DATA PROCESSING ORIENTATION

"EIT" PROGRAM

FOR BETTER HIGHWAYS

Data Processing
owa Highway Commission
Ames, Iowa

DATA PROCESSING ORIENTATION

'EIT' PROGRAM

DATA PROCESSING
IOWA HIGHWAY COMMISSION
AMES, IOWA

PURPOSE

This program is to acquaint, through practical application, EIT's with the function, capabilities, and related problems, of the Iowa Highway Commission Data Processing Department. This is to be accomplished by guiding all EIT's through a two-week schedule that gives exposure to all phases of data processing and related problems.

SCHEDULE OF ACTIVITIES

ACTIVITY	TIME (DAYS)
I. ORIENTATION (Complete Overview)	1/4
II. SYSTEMS DESIGN . Design of two-program system . Write program specifications	1 3/4
III. PROGRAMMING AND JOB CONTROL (JCL) . Learn subset of PL/1 Language . Code Programs . Learn subset of JCL . Code JCL	2 1/2
IV. OPERATIONS, PROGRAM ERROR CORRECTING (DE-BUGGING) AND TSO OVERVIEW . Hardware system orientation . TSO Overview 'EDITING' . Run batch tests shots . De-bug own programs	4
V. TECHNICAL SERVICES . Transfer batch programs to run under TSO	1
VI. CRITIQUE AND SUMMARY	1/2
VII. EXHIBITS . EIT Questionnaire . Sample program flowchart . Sample program data card . Sample program listing	

I. ORIENTATION

A. General--D.P. is a service function to others

B. Trends of growth in D.P.

1. Card systems

2. Tape systems

3. Disk systems (on-line)

C. Need for generalized background

1. Accounting - old > 50%; now < 50%;

2. Engineering - old < 50%; now > 50%

D. Processing of work request

1. Who initiates

2. Actual routing

E. Processing program/system request

1. Who initiates

2. Sections of D.P.

F. Explain balance of activities on schedule

II. SYSTEMS DESIGN

A. SYSTEMS ANALYSIS--Definition of--

The examination of an activity, procedure, method, technique or a business to determine what must be accomplished and how the necessary operations may best be accomplished.

B. History of Systems Analysis

1. Relationship between man and machine
2. Development as a separate function

C. Function at Highway Commission Today

1. Recognition of a need by potential user
2. Investigation of ramifications, including justification, by analyst
3. General definition of system
   a. For User
   b. For D.P.
4. Breakdown into manageable units of work
5. Writing of specifications
6. Relationship between S.A. and Programmer
7. Creation of forms and manuals
   a. Users
   b. Operations

D. Future Trends

III. PROGRAMMING AND JOB CONTROL (JCL)

A. Programming

We would like for you to become exposed to the nature of a programmers work. After specifications have been completed by a Systems Analyst, they are usually given to a programmer. The programmer interprets the problem, codes it in some compiler language and tests the program. After the programmer and analyst feels the program is completed it is put into production status.

The compiler language we will be using is a subset of PL/1.

B. PL/1 TOPICS

1. Character Set
2. Identifiers
   a. Procedure names
   b. File names
   c. Data items
   d. LABELS
   e. PL/1 Words

Must begin with A-z, $, @, # and cannot be greater than 31 characters in length. Certain names are also used by operating system (main procedure label, file name) and they should be limited to seven characters.
A larger name is truncated.

C. Data

1. Raw Data (Accounting: interest amount)

2. Programmer representation of value (DCL Interest DEC Fixed (5,2);)

3. Compiler input data (i.e. Representation as punched in a card, 0 is 0 punch, - is 11 punch)

4. Internal data (i.e. 0 is F0 byte stored internally)

EXAMPLE:

```
DCL INTEREST      DEC FIXED (5,2)      INIT (0);
 variable          attributes          constant
```

D. Data Types

1. DCL DATE CHAR(6);--Character String

2. DCL X DEC FIXED (6,1)--real decimal fixed point

3. DCL Y BIN FIXED (15,2)--real binary fixed point

4. DCL Z DEC FLOAT (6)--real decimal floating point

5. DCL W BIN FLOAT (7)--real binary floating point always assumed

E. Operators

1. Prefix + and -

2. ** exponentiation

3. * multiplication

   / division

4. Infix + and -

   A <u>prefix</u> operator is one which is not preceded by another identifier or constant (EX.--x).

   An <u>infix</u> operator is one which is preceded by an identifier or constant. (EX.--a-b)

   EX: Others: <, <=, =, ¬=, >=, >, ¬, &, 1,

F. Order of Evaluation

1. Inside ()
2. Exponentiation, prefix +, prefix -
3. Mult-Div
4. Add-Subt
5. Equal levels done right to left

G. Expressions

An expression may be a single constant or a name or it may be a combination of them, including operators and other delimiters. Delimiters are like (), `´, etc.

(EX.:
```
X
X + y
( (X + Y - (Z**2))/3)* (a + b)
-x
a = x/c; )
```

H. Control Statements

1. IF
2. DO — Explain with examples
3. GO TO
4. CALL - Unnecessary to mention
5. RETURN
6. END

   Explain with examples

7. STOP - No need to mention

I. Structures & Arrays (Tables)

Use illustrations to explain

J. Files

1. <u>File declaration</u>

   A file name is declared for each data set and given attributes that describe the data set and the manner in which it will be handled.

2. Standard file

   SYSIN-SYSPRINT

3. Stream-oriented transmission

   Stream-oriented transmission: the data is considered to be a continuous stream of data items, in character form, to be assigned from the stream to variables, or from variables into the stream.

   Three modes of stream-oriented transmission:

   a. List-directed--list-directed data transmission permits the programmer to specify the variables to which data is to be assigned (or from which data is to be acquired) without specifically stating a format for the data.

      GET FILE (file name) LIST (data list);

      PUT FILE (file name) LIST (data list);

   b. Data-directed--data-directed data transmission

      GET FILE (file name) DATA; may have a data list

      PUT FILE (file name) DATA (data list);

      Data items must be declared.

   c. Edit-directed--edit-directed data transmission allows a programmer to specify the format of data as it appears in the stream to be read and have it will appear when a data item is written.

      GET FILE (file name) EDIT (data list) (format list);

      PUT FILE (file name) EDIT (data list) (format list);

K. On Conditions

These are conditions given by the programmer to override standard system action.

L. Build-In Function

Functions provided by PL/1 as part of the compiler.

EX.: SUBSTR
MATHEMATICAL

M. Develop Sample Program

N. Sample Program

The purpose of the program is to compute the return on the number of dollars invested at a certain rate where compound interest may be paid daily, monthly, quarterly, semiannually, or annually.

The basic expressions to be used are:

At interest compounded annually the present value is:

$$A = P(1 + i)^n$$

P = Principle

A = Amount

i = interest rate

n = number of years

At interest compounded x times per year is

$$A = P(1 + \frac{i}{x})^{nx}$$

Input to the program will consist of a card containing the interest rate, principal placed at interest, the number of years involved.

Output from the program will be a list of the input quantities and the present value.

O. Job Control (JCL)

1. 'JOB' Card Format - definition of a job to the operating system

```
//jobname JOB accounting info,'programmer name;
   class=value,
//      Region=value, MSGLEVEL=(X,Y)
```

2. 'EXEC' Card Format - definition of a step to the operating system

```
                  PGM
//stepname EXEC PROC=      , ADDRSPC=REAL, COND=VALUE,
                  PROCNAME

//    PARM=VALUE', REGION=VALUE, TIME=VALUE
```

3. 'DD' Card Format - definition of files to the operating system

```
//ddname  DD *
            DUMMY
            DCB = (List of attributes),
            DISP = (status, disposition),
            DSN = filename,
            LABEL = value,
            SPACE = value,
            SYSOUT = A,
            UNIT = unit,
            VOL = serial #
```

IV. OPERATIONS AND DE-BUGGING

A. Function of Operations--to receive, organize, and process requests for computer services. This is accomplished in the following manner:

1. Workflow

a. Work requests

b. Computer schedule

c. Priorities

d. Validation

e. Distribution

2. Data Entry

a. Conversion of source data to punched card

b. Verification of punched cards

c. Adherence to schedule

d. Priority

3. Computer Room

a. Power up

b. Power down

c. Initial program load

d. Processing

e. Logs

f. Adherence to schedule

B. Job Sequence--at this time we will attempt to follow a job through all of the above phases. It should be pointed out that this is a production run.

1. Three Types of Computer Requests

a. Development -

New programs

b. Maintenance -

Alteration to existing program

c. Production -

User

C. De-Bugging--programmer assistance to help complete programming

V. TIME SHARING OPERATIONS

A. TSO Overview

Use of computer by several users in what appears to be simultaneous usage

1. Development that made time sharing possible

a. Hardware —

i. Cheaper and faster main storage

ii. Much faster central processing units

iii. Introduction of high speed, high-capacity direct access units

b. Software —

Development of operating systems that were designed to allow shift from one job to another and to keep track of allocation of input/output devices to different jobs.

B. Teleprocessing

1. Define teleprocessing--use of computer from remote locations through time sharing principle and use of communication lines or microwave

2. What made teleprocessing possible in hardware area

a. Time sharing or multiprocessing capability of the computer system

b. Capability of interleaving i/o into computer from many sources on one channel. (Multiplexor)

c. Development of teleprocessing control units and terminals.

d. Development of interfaces or modems between communication equipment and the computer equipment.

3. What made teleprocessing possible in software area

a. Operating systems that allowed multiprocessing

b. Access methods for Tp equipment

c. Terminal support software systems--conversational

C. Areas in which teleprocessing is used

1. Data retrieval

2. Data entry

3. Problem solving

4. Programming

5. Real-time update of files

D. Use of teleprocessing at ISHC

1. Brief history

a. Six 2741 terminals installed in Jan. 1970

i. Two in bridge design

ii. Two in road design

iii. One in materials lab

iv. One in data processing

v. Supported by CPS (conversational programming system). Entire system in memory. Took 268K out of 512K

b. Four additional 2741's in Jan. 1972--ROW, Road Design, Data Processing, District 4

c. Four 2741's in Mar. 1972 to Districts 2, 3, 5, 6

d. Four in programming in Fall 1973

e. Nov. 71 went to ITF as terminal support system prog. product. Ran in 120K swaps in and out of user area

f. Went to TSO winter 1973 full time

2. Current Use

a. Road design (3 terminals)

Trig, TRV, Vertical Curve, Elevations,

Horizontal Curves, Geo, Sliding Block Analysis

b. Bridge Design

Wisconsin Beam Analysis, Pier Analysis, Culvert

Analysis, Influence Lines, Continuous Beam, Etc.

c. Materials

PCM, AMX, PIJ, SD, COR, SIV, Proctors, etc.

d. Right of Way

TV6, DEF, CO1-CO4

e. D.P. Internal Systems

Allocate, catalog, scratch, etc.

f. D.P. Programming

3. TSO System

a. Advantages

b. What made it possible

E. Conversion of Programs written by EIT's for Terminal Use

1. Brief rundown on what type of programs are reasonable for terminal use

2. Conversational I/O--Explain

Revise program I/O

3. Use compiler--link editor

4. Write procedure for allocation of data sets and running of program

VII. EXHIBITS

NAME_____

DATE_____

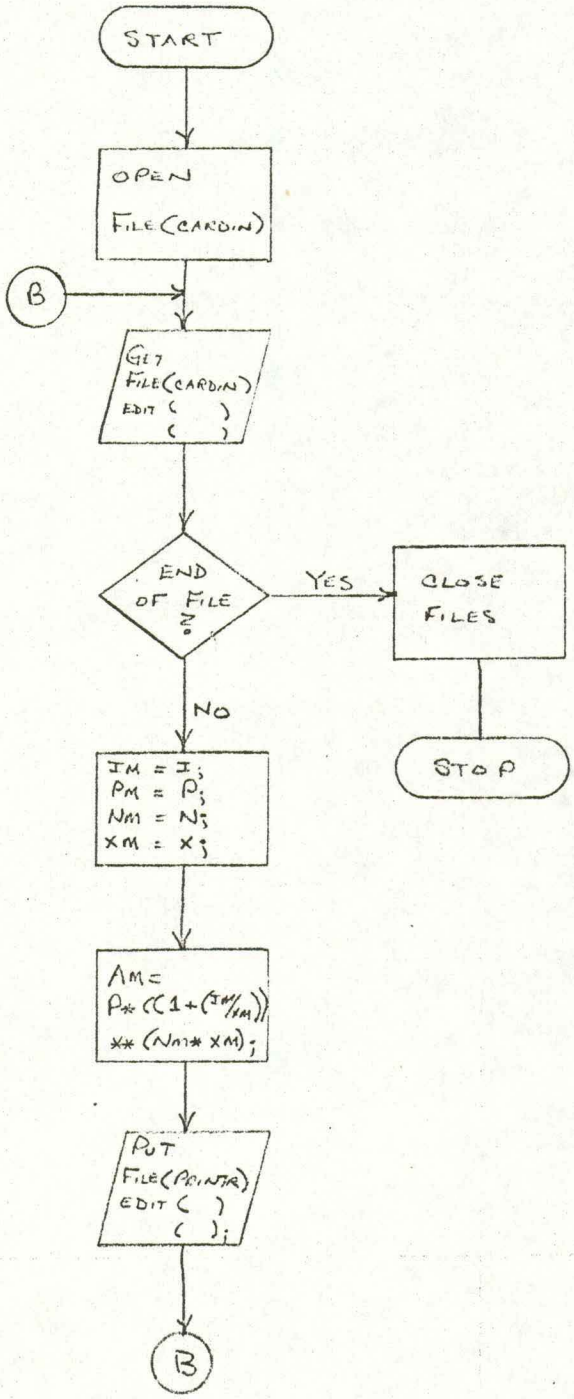
DATA PROCESSING BACKGROUND QUESTIONNAIRE

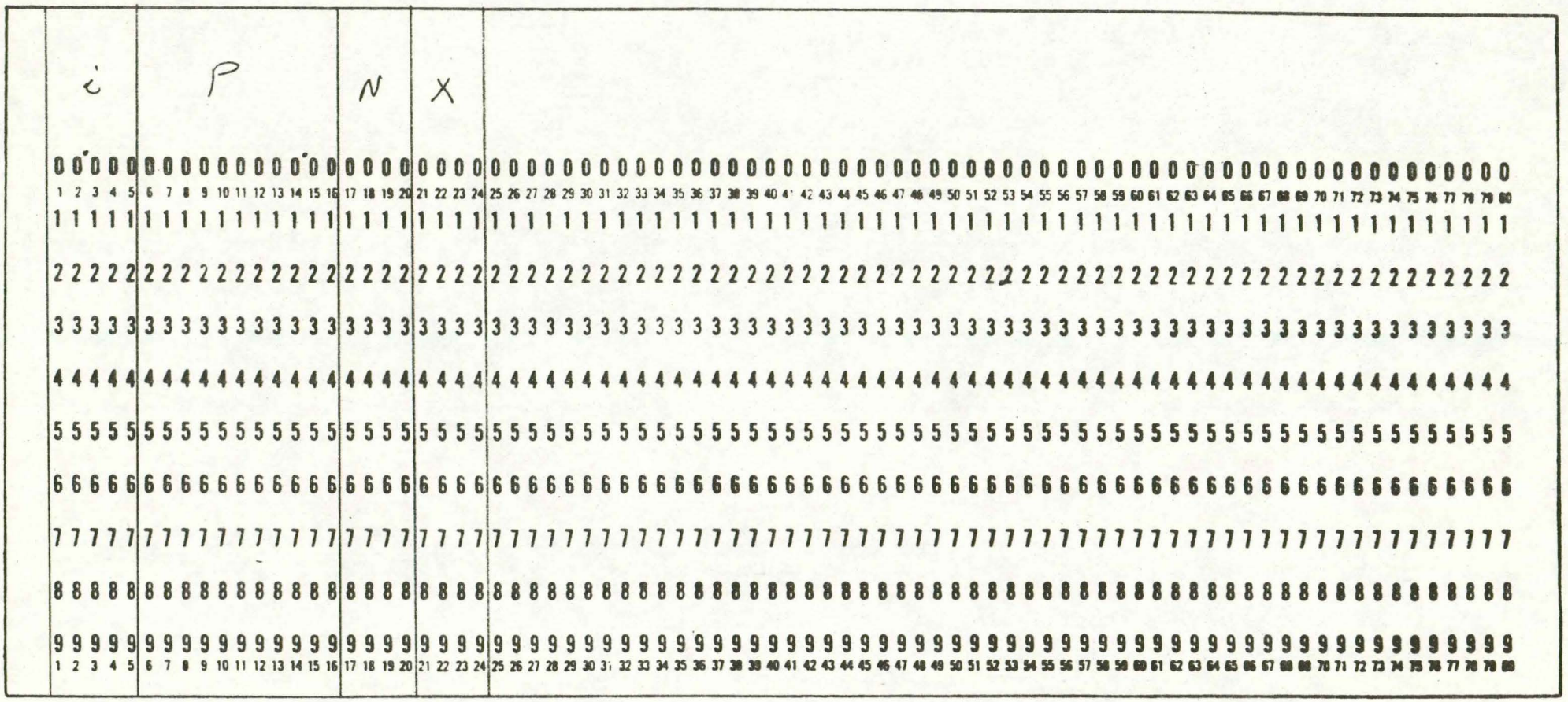
1. Please list all Data Processing course work taken in college.

2. Please briefly explain any data processing experience you may have not included in item one above.

Application PL/I - SAMPLE PROGRAM Date Page 1 of 1

Procedure Drawn By




i
P
N
X

```
//SAMPROG  JOB (11-100,D),'ROGER MCGREGOR',CLASS=E
//S10 EXEC PL1XCLG
XXPL1XCLG  PROC CCORE=192K,SCOND=11,                              00020000
XX              MAC=M,SL=S,EX=NOESD,CSZ=124K,M1=2,M2=72,M3=1,     00040000
XX              LCORE=202K,LCD=9,                                 00060000
XX              PRT=NOLIST,EXQ=LET,LREF=NOXREF,LSIZ=192K,LBUF=64K, 00080000
XX              SYS=SYS1,LIB=PL1BASE,                             00100000
XX              GCORE=64K,GCD=9                                   00120000
XXP_1L     EXEC PGM=IEL0AA,REGION=&CCORE,COND=(&SCOND,LT),       00140000
IEF_53I SUBSTITUTION JCL - PGM=IEL0AA,REGION=192K,COND=(11,LT),
XX PARM='&MAC,&SL,&EX,SZ(&CSZ),MAR(&M1,&M2,&M3)'                  00160000
IEF_53I SUBSTITUTION JCL - PARM='M,S,NOESD,SZ(124K),MAR(2,72,1)'
XXSYSLIB   DD DISP=SHR,DSN=LIB.DECLARE                            00180000
XXSYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LREC_=125,BLKSIZE=1629)     00200000
XXSYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSSQ,           00220000
XX         DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),                   00240000
XX         SPACE=(400,(300,100),RLSE)                             00260000
XXSYSUT1   DD DSN=&&UT1,UNIT=SYSSQ,                               00280000
XX         SPACE=(4096,(250,20),RLSE,CONTIG),DCB=BLKSIZE=4096     00300000
//PL1L.SYSIN DD *
IEF236I ALLOC. FOR SAMPROG  PL1L     S10
IEF237I 152   ALLOCATED TO SYSLIB
IEF237I 151   ALLOCATED TO SYSPRINT
IEF237I 152   ALLOCATED TO SYSLIN
IEF237I 154   ALLOCATED TO SYSUT1
IEF237I 151   ALLOCATED TO SYSIN
```

SOURCE LISTING

```
STMT                                                                    R

      /*PROGRAM # -SAMPROG   NAME -                                    */
      /*PROGRAMMER -ROGER J MCGREGOR      CHG CODE -11-100 DATE -03/26/74  */
    1 SAMPROG: PROC OPTIONS (MAIN);
    2    DCL  CARDIN    FILE      RECORD    INPUT,
              PRINTR    FILE      RECORD    OUTPUT;
    3    DCL  1 CARDREC      STATIC,
                2 I               PIC '99V999',
                2 P               PIC '(9)9V99',
                2 N               PIC '9999',
                2 X               PIC '9999',
                2 FILLER          CHAR(56);
    4    DCL  1 PRINTREC     STATIC,
                2 CC              CHAR(1)        INIT(' '),
                2 OP              PIC '(8)Z9V.99',
                2 FILL1           CHAR(5)        INIT(' '),
                2 OI              PIC 'Z9V.999',
                2 FILL2           CHAR(5)        INIT(' '),
                2 ON              PIC 'ZZZ9',
                2 FILL3           CHAR(5)        INIT(' '),
                2 OX              PIC 'ZZZ9',
                2 FILL4           CHAR(5)        INIT(' '),
                2 OA              PIC '(8)Z9V.99',
                2 FILL5           CHAR(74);
    5    DCL  HEADLINE           CHAR(133);
    6    DCL  (IM, PM, NM, XM, AM)     DEC FLOAT(16)  STATIC    INIT(0);
         /* * * * * * * * */
         /* OPEN FILES */
    7         OPEN FILE(CARDIN), FILE(PRINTR);
         /* * * * * * * * */
         /* ON ENDFILE  */
    8    ON ENDFILE (CARDIN)   GO TO EOJ;
         /* * * * * * * * */
              /* * *    PRINT   HEADING     * * */
    9         HEADLINE = '1 PRINCIPAL      INTEREST   NO. YR   FREQ.        A
   MOUNT';
   10         WRITE FILE(PRINTR) FROM(HEADLINE);
              /***                          *****/
   11 B:
              READ FILE(CARDIN) INTO(CARDREC);
   12         IM = I;
   13         PM = P;
   14         NM = N;
   15         XM = X;
   16         AM = P * ((1 + (IM / XM)) ** (NM * XM));
              /**  **  MOVE  INFORMATION TO RECORD TO BE PRINTED   ** */
   17         CC = '0';
   18         OP = PM;        OI = IM;
   20         ON = NM;        OX = XM;
   22         OA = AM;
   23         WRITE FILE(PRINTR) FROM(PRINTREC);
              /**     **                                          ** */
```

```
STMT                                                              R

  24            GO TO B;
  25  EOJ:
                CLOSE FILE(CARDIN), FILE(PRINTR);
  26            END SAMPROG;
```

ATTRIBUTE AND CROSS-REFERENCE TABLE

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
6	AM	STATIC ALIGNED INITIAL DECIMAL /* DOUBLE */ FLOAT (16) 16,22
11	B	/* STATEMENT LABEL CONSTANT */ 24
2	CARDIN	EXTERNAL FILE RECORD INPUT 7,8,11,25
3	CARDREC	STATIC /* STRUCTURE */ 11
4	CC	/* IN PRINTREC */ STATIC UNALIGNED INITIAL CHARACTER (1) 17
25	EOJ	/* STATEMENT LABEL CONSTANT */ 8
3	FILLER	/* IN CARDREC */ STATIC UNALIGNED CHARACTER (56)
4	FILL1	/* IN PRINTREC */ STATIC UNALIGNED INITIAL CHARACTER (5)
4	FILL2	/* IN PRINTREC */ STATIC UNALIGNED INITIAL CHARACTER (5)
4	FILL3	/* IN PRINTREC */ STATIC UNALIGNED INITIAL CHARACTER (5)
4	FILL4	/* IN PRINTREC */ STATIC UNALIGNED INITIAL CHARACTER (5)
4	FILL5	/* IN PRINTREC */ STATIC UNALIGNED CHARACTER (74)
5	HEADLINE	AUTOMATIC UNALIGNED CHARACTER (133) 9,10
3	I	/* IN CARDREC */ STATIC UNALIGNED PICTURE'99V999' 12
6	IM	STATIC ALIGNED INITIAL DECIMAL /* DOUBLE */ FLOAT (16) 12,16,19
3	N	/* IN CARDREC */ STATIC UNALIGNED PICTURE'9999' 14
6	NM	STATIC ALIGNED INITIAL DECIMAL /* DOUBLE */ FLOAT (16) 14,16,20
4	OA	/* IN PRINTREC */ STATIC UNALIGNED PICTURE'ZZZZZZZZ9V.99' 22

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
4	OI	/* IN PRINTREC */ STATIC UNALIGNED PICTURE'Z9V.999' 19
4	ON	/* IN PRINTREC */ STATIC UNALIGNED PICTURE'ZZZ9' 20
4	OP	/* IN PRINTREC */ STATIC UNALIGNED PICTURE'ZZZZZZZZ9V.99' 18
4	OX	/* IN PRINTREC */ STATIC UNALIGNED PICTURE'ZZZ9' 21
3	P	/* IN CARDREC */ STATIC UNALIGNED PICTURE'999999999V99' 13,16
6	PM	STATIC ALIGNED INITIAL DECIMAL /* DOUBLE */ FLOAT (16) 13,18
2	PRINTR	EXTERNAL FILE RECORD OUTPUT 7,10,23,25
4	PRINTREC	STATIC /* STRUCTURE */ 23
1	SAMPROG	EXTERNAL ENTRY RETURNS(DECIMAL /* SINGLE */ FLOAT (6))
3	X	/* IN CARDREC */ STATIC UNALIGNED PICTURE'9999' 15
6	XM	STATIC ALIGNED INITIAL DECIMAL /* DOUBLE */ FLOAT (16) 15,16,16,21

AGGREGATE LENGTH TABLE

DCL NO.	IDENTIFIER	LVL	DIMS	OFFSET	ELEMENT LENGTH.	TOTAL LENGTH.
3	CARDREC	1			80	80
	I	2			5	
	P	2		5	11	
	N	2		16	4	
	X	2		20	4	
	FILLER	2		24	56	
4	PRINTREC	1			133	133
	CC	2			1	
	0P	2		1	12	
	FILL1	2		13	5	
	0I	2		18	6	
	FILL2	2		24	5	
	0N	2		29	4	
	FILL3	2		33	5	
	0X	2		38	4	
	FILL4	2		42	5	
	0A	2		47	12	
	FILL5	2		59	74	
			SUM OF CONSTANT LENGTHS			213

STORAGE REQUIREMENTS

BLOCK, SECTION OR STATEMENT	TYPE	LENGTH	(HEX)	DSA SIZE	(HEX)
SAMPROG1	PROGRAM CSECT	1024	400		
SAMPROG2	STATIC CSECT	768	300		
SAMPROG	PROCEDURE BLOCK	1024	400	424	1A8
A	ON UNIT	(NO DSA)			

TABLES OF OFFSETS AND STATEMENT NUMBERS

WITHIN PROCEDURE SAMPROG

OFFSET (HEX)	0	8C	96	B6	C6	D8	E2	118	176	1A8	1DA	272	276	2D2	30C
STATEMENT NO.	1	7	8	9	10	11	12	13	14	15	16	17	18	19	20

OFFSET (HEX)	342	378	3D0	3DA	3DE	3E8
STATEMENT NO.	21	22	23	24	25	26

COMPILER DIAGNOSTIC MESSAGES

ERROR ID L STMT MESSAGE DESCRIPTION

WARNING DIAGNOSTIC MESSAGES

IEL0916I W 1 ITEM(S) 'PRINTREC.FILL5','CARDREC.I','PRINTREC.OP','PRINTREC.OI','PRINTREC.ON','PRINTREC.OX', 'PRINTREC.OA','CARDREC.X','CARDREC.FILLER','CARDREC.P','CARDREC.N' MAY BE UNINITIALIZED WHEN USED IN THIS BLOCK.

COMPILER INFORMATORY MESSAGES

IEL0541I I 1 'ORDER' OPTION APPLIES TO THIS BLOCK. OPTIMIZATION MAY BE INHIBITED.

END OF COMPILER DIAGNOSTIC MESSAGES

COMPILE TIME 0.09 MINS SPILL FILE: 0 RECORDS, SIZE 4051

```
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS1.PL1BASE                                    KEPT
IEF285I   VOL SER NOS= VS217R.
IEF285I   SYS1.PL1BASE                                    KEPT
IEF285I   VOL SER NOS= VS217R.
IEF285I   SYS1.PLXPLOT                                    KEPT
IEF285I   VOL SER NOS= VS217L.
IEF285I   SYS1.SUBMODUL                                   KEPT
IEF285I   VOL SER NOS= VS217R.
IEF285I   SYS1.FORTLIB                                    KEPT
IEF285I   VOL SER NOS= VS217R.
IEF285I   SYS74302.T114336.RV000.SAMPROG.GOSET            PASSED
IEF285I   VOL SER NOS= SP00L2.
IEF285I   SYS74302.T114336.RV000.SAMPROG.UT1              DELETED
IEF285I   VOL SER NOS= 444444.
IEF285I   SYS74302.T114336.SV000.SAMPROG.R0000045         SYSOUT
IEF285I   VOL SER NOS= SP00L1.
IEF285I   SYS74302.T114336.RV000.SAMPROG.LOADSET          DELETED
IEF285I   VOL SER NOS= 444444.
IEF373I STEP /LKED    / START 74302.1218
IEF374I STEP /LKED    / STOP  74302.1219 CPU   0MIN 01.39SEC STOR VIRT 200K
XXGO       EXEC PGM=*.LKED.SYSLMOD,REGION=&GCORE,                    00600000
IEF653I SUBSTITUTION JCL - PGM=*.LKED.SYSLMOD,REGION=64K,
XX         COND=((&LCD,LT,PL1L),(&GCD,LT,LKED))                      00620000
IEF653I SUBSTITUTION JCL - COND=((9,LT,PL1L),(9,LT,LKED))
XXSYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=1648)        00640000
XXSORTMSG  DD SYSOUT=A,SPACE=(TRK,(2,3),RLSE)                       00660000
XXSYSUDUMP DD SYSOUT=A,SPACE=(CYL,(0,5),RLSE)                       00680000
//GO.PRINTR    DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1729)
//GO.CARDIN    DD *
IEF236I ALLOC. FOR SAMPROG  GO       S10
IEF237I 152   ALLOCATED TO PGM=*.DD
IEF237I 151   ALLOCATED TO SYSPRINT
IEF237I 154   ALLOCATED TO SORTMSG
IEF237I 152   ALLOCATED TO SYSUDUMP
IEF237I 153   ALLOCATED TO PRINTR
IEF237I 151   ALLOCATED TO CARDIN
```

PRINCIPAL	INTEREST	NO. YR	FREQ.	AMOUNT
1000.00	0.049	1	4	1050.94
1000.00	0.054	1	4	1056.14
1000.00	0.059	1	4	1061.36
1000.00	0.064	1	4	1066.60