
TEM, 0

0
0
0
0
0
0
0

N70, 70
N200, 200
N20, 20
ID2, 0
FB, 0
U3, 030000
K, 0
N, 0
L, 0
BF, 0

/FILE BLOCK FROM MASTER BLOCK 5

/SHIFT 12 LEFT

ALS12, 0
RTL
RTL
RTL
RTL
RTL
RTL
JMP I ALS12

/MTAPE ROUTINE - SPECIAL READ ONLY - NOT ON INTERRUPT
/ENTERED WITH UNIT AND BLOCK IN AC
/ADDRESS OF CORE STORAGE FOUND BY XCT OF INST. FOLLOWING
/JMS MTAPE

MTAPE, OCTAL
0
MMSE
AND (1777
DAC BLOCK
S1, LAW 41
MMLC
SEARCH, JMS RED
AND (1777
SAD BLOCK
JMP FOUND

	CMA	
	TAD BLOCK	
	SMA	
	JMP SEARCH	/MORE THAN 0
	LAW 61	/LESS THAN 0, GO REVERSE
	MMLC	
S2,	JMS RED	
	AND C1777	
	CMA	
	TAD BLOCK	
	TAD MN2	
	SPA	
	JMP S2	
	JMP S1	
MN2,	777776	
/MTAPE		
FOUND,	XCT I MTAPE	/LAC CORE ADDRESS
	TAD MN1	
	DAC 10	
	LAM -377	
	DAC COUNS	
	LAW 42	
	MMLC	
	JMS RED	/GET FIRST CHECK SUM
	DAC CHECK	
LOAD,	JMS RED	
	DAC I 10	
	ADD CHECK	
	DAC CHECK	
	ISZ COUNS	
	JMP LOAD	
	MMBF	
	JMP .-1	
	MMRD	
	ADD CHECK	
	DAC CHECK	
	CLA	
	MMLC	
	MMSE	
	JMS OLC	
	ISZ MTAPE	
	JMP I MTAPE	
RED,	0	
	MMEF	

```

        SKP
        JMP ERR
        MMDF
        JMP RED+1
        MMRD
        JMP I RED

ERR,      WRITE
          JMS .IO2
          2
          FOR .ER
          ENDIO
          LAC BLOCK
          DAC 7
          MMRS
          HLT
.ER,      320037
          TEXT
BAD TAPE, STATUS IN AC, BLOCK IN CELL 7
          600000

OLC,      0
          LAC TTC
          DAC COUNS
          ISZ COUNS
          JMP .-1
          JMP I OLC

JAX,      0
MN1,      LAM -0
COUNS,    0
FAIL,     LAM -2
BLOCK,    0
CHECK,    0
TTC,      754300

BIT7,     002000
BIT8,     001000

START
```

TABLE 10

CSE2 CORRECTED LISTINGS

CSE2 no. beneath the page no. designates
the corrected page of CSE2 code.

CSE2 57

```
ISZ COUNTS
JMP .-2
LAW 212
JMS PDC
LAW 215
JMS PDC
JMP LOOP
TROUT, JMS RDC
JMS RNOTST
DAC IDL.
ISZ IDL.
JMS RDD
LAC ID.
DAC IDR.
JMP TIE22
DECIMAL
/ AUTOMATIC SEQUENCE INPUT
PRO1,AROUT, JMS RDD
SNA
JMP ERROR
CAL DATA.
ARX IDW.
ARX N1.
LAC N55.
JMP CHAIN
N55., 45
VROUT, JMS RDD
SNA /GET REGION #
JMP ERROR
CAL NG.
ARX IDW.
LAC I TEMAD
DAC I. /NO OF GROUPS
SNA
JMP ERROR
CMA
TAD N1.
DAC YGO /COUNTER FOR .477 LOOP
CAL LLG
ARX IDW.
```

CSE2 57 A

```

LAC I TEMAD          /GET LOWER LIMIT
DAC TEM
JMS LUI      /GET UPPER LIMIT -RETURN WITH J COUNT
DZM XX.
DZM XX.+1 /SET XX = 0.0
WRITE      /WRITE HEADING
JMS .102
2
FOR .910
ENDIO
CAL JC.
ARX ID.
ARX TEM
LAC TEMAD /ADDRESS OF LOWER LIMIT
TAD N1.
DAC I1.
DAC TEMAD
/GET TOTAL SUM BETWEEN MARKERS
.277,  LAC I TEMAD
      FLO
      ADD XX.
      DAC XX.
      LFM
      ISZ TEMAD
      ISZ J.
      JMP .277
/GET I PARTIAL SUMS
      LAC N1.
      DAC I.      /SET I =1
.477,  DZM XT.
      DZM XT.+1 /SET XT =0.0
      JMS LUI      /GET UPPER LIM OF I-TH GROUP
.377,  LAC I I1.
      FLO
      ADD XT.
      DAC XT.      /XT =XT+ JC(IDW.,N)
      LFM

```

CSE2 58

```

ISZ I1.
ISZ J.
JMP .377
EFM
LAC XT.
FDV XX.
DAC IS.    /PROBABILITY = XT/XX
LFM
WRITE
JMS .102
2
FOR .912
ARX I.
ARF IS.
ARF XT.
ENDIO
LAC ILD.
DAC TEM
ISZ TEM
ISZ I.
ISZ YGD
JMP .477
JMP VLOOP4

```

```

LUI,      0          /THIS SUB. GETS UPPER LIMIT OF 1-TH GROUP
CAL LU.   /AND SETS UP LOOP COUNTER J.
ARX IDW.
ARX I.
LAC I TEMAD
DAC ILD.
CMA
TAD TEM
DAC J.
JMP I LUI

```

```

OCTAL
.910,     320023      /C19H
TEXT
GROUP-PROBABILITIES-COUNTS
        600000

.912,     220003      /C13
        040207      /F7.4
        040012      /F10.0
        600000
DECIMAL

```


CSE2 62

```
2
FOR .211
ENDIO
READ
JMS .I01A
1
FOR .5001
ARX #IY.
ENDIO
/SET UP PULSE COUNT FOR ANALYZERS
CAL LTIME.
ARX ID.
LAC (-10000
JMS MUL
LAC IY.
DAC I TEMAD
/   FLOAT SET FOR OTHER CALCULATIONS
CAL XLTIME.
ARX ID.
LAC IY.
FLO
FMP SIXTY.
DAC I TEMAD
LFM
JMP STAR
/DG READS IN UPPER LIMITS AND ONE LOWER(FIRST) LIMIT
DG,
JMS RDD
SNA
JMP ERROR
WRITE
JMS .I02
2
FOR .120
ENDIO
READ
JMS .I01A
1
FOR .5001
ARX ISCL.
ENDIO
```

CSE2 62 A

```
WRITE      /WRITE OUT LOW LIMIT =
JMS .102
2
FOR .162
ENDIO
CAL LLG
ARX IDW.
READ
JMS .101A /READ LOW LIMIT = LLG
1
FOR .5001
ARX I TEMAD
ENDIO
LAC I TEMAD
SNA
JMP ERROR /LOWER GP LIMIT =0
WRITE
JMS .102
2
FOR .160
ENDIO
LAC ISCL. /IS NG>0 AND <21
SNA
JMP ERROR
TAD NM24.
```

SMA!SZA
 JMP ERROR
 CAL NG.
 ARX IDW.
 LAC ISCL.
 DAC I TEMAD
 DZM I1.
 LAC N1.
 DAC I.

-53-
 CSE2 63

.ACF,

WRITE
 JMS .I02
 2
 FOR .5001
 ARX I.
 ENDIO

.180,

READ
 JMS .I01A
 1
 FOR .5001
 CAL LU.
 ARX IDW.
 ARX #I.
 JMS XARAD
 JMS DONE
 ENDIO
 LAC I TEMAD
 CMA
 TAD I1.
 SMA
 JMP .ACG
 LAC I TEMAD
 DAC I1.
 ISZ I.
 LAC I.
 CMA
 ADD ISCL.
 ADD N1.
 SMA
 JMP .ACF

/IS LU(I+1) > LU(I)

NM24.,

JMP STAR
 OCTAL
 777754
 DECIMAL

.ACG,

CAL NG.
 ARX IDW.
 DZM I TEMAD
 JMP ERROR

.20,

106513
 TEXT

ISOTOPIIC ANALYSIS

-90093
 TEXT

NO. OF ISOTOPES =

-65535

.70,

106514
 TEXT

ISOTOPE NUMBERS =

-65535

CSE2 65 B

```
.ADC,      WRITE      /TYPE INDEX
           JMS .I02
           2
           FOR .5001
           ARX I.
           ENDIO

           READ        /READ COLLECTION TIME
           JMS .I01A
           1
           FOR .920
           ARF I TEMAD
           ENDIO

           ISZ TEMAD
           LAC I TEMAD      /IS FLT. PT. INPUT NEGATIVE
           SPA
           JMP STAR
           ISZ TEMAD
           ISZ I.
           ISZ I1.
           JMP .ADC
           JMP STAR

.916,      320037      /C31H
           TEXT
TIME (MINUTES) MAYPACKS SAMPLED
           600000

.918,      320037      /C43H
           TEXT
DURATION (MINUTES) OF SAMPLE COLLECTION
           600000

.920,      260013      /C11.0
           600000

TOS.,      JMS CALSB
           LAW 1
           .TOS

CT.,       JMS CALSB
           LAW 2
           .CT

DFTST,     SAD N306.
           SKP          /IS IT AN F
           JMP ERROR /NO
           JMS RDC      /LF
           JMS RDC      /CR
           LAC N50.     /BRING IN DFLINK
           JMP CHAIN

N50.,      50

.162,      320013      /FORMAT(13H LOW LIMIT =)
           TEXT
LOW LIMIT =
           600000

LLG,       JMS CALSB
           LAW 1
           .LLG
```

CSE2 66

TEM1,
END/

	OCTAL				
6400/	.AAK,	/JC			
6366/	.AAH,	/TIME			
	0	0	0	0	0
	0	0	0	0	0
6361/	.AAI,	/LTIME			
	754360	754360	754360	754360	754360
11263/	.AAQ,	/LU(4,20)			
11406/	.LLG,	/LOWER LIMIT ON CHANNEL GROUPING			
11404/	PX.,	/RHO FACTOR			
	773631	231462	/ .025		
/READ = 11413 DONT TOUCH THIS AREA FOR STORAGE					
15555/	.AAP,	/DATA(4,12)--OVERLAYS 00BP			
15635/	.AAQ,	/ISO(4,8)			
15675/	.IDDX,	/LINK5 SCRATCH STORAGE			
16145/	.TOS,	/TOS(12)			
16175/	.CT,	/CT(12)			
16225/	.NROTO,	/NROTO(15)			
16244/	.ALOC,	/ALOC(15)			
16263/					
/UNPOSA=16265					
17100/	.AAL,	/IBUFF.=INBUF			
17773/	.AAE,	/IDATR.			
17730/	.AAF,	/IDPAR			
17723/	.AAB,	/IHIMK			
17716/	.AAC,	/LOMRK			
17704/	.AAJ,	/T1			
	001400	0	001400	0	001400 0

CSE2 77

```
LFM
LAM -114 /-115
DAC I.
LAC IBUFF. 2 /ADDR. OF BKGRD.
DAC TEM
TAD N400.
DAC TEM 1
CAL TIME.
ARX ID. /TEMAD HAS ADDR. OF TIME OF STD.
```

```
.AXX, LAC I TEM /BKGRD
FLO
FDV BTIM. /DVD BY BKGRD. TIME
DAC XX.
LFM
LAC I TEM 1 /STD
FLO
FDV I TEMAD /DVD BY STD TIME
FSB XX.
DAC XX.
ADD S2.
DAC S2. /ADD TO NET COUNT RATE
LAC XX.
FMP I TEMAD /NORMALIZE FOR PRESERVING ACCURACY
ADD .ABE
FXA
DAC I TEM 1
ISZ TEM
ISZ TEM 1
ISZ I.
JMP .AXX
```

/NOW COMPUTE THE GEOMETRY FACTOR WHICH IS D/M OR AMT. READ IN
/BY EXPERIMENTER, TIMES THE EXPF FACTOR. THIS CHANGE
/MADE ON 11/21/66. THE OLD GEO. FACTOR HAD THE ABOVE PRODUCT
/IN THE DENOMINATOR AND THE NET COUNT RATE IN THE NUMERATOR

```
EFM
LAC XT.
FMP .ABY / .693 * ELAPSED TIME
FDV XH. /XT * .693/ HALF LIFE
DAC YY.
FCS YY.
DAC TEM+2
EXTERNAL EXPF.
CAL EXPF.
ARF TEM+2
LAC AMT. /D PER M (NOT MICROCURIES)
FMP I TEMAD
DAC YY. /GEO. FACTOR TO BE STORED ON TAPE
LFM
```

CSE2 66 A

17672/	001400	0	001400	0		
	.AAG,	/T2				
	002400	0	002400	0	002400	0
	002400	0	002400	0		
1575/	IS.,					
1576/	IH.,					
1577/	II.,					
2/	.AAN,	/IBB				
22/	IRUNN.,					
24/	ID.,					
25/	IEX.,					
26/	.AAD,	/XLTIME				
	006740	0	0 6740	0	006740	
	0	006740	0	006740	0	
40/	.AAM,	/NG				
	0	0	0	0		
44/	BLOCK,					
45/	IY.,J.,I1,					
46/	FAK,IERR.,I1.,IDS1.,					
47/	XORG,ICHAN.,COTP,					
50/	ASET,DATPT,IX.,					
51/	XCONST,ISTR.,I.,					
52/	MINLO,ICNU.,					
53/	MSKX,ICH.,					
54/	MINHI,IPRT1.,					
55/	DSPLC,IPRT2.,					
56/	YORG,IFIN.,					
57/	SCALE,IBASKT.,					
60/	YGO,IA.,I2.,IDS2.,					
61/	XAXIS,COUNS,					
62/	DASH,IUNIT.,					
63/	DASHNM,XXS.,					

CSE2 94

LINK ONE OF A SEQUENCE
/LINK1 LOADED AT BLOCK 10 (OCTAL) ON TAPE 4
DECIMA FIDDEC
/

OCTAL
T1.=132
LINKER=305
T2.=121
ISOC.=141
NG.=170
LU.=153
ISO.=161
IBB.=165
IBUFF.=150
JC.=135
SUB=2605 /END OF LINK
ID.=24
BIT0=263
BIT2=265
BIT3=266
IDW.=70 /ID-1
AUTX=10
AUTY=11
SEC.=323
MODE=16455

1600/

DECIMAL
.AAA, JMP AA2.

CSE2 94 A

HT,	JMS CALSB /STD. SUM BETWEEN MARKERS (ARRAY)
	LAW 2
	XX.
A.,	JMS CALSB
	LAW 9
	LAW 2
	.AAB
LLG,	JMS CALSB
	LAW 1
	.LLG
G.,	JMS CALSB
	LAW 20
	LAW 2
	.AAC
Y.,	JMS CALSB
	LAW 2
	.AAD
B.,	JMS CALSB
	LAW 2
	.AAG
S.,	JMS CALSB
	LAW 2
	.AAH
IDENT,	JMS CALSB
	LAW 2
	.IDDX
HL.,	JMS CALSB
	LAW 2
	.AAN
GG.,	JMS CALSB
	LAW 2
	.AAO

CSE2 96

```
ADD ID.
XCT LINKER+9      /DAC ILD
LAC IBUFF.+2
DAC IY.
XCT LINKER+3      /JMS RUST
JMS DELA /WAIT 35 MS. FOR TAPE DESELECT
/UNLOAD THE BKGD TIME
/
LAC IY.
XOR MODE
TAD X120
DAC IY.
LAC I IY.
DAC BTIM.
ISZ IY.
LAC I IY.
DAC BTIM. +1
/ DO CHANNEL GROUPING. B(I) IS BKGD RATE FOR THE I-TH
/ GROUP AND Y(I) IS THE OBSERVED TOTAL COUNT FOR THE I-TH
/ GROUP
/
LFM
CAL NG.
ARX IDW.
LAC I TEMAD
DAC NGG
SNA
XCT LINKER+6
LAC ONE
DAC I.
CAL LLG
ARX IDW.
LAC I TEMAD
DAC LLGG
DAC IX.
DAC J.      /INDEX TO UNGROUPED ARRAYS ( SINGLE )

.AAT,
CAL B.
ARX I.
LAC TEMAD
DAC SPARE
CAL Y.
ARX I.
LAC TEMAD
DAC SPARE+1
EFM
LAC .AAP
DAC I SPARE
DAC I SPARE+1
LFM
CAL LU.
ARX IDW.
ARX I.
LAC I TEMAD
DAC IY.
```

```

ADD ONE
RCR      /DIVIDE BY 2
DAC SPARE+5
CAL IBB. /GET BLOCK # FOR ISOTOPES 1 AND 2
ARX IDW.
LAC SPARE+5
ADD I TEMAD
/ CALCULATE UNIT NO., BIT2 INDICATES UNIT 3
TAD BIT2
TAD BIT0
XCT LINKER+9      /DAC ILD
LAC Ibuff.+2
DAC IY.
XCT LINKER+3      /JMS RUST
JMS DELA
/ STANDARD BLOCK NOW IN BUFFER, NEXT CHECK FOR ODD OR EVEN
/ ISOTOPE. ODD STORED ON FIRST HALF OF BLOCK AND EVEN
/ STORED ON SECOND HALF.
CAL ISO.
ARX IDW.
ARX J.
LAC I TEMAD
AND ONE
SZA      /EVEN OR ODD
JMP ODD
LAC X129
JMP ODD+1
ODD,     LAC ONE
DAC M.      /M=1 OR 129
/ NEXT, GET STAND. TIME IN ORDER TO RE-NORMALIZE.
/ CORRECTED COUNT RATE WAS MULTIPLIED BY TIME IN A1
/ SEQUENCE, NOW WE DIVIDE BEFORE CALCULATING
ADD X123
ADD IY.
/ AC NOW HAS ADDRESS OF Ibuff+124 OR Ibuff+252

```

CSE2 98 A

```
DAC IX.
LAC I IX.
DAC YY.
ISZ JX.
LAC I IX.
DAC YY.+1
CAL LL.
ARX IDW.
ARX NGG
LAC I TEMAD          /GET UPPER LIMIT OF LAST GROUP
TAD MN54
SPA
XCT LINKER+6         /JMP ERROR
JMS DIV
LAC THREE
TAD N54
TAD M.               /CUP - 44)/3 + 44 + M.
DAC M2.
LAC M.
TAD MSK18
ADD LLGG
DAC I.              /I= M +(LLG-1)
CAL HT
ARX J.
LAC TEMAD
DAC XX1
EFM
LAC .AAP
DAC I XX1 /SET SUM =0.0
LFM
.AAZ.
CAL Ibuff.
ARX I.
LAC I TEMAD
FLO
```

```

      FDV YY.
      ADD I XX1
      DAC I XX1 /STORE SUM IN HI ARRAY
      LFM
      ISZ I.
      LAC I.
      CMA
      ADD M2.
ADD,   ADD ONE
      SMA
      JMP .AAZ
/XX IS THE SUM OF ALL STD. CHANNELS BETWEEN THE UPPER AND LOWER
/MARKERS. STORED IN COMMON AND USED IN LINK4 TO CALC.
/THE TOTAL DISINTEGRATIONS.COMMON IS OCTAL 11413 AND 11414
      LAM 0
      TAD M.
      DAC M.
      LAC ADD
      DAC KIND
      LAC ONE
      DAC I.
      LAC LLGG
      DAC IX.
      DAC K.
.ABA,  CAL LU.
      ARX IDW.
      ARX I.
      LAC I TEMAD
      DAC M2.
      CAL G.
      ARX J.
      ARX I.
      LAC TEMAD
      DAC SPARE+3
      EFM
      LAC .AAP
      DAC I SPARE+3
      LFM
.ABC,  LAC M.
      ADD IX.
      DAC SPARE+1
      CAL IBUFF.
      ARX SPARE+1
      LAC I TEMAD
      FLO
      FDV YY.
      ADD I SPARE+3
      DAC I SPARE+3
      LFM
      ISZ IX.
      LAM -44
      DAC SPARE+7
      LAC K.
KIND,  ADD ONE
      DAC K.
      TAD SPARE+7
      SMA
      JMP KSET
KSR,   LAC K.
      CMA
      ADD M2.

```

CSE2 100

```
SMA
JMP .ABC
EFM
LAC I SPARE+3
FDV I XX1
DAC I SPARE+3
LFM
ISZ I.
LAC I.
CMA
ADD NGG
ADD ONE
SMA
JMP .ABA
/ NEXT GET BUFFER ADDRESSES IN ORDER TO PICK UP THE
/ HALF LIFE AND GEOMETRY AND ALPHA-NUMERIC IDENT.
LAC IBUFF.+2
ADD M. /ODD AND EVEN INDEX
ADD X115 /GET SET UP FOR LOADING ALPHA - N. IDENT
DAC AUTX
CAL IDENT
ARX J.
LAC I AUTX /GET 1ST WORD OF ALPHA-NUMERIC
DAC I TEMAD /STORE IN IDENT(J)
ISZ TEMAD
LAC I AUTX /GET 2ND ALPHA WORD
DAC I TEMAD /STORE IN IDENT(J+1)
ISZ AUTX /BY PASS AMT.
ISZ AUTX
CAL HL.
ARX J.
LAC I AUTX /GET AND STORE HALF LIFE
DAC I TEMAD
ISZ TEMAD
LAC I AUTX
DAC I TEMAD
CAL GG. /GET AND STORE GEOMETRY
ARX J.
LAC I AUTX
DAC I TEMAD
ISZ TEMAD
LAC I AUTX
DAC I TEMAD
.849, LFM
LAC J.
ADD ONE
DAC J.
CMA
ADD ISOCC
ADD ONE
SMA
JMP .AAX
/ END OF BIG LOOP
/ THE FOLLOWING SET USED TO MODIFY CHAIN TO BRING IN ONLY
/ TWO BLOCKS INSTEAD OF FIVE
LAC LINKER
AND MSK7
TAD THREE
DAC SPARE
```

CSE2 101

```

LAM -1
DAC I SPARE
LAC N15. /SECOND LINK
XCT LINKER /JMP CHAIN - GO GET 2ND LINK
SETJ, LAC ATH
DAC JIND
JMP JSR

KSET, LAC ATH
DAC KIND
JMP KSR

BTD, LFM
WRITE
XCT LINKER+8
2
FOR - 140
ARX IDW.
ENDIO
XCT LINKER+6 /JMP ERROR
/ROUTINE TO DELAY 35 MS.
OCTAL
DELA, 0
CLA
MMLC
JMS OLC
CLA
MMSE
JMS OLC
JMP I DELA

OLC, 0
LAC TTCC
DAC SPARE+7
ISZ SPARE+7
JMP -1
JMP I OLC

TTCC, 754777
.2010, 320020
TEXT
TOO HOT, RECOUNT
400000
600000

.860, 320037
TEXT
BACKGROUND COUNT TOO OLD ON ANAYLZER #
020003
600000
/ CONSTANTS USED BY LINK ONE - NOT IN COMMON WITH THE MAIN PROG.

MSK18, 777777
N15., 15
IY., 0
MSK7, 007777
X120, 170
ATH, ADD THREE
THREE, 3
ONE,N1., 1
TWO, 2
N41., 41

```

CSE2 102

```

LLGG,      0
IX.,       0
I.,        0
J.,        0
K.,        0
IS000,     0
M.,        0
N400.,     400      /265
M2.,       0
YY.,       0
           0
.ABX,      015702    /7200 SEC = 2 HOURS
           000000
FIVE,N5.,  5
X129,      201
X123,      173
X114,      162
X115,      163
NGG,       0
N54,       54
MN54,      777724
SPARE,     0
           0
           0
           0
           0
           0
           0
           0
           0
           0
.AAP,      0
           0
XX1,       0
           0
/SET LOWER COMMON
SUB/       .AAC,      /G(8,20)
           DECIMAL
.+320/     .AAB,      /A(8,9)
.+144/     BTIM.,     /BKGD
.+2/       .GGS,      /GS(8,17) - WORKING ARRAY
.+272/     XX.,       /HT ARRAY - ENDS AT 4170
/SET UPPER COMMON AREA
           OCTAL
15675/     .AAD,      /GG(8)
           DECIMAL
.+16/      .AAN,      /HL(3)
.+16/      .AAG,      /B(20)
.+40/      .AAD,      /Y(20)
.+40/      .AAH,      /S(20)
.+40/      .IDDX,     /IDENT
/IDENT ARRAY ENDS AT 16145
           OCTAL
11406/     .LLG,      /LOWER LIMIT ARRAY FOR REGIONS
START

```


LINK 2 OF AUTO SEQUENCE - BLOCK 15 ON UNIT FOUR

DECIMA FIODEC

/T2,T1,NG,ISOC ARE COMMON TO MAIN AND LINKS

/

/G,A,GG,HL,B,Y, BTIM. ARE COMMON TO LINKS ONLY

/

/EQUIVAL. OR MAIN COMMON

/

OCTAL

IDW.=70

ID.=24

T1.=132

T2.=121

NG.=170

ISOC.=141

LINKER=305

MODE=16455

SUB=2605

TIME.=124

/

1600/

DECIMAL

JMP .AAA

A., JMS CALSB

LAW 9

LAW 2

.AAB

G., JMS CALSB

LAW 20

LAW 2

.AAC

Y., JMS CALSB

LAW 2

.AAD

B., JMS CALSB

LAW 2

.AAG

S., JMS CALSB

LAW 2

.AAH

GS., JMS CALSB

LAW 17

LAW 2

.GGS

/ GENERATE WEIGHTING FACTORS , S(1) WHICH ARE THE APPROXIMATE

/ VARIATION OF COUNTER ERRORS. PX HS FRACTIONAL INPUT

/

/

.AAA, CALST

SET2W

CAL TIME.

ARX ID.

EFM

LAC I TEMAD

CSE2 107

DAC DELT.

FMP DELT.
DAC DELT2.

LFM
CAL NG.
ARX IDW.
LAC I TEMAD
DAC NGG
LAC ONE
DAC #I.

.AAJ,

CAL Y.
ARX #I.
LAC TEMAD
DAC TEM+2
EFM
LAC PX.
FMP I TEM+2
DAC XX. /XX=(PX*Y(I))**2

FMP XX.
DAC XX.
LFM
CAL B.
ARX #I.
LAC TEMAD
DAC TEM+1
EFM
LAC DELT.
FMP PX.
FMP I TEM+1
DAC YY.

FMP YY.
DAC YY. /YY=DELT*PX*B(I))**2

.B60,

LFM
CAL S.
ARX #I.
LAC TEMAD
DAC TEM+0
EFM
LAC I TEM+2
ADD XX.
DAC TEM+3 /C(TEM+3)=Y(I)+XX
LAC DELT2.
FMP I TEM+1
FDV BTIM.
ADD TEM+3
ADD YY.
DAC I TEM+0

/S(I)=YY+Y(I)+XX+DELT**2*B(I)/BTI4

LFM
ISZ I.
LAC I.
CMA
ADD NGG

```

      FDV I TEM+3
      DAC I TEMAD
      LFM
/ GSC(1,N+1) =1.0/A(1,1)
      LAC N22.
      XCT LINKER                                /JMP CHAIN
.AZZ,     LAC LINKER
          AND MSK7
          TAD THREE
          DAC TEM
          LAM -4
          DAC I TEM
          LAC N41.
          XCT LINKER
/CONSTANTS
          OCTAL
I.,       0
N22.,    00
.20000,  32001
          TEXT
MATRIX IS SHOT
          400000
          600000
N.,       0
MSK7,    007777
THREE,   3
N41.,    41
J.,      0
K.,      0
MN999,   754361
KK.,     0
DELT.,   0
         0
DELT2.,  0
         0
XX.,     0
         0
YY.,     0
         0
TEM,     0
         0
         0
         0
         0
         0
         0
         0
TWO,     2
N27.,    27
ONE,     1
FOUR,    4
NINE,    11
HUNDRO,  144
NGG,     0
ISOCG,   0
PX.,     773631      231462      /RHO .025
          DECIMAL
.AAM,    0           0

```

LINK 4 OF AUTO-SEQUENCE

/ LINK 4 IS LOADED AT BLOCK 27 (OCTAL) ON UNIT FOUR

DECIMAL FIDEC
EXTERNAL SORTF.,EXPF1
OCTAL
REFI.=321
SEC.=323
ID.=24
IDW.=70
LINKER=315
SUB=2605
ISOC.=141
IRUNN=22
DATA=144
N6.=224
LTIME.=127
TIME.=124

1600/

DECIMAL
.AAA, JMP .AA

A., JMS CALSB
LAW 9
LAW 2
.AAB

Y., JMS CALSB
LAW 2
.AAD

IDENT, JMS CALSB
LAW 2
.IDDX

HL., JMS CALSB
LAW 2
.AAN

GG., JMS CALSB
LAW 2
.AAO

HT., JMS CALSB /DISINTEG. ARRAY AND ALSO STD. SUM ARRAY
LAW 2
.HHT

.AA, SET2W
CAL ISOC.
ARX IDW.
LAC I TEMAD
DAC ISOCC
CAL TIME.
ARX ID.
LAC TEMAD
DAC TIMID

/HT(J)=GG(J)*A(J,1)*ALIQUT*DECAY/ (XX. *LIVE TIME)
/ WHERE GG IS GEOMETRY= AMT. * ASSOCIATED DECAY AND XX
/ IS STD. CHANNEL SUM (COUNT RATE) BETWEEN THE MARKERS

LAC ONE
DAC J.
.AAX, CAL HT.
ARX J.
LAC TEMAD
DAC TEM
CAL DATA /GET

```
ARX IDW. /ALIQUT
ARX THREE /MULTIPLIER
LAC TEMAD
DAC TEM 2
CAL GG.
ARX J.
LAC TEMAD
DAC TEM+1
CAL HL.
ARX J.
LAC TEMAD
DAC TEM+3
CAL A.
ARX J.
ARX ONE
EFM
LAC I TEM 2
FMP I TEMAD
FMP I TEM+1
FDV I TIMID
FDV I TEM /DIVIDE BY STD. SUM FROM LINK1
DAC I TEM /STORE BACK IN HT ARRAY
LAC SEC.
FSB REFT.
FDV I TEM+3
FMP .ABY
DAC TEM+2
CAL EXPF1
ARF TEM+2
LAC I TEMAD
FMP I TEM
DAC I TEM
LFM
ISZ J.
LAC J.
CMA
ADD ISOCC
ADD ONE
SMA
JMP .AAX
```

/ WRITE OUT "RUN NO. - SAMPLE ID. - SAMPLE TIME"

```
WRITE
XCT LINKER+8
2
FOR .BAA
ARX IDW.
ENDIO
```

```
CAL DATA /SAMPLE ID
ARX IDW.
ARX ONE
LAC TEMAD
DAC TEM
```

```
ISZ TEMAD
LAC TEMAD
DAC TEM+1
```

```
CAL DATA /TIME OF SAMPLE
ARX IDW.
ARX N6.
LAC I TEMAD
DAC TEM 2
CAL LTIME.
ARX ID.
LAC I TEMAD
JMS 0 V /DIVI BY 10000 PULSES FOR PRINT OUT ONLY
END MN999
DAC TEM+5
WRITE
XCT LINKER+8
2
FOR .CRR /WRITE OUT HEADING ONLY
ENDIO
WRITE
XCT LINKER+8
2
FOR .CAA
ARX IRUNN
ARX IRUNN+1
ARX I TEM
ARX I TEM+1
ARX TEM 2
ARX TEM+5
ENDIO
/ WRITE "NUCLIDE - COUNTS - VARIANCE - DISINTEGRATIONS"
WRITE
XCT LINKER+8
2
FOR .DAA
ENDIO
LAC ONE
DAC J.
DAC L.
ISZ L.
.AAY, CAL IDENT /NUCLIDE
ARX J.
LAC TEMAD
```

CSE2 128 A

```
DAC TEM
TAD ONE
DAC TEM+1
CAL HL.
ARX J.
LAC TEMAD
DAC TEM 7
CAL A.      /SAMPLE PERIOD COUNTS
ARX J.
ARX ONE
LAC TEMAD
DAC TEM+2
CAL A.      /CALCULATE RELATIVE STANDARD ERROR
ARX J.      /((A(J,J+1)**.5)/A(J,J)=TEM+4
ARX L.
LAC TEMAD
DAC TEM+3
CAL SORTF.
ARF I TEM 3
EFM
LAC I TEMAD
FDV I TEM 2
DAC I TEM 7      /INSERT ERROR INTO HL ARRAY
DAC TEM+4
LFM
CAL HT.
ARX J.
WRITE
XCT LINKER+3
2
FOR .EAA
ARX I TEM
```

CSE2 129

```

ARX I TEM+1
ARF I TEM+2
ARF TEM+4
ARF I TEMAD
ENDIO
ISZ L.
ISZ J.
LAC J.
CMA
ADD ISOCG
ADD ONE
SMA
JMP .AAY
/BRING IN LINK 5
LAC N46.
XCT LINKER
/ FORMAT TABLES
OCTAL
.BAA, 200000 / (
      400000 / /
      400000 / /
      120014 /14H
      TEXT
-----
      520016 / /14H
      TEXT
ANALYZER #
      020003 /,13
      400000
      600000
.CRR, 320037 / (37H
      TEXT
      RUN NO. SAMPLE ID SAMPLE TIME (MIN) LIVE TIME (MIN)
      400000
      600000

.CAA, 300004 / (4X
      142003 /,2A3
      100005 /,5X
      142003 /,2A3
      100010 /,8X
      020006 /,16
      100013 /,11X
      020006 /,16
      400000
      400000
      600000
.DAA, 320037
      TEXT
      NUCLIDE COUNTS REL STD ERROR DISINTEGRATIONS
      400000
      600000
.EAA, 300004 / (4X
      142003 /,2A3
      100002 /,2X
      040012 /,F10.0
      040154 /,F12.3
      100011 /,9X
      000313 /,E11.6
      600000

```



```

.FAA,      320010      / (8H
TEXT
-----
        600000      / )

/ CONSTANTS
MN999,      754361      /-9999
L.,         0
J.,         0
K.,         0
TEM,        0
            0
            0
            0
            0
            0
            0
            0
            0
ONE,        1
TWO,        2
THREE,      3
X11,        13
SIXTY,      006740      0
TIMID,      0
IS0CC,      0

.ABY,       000542
            344400      / .693
N46.,       46

/SET LOWER COMMON
SUB/        .AAC,      /G(8,20)
            DECIMAL
.+320/      .AAB,      /A(8,9)
.+144/      BTIM.,    /BKGD
.+2/        .GGS,      /GS(8,17) WORKING ARRAY
.+272/      .HHT,      /HT ARRAY ENDS AT 4170
/SET UPPER COMMON
OCTAL

15675/      .AA0,      /GG(8)
            DECIMAL
.+16/       .AAN,      /HL(8)
.+16/       .AAG,      /B(20)
.+40/       .AAD,      /Y(20)
.+40/       .AAH,      /S(20)
.+40/       .IDDX,     /IDENT

/IDENT ARRAY ENDS AT 16145

N46. +1/

START

```

TABLE 11

PSEUDO FORTRAN FOR VROUT

```
      I = YGO = NG                                /no. of groups
      J = LU(IDW.,I)                               /get upper limit
      K = LLG(IDW.) + 1                             /get lower limit
      /get total sum between markers
      XX = 0.0
      DO 277 L = K,J
277  XX = XX + JC(IDW,L)
      DO 477 L = 1,I                               /get I partial sums
      XT = 0.0
      J = LU(IDW,L)
      DO 377 N = K,J
377  XT = XT + JC(IDW.,N)
      IS = XT/XX                                    /get probability
      WRITE OUT 2, 912, L IS, XT
      K = J + 1
477  CONTINUE
```

-77-
TABLE 12
LINK 5 LISTING

LINK 5 OF AUTO SEQUENCE
/LINK 5 LOADED IN BLOCKS 46-47 OF TAPE 4

DECIMA FIDDEC
 EXTERNAL SQRTF.
 OCTAL
 LINKER=305
 IRUNN.=22
 DATA.=144
 TIME=124
 IBUFF.=150
 SUB=2605
 ISOC.=141
 NG.=170
 ID.=24
 IDW.=70
 ERROR=1422

1600/

	DECIMAL
.AAA,	JMP .AA
A.,	JMS CALSB
	LAW 9
	LAW 2
	.AAB
Y.,	JMS CALSB
	LAW 2
	.AAD
G.,	JMS CALSB
	LAW 20
	LAW 2
	.AAC
S.,	JMS CALSB
	LAW 2
	.AAH
B.,	JMS CALSB
	LAW 2
	.AAG
HT,	JMS CALSB
	LAW 2
	.HHT
HL,	JMS CALSB
	LAW 2
	.AAN
IDENT,	JMS CALSB
	LAW 1
	.IDDX
.AA,	CALST
	SET2W
	CAL ISOC.
	ARX IDW.
	LAC I TEMAD
	DAC ISOCC
	CAL NG.
	ARX IDW.
	LAC I TEMAD
	DAC NGG
	CAL TIME
	ARX ID.
	EFM
	LAC I TEMAD
	DAC DELT

```

/PAGE 2 OF LINK 5
      LFM
      DZM X2      /ZERO OUT SUMMER
      DZM X2+1
/SET UP OUTER LOOP  I=1,NGG
      LAC ONE
      DAC I.
.100,      DZM RR
      DZM RR+1    /ZERO OUT NEXT SUMMER
/SET UP INNER LOOP  J=1,ISOCC
      LAC ONE
      DAC J.
.75,      CAL A.
      ARX J.
      ARX ONE
      LAC TEMAD
      DAC TEM
      CAL G.
      ARX J.
      ARX I.
      EFM
      LAC I TEM
      FMP I TEMAD
      ADD RR
      DAC RR
/RR=RR + A(J,1)*G(J,I)
      LFM
      ISZ J.
      LAC J.
      CMA
      ADD ISOCC
      ADD ONE
      SMA
      JMP .75
/THRU WITH INNER LOOP, GET SQUARE TOOT OF VAR - S(I)
      CAL S.
      ARX I.
      LAC TEMAD
      DAC TEM+1
      CAL SQRTF.
      ARF I TEM+1
      EFM
      LAC I TEMAD
      DAC SS
/CALCULATE THE RESIDUALS
      LFM
      CAL Y.
      ARX I.
      LAC TEMAD
      DAC TEM
      CAL B.
      ARX I.
      EFM
      LAC DELT
      FMP I TEMAD
      DAC TEM+4
      LAC I TEM
      FSB RR
      FSB TEM+4
      FDV SS

```

```

DAC RR
/PAGE 3 OF LINK 5
FMP RR
ADD X2
DAC X2      /X2=X2 + RR**2
LFM
ISZ I.
LAC I.
CMA
ADD NGG
ADD ONE
SMA
JMP .100
/THRU WITH OUTER LOOP
/NORMALIZE THE CHI SQUARE
LAC ISOCC
CMA
TAD ONE
TAD NGG
DAC IG
DAC SUB+10
SPA
JMP .200
SNA
JMP .200
/DEGREES OF FREEDOM MUST BE >0 DEG. OF F.=NGG-ISOCC
.150,   FLO      /FLOATF(NG-ISOCC)
DAC XZ
FMP TF      /TF=2.0
DAC SS
CAL SORTF.
ARF SS
LAC X2
FSB XZ
FDV I TEMAD
DAC CHI
DAC SUB+12
LFM
WRITE
XCT LINKER+8
2
FOR .500
ARF CHI
ENDIO

.200,   WRITE
XCT LINKER+8
2
FOR .600
ARX IG
ENDIO
```

/PAGE 4

/WRITE OUT SAVE ANALYSIS, Y(YES),N(NO)

```
.602,  WRITE
      XCT LINKER+8
      2
      FOR .620
      ENDIO
      XCT LINKER+5      /READ FIRST LETTER(JMS RDC)
      DAC TEM
      XCT LINKER+5
      XCT LINKER+5
      LAC TEM
      SAD YY      /SAVE OR NOT
      JMP .605  /YES
      SAD NN      /IS IT AN N
      SKP
      JMP .602  /ILLEGAL LETTER
```

/DON'T SAVE, RESET CHAIN TO BRING IN 5 BLOCKS OF MAIN

/WRITE OUT THE DASHES

```
.750,  WRITE
      XCT LINKER+8
      2
      FOR .FAA
      ENDIO
      LAC THREE
      ADD LINKER
      DAC TEM
      LAM -4
      DAC I TEM
      JMP ERROR+1
.605,  LAC N121  /BRING IN LINK 6
      XCT LINKER      /JMP CHAIN
      OCTAL
.620,  320034
      TEXT
SAVE ANALYSIS, Y(YES), N(NO)
600000
```

/PAGE 5

/FORMAT TABLES

	OCTAL	
.FAA,	320010	/C8H
	TEXT	

	600000	/)
.500,	300004	/C4X
	120011	/,9H
	TEXT	
CHI**2 =		
	040147	/F7.3
	600000	
.600,	300004	
	120012	/10H
	TEXT	
DEG. F. =		
	020003	/13
	600000	

/CONSTANTS FOR LINK 5

N1,	1
BIT0,	400000
TIC,	754777
YY,	331
N121,	121
NN,	316
L.,	0
N5,	5
N3,	3
N375,	375
N35,	35
CHI,	0
	0
IG,	0
THREE,	3
TEM,	0
	0
	0
	0
	0
	0
XZ,	0
	0
TF,	002400
	0
SS,	0
	0
X2,	0
	0
RR,	0
	0
I.,	0
ISOCC,	0
ONE,	1

```
NGG,      0
DELT,     0
          0
J.,       0
          LAC LAST /TEST TO BE REMOVED
/SET LOWER COMMON AT SUB
SUB/      .AAC,      /G(8,20)
          DECIMAL
.+320/    .AAB,      /A(8,9)
.+144/    BTIM.,     /BKGD
.+2/      .GGS,      /GS(8,17)  WORKING ARRAY
.+272/    .HHT,      /HT(8)  DISINTEGRATIONS

/SET UPPER COMMON IN UNUSED OTS AREA
          OCTAL
15675/    .AAO,      /GG(8)
          DECIMAL
.+16/     .AAN,      /HL(8)
.+16/     .AAG,      /B(20)
.+40/     .AAD,      /Y(20)
.+40/     .AAH,      /S(20)
.+40/     .IDDX,     /IDENT
/IDENT ARRAY ENDS AT 16145

J.+1/
START
```

FIGURE 1
Retrieval ("Get") Flow Diagram

FIGURE 1 cont'd
Retrieval ("Get") Flow Diagram

DISTRIBUTION

Number
of Copies

53

Battelle-Northwest

R. T. Alleman
J. M. Atwood
J. M. Batch
F. P. Brauer
J. C. Campbell
L. D. Coffin (10)
B. L. Combs
L. F. Coleman (4)
R. E. Connally
R. Y. Dean
J. L. Deichman
B. H. Duane
G. E. Driver
H. Harty
R. K. Hilliard
F. E. Holt
J. H. Kaye
C. E. Linderoth
W. L. Nicholson
J. D. McCormack
H. J. Rieck
G. J. Rogers
J. T. Russell
S. J. Simmons
R. W. Walker
W. R. Weissenberger
E. L. Wells
N. P. Wilburn (10)
M. E. Witherspoon
Central Files (3)

12

AEC-Washington

R. R. Newton, DRD&T

1

Division of Technical Information Extension

Oak Ridge

2

AEC, DRPT

P. G. Holsted

3

AEC, RL

C. L. Robinson
R. K. Sharp
Technical Library

APPENDIX A

REPORT PROGRAM CHANGES NEEDED

AFTER COMPILATION

If Report is changed and the Fortran program is compiled, it is necessary to make the following changes to the symbolic listing.

1. After the last EXTERNAL statement on page 1, insert this sequence:

```
OCTAL  
IRUN. = 26  
.AAD = 30  
DECIMAL
```

This sets the data array to GET's common area.

2. On the last page, delete the line which is tagged by .AAD.