DECUS

PROGRAM LIBRARY

DECUS NO.	8-504B
TITLE	ESIX - EXTENDED ESI
AUTHOR	Robert Supnik
COMPANY	Applied Data Research Cambridge, Massachusetts
DATE	1967
SOURCE LANGUAGE	MACRO-8X

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

ESIX - EXTENDED ESI

DECUS Program Library Write-up DECUS NO. 8-504B

ESIX is an enhanced version of ESI (Engineering and Scientific Interpreter) for the PDP-8 family of computers. ESIX runs on any PDP-8 family machine with 8K or more of core and a teletype. Compared with basic ESI, it offers four times as much program and array storage, many extended functions, paginated output, generalized exponentiation, and a number of new statement types.

This manual details the extended features of ESIX and the revised operating procedures used to lead and start ESIX. It is not, however, a guide to the ESI language and environment and must be read in conjunction with the explanatory literature on basic ESI. In particular, it is assumed that the reader is familiar with the Introduction to ESI and related documents.*

1. New Features

1.1 Extended array and program storage. ESIX offers over 3800 words of free storage, as compared to about 800 words for basic ESI. Therefore, ESIX can store over 7500 characters of program, or over 600 array elements, or an intermediate combination of the two. For example, ESIX can store both the program and arrays required to solve a 7 x 7 zero sum game (see example program at end).

At any time, the user can find out how much free storage remains by typing the command
←TYPE SIZE.
ESIX responds by typing out the number of free words remaining, initially 3872. TYPE SIZE can be given as either a direct or an indirect statement.

1.2 Extended functions. ESIX contains the following additional built-in numeric functions:

SQRT(N) - square root
SIN(N) - sine, argument in radians
COS(N) - cosine, argument in radians
ARCTAN(N) - arctangent, answer in radians
EXP(N) - exponentiation, answer is e^N
LOG(N) - common logarithm
LN(N) - natural logarithm
DP(N) - digit part. The digit part of 123.456 is 1.23456, of .00004231 is 4.231
XP(N) - exponent part. The exponent part of 123.456 is 2, of .00004231 is -5. For any number, N=DP(N) * 10 ↑ XP(N)

* DECUS NO. 8-504A

ESIX also includes all the functions built into basic ESI:

ABS(N) - absolute value
IP(N) - integer part. The integer part of 123.456 is 123, of .00004231 is 0.
FP(N) - fraction part. The fraction part of 123.456 is .456, of .00004231 is .00004231
SGN(N) - sign part. The sign part of +3 is 1, of -4 is -1.

1.3 Generalized Exponentiation. In basic ESI, exponents had to be integers in the range 0-9. In ESIX, exponents can be arbitrary arithmetic expressions. For example:

```
←TYPE "THE CUBE ROOT OF "I" IS" I ↑ (1/3).
```

Integer exponents in the range 1-9 are handled by repetitive multiplication. All other exponents are handled by the expansion:

$$A^B = \mathrm{EXP}(B*\mathrm{LN}(A))$$

1.4 Compound statements. More than one statement can be placed on the same line, each statement separated by a semicolon:

```
←TYPE A; SET A = SQRT(A); TYPE A.
```

would result in: (if A = 4)

```
A =    4
A =    2
```

Statements on the same line joined by a semi-colon are called compound statements. A compound statement is for all intents and purposes a single statement. Compound statements are treated by FOR clauses, IF clauses, and DO statements just like simple statements. For example:

```
←FOR I = 1(1)10, SET A[I] =0; SET B[I] = I.
```

sets $A_1,\ldots,A_{10}$ to zero and $B_1,\ldots,B_{10}$ to $\{1,\ldots,10\}$
Or again

```
←FOR I = 1(1)10, DEMAND A[I]; DEMAND B[I].
```

asks first for A_1, then B_1, then A_2, then B_2, etc.
If step 4.2 is

```
4.2   SET F = 1; SET G = 1E10.
```

then

```
←DO STEP 4.2.
```

defines both F and G. The programming, example at the end demonstrates the uses of compound statements.

1.4 Comments

Lines beginning with * are treated as comments and ignored:

```
←*THIS IS A COMMENT.
```

Comments may be used as indirect statements:

```
1.0001   *THIS PROGRAM SOLVES ALL PROBLEMS
1.0002   *TO OPERATE, SIMPLY PLACE 25 CENTS NEAR
```

1.5 Pagination. ESIX automatically paginates its output into 8 1/2 x 11 pages, provides tear marks, types out a heading and page numbers. To use the pagination mechanism the user should type

```
←LOG ON.
```

ESIX will respond by asking for the title of the upcoming session

```
TITLE:←
```

After the user provides a title, ESIX will space to the top of a new page and await further input.

At the end of his session, the user should terminate by typing

```
←LOG OFF.
```

ESIX will space to the end of page and type a final set of tear marks to delimit the user's output.

At any time, the user can force ejection to the beginning of a new page by typing

```
←PAGE.
```

PAGE can be either a direct or indirect statement. Used in the latter form, it is extremely useful in formulating reports and in teletype I/O.

2. Operating Instructions

2.1 Use the RIM loader to load the binary loader into field 1. To do this:

- Place the tape of the binary loader in the paper-tape reader.
- Set the instruction field switches to 0, the data field switches to 1.
- Start the RIM loader at 7756.

2.2 When the binary loader is successfully loaded into field 1, start it at 17777 and use it to load ESIX. To do this:

- Place the binary tape of ESIX in the paper-tape reader.
- Set the instruction and data field switches to 1.
- Set up a starting address of 7777.
- Set up the console switches to select the low-speed or high-speed paper-tape reader.
- Start the BIN loader.

2.3 When the BIN loader halts, the AC should be 0000. If it is not, a check sum error has occurred; step 2.2 should be repeated. ESIX starts at location 5400, the same location as PDP-8 DDT.

PERMISSIBLE FORMS IN ESI-X, COPYRIGHT 1967, APPLIED DATA RESEARCH, INC.

DIRECT OR INDIRECT:

```
SET C = A*B+C*D.
SET C[I,J] = B[I-1,J↑2]-C[I+1,J/2].
SET Y = IP(X/I).
FOR I = 1(1)N, SET A[I] = B[I]*C[I].

DO PART 3.
FOR R = Ø(Ø.1)1.5, DO PART 2.
DO STEP 3.7.
FOR J = N(-1)1, DO STEP 7.352.

TYPE 2+3+5.
TYPE X.
TYPE X, IP(X), SGN(X), ARCTAN(X).
FOR I = 1(1)N, TYPE A[I].
TYPE "THIS IS A STRING".
TYPE "THE SQUARE OF" X "IS" X↑2.
TYPE STEP 2.3.
TYPE PART 6.
TYPE ALL PARTS.
TYPE ALL VALUES.
TYPE ALL A.
TYPE ALL.

DELETE X.
DELETE A[1,3], B[I,J], C, D.
DELETE ALL VALUES.
FOR I=1(1)N, DELETE A[I].

LINE.
PAGE.

*THIS IS A COMMENT
```

DIRECT ONLY:

```
DELETE STEP 1.1.
DELETE PART 2.
DELETE ALL PARTS.
DELETE ALL.

X = 3*ABS(A-B)
Y=Z+R
Z=14
A[I]=1.3E-6
A[43] = 1.414E+32
A[2] = 3175*I

LOG ON.
LOG OFF.
```

INDIRECT ONLY:

```
1.1 TO STEP 1.7.
1.7 TO PART 4.

2.3 END.

4.1 STOP.

6.1 DEMAND X.
7.35 DEMAND A[I,J].
8.1 DEMAND A[45].
```

CONDITIONAL CLAUSES:

```
IF A = B,
IF ABS(N-O/N) < 1E-6,
IF LOG(X) GE LN(Y),
IF EXP(X) NE 1,
IF (A-B)/C LE D-X↑5,
```

FUNCTIONS:

IP(X)	INTEGER PART
FP(X)	FRACTION PART
SGN(X)	SIGN PART
ABS(X)	ABSOLUTE VALUE
DP(X)	DIGIT PART

NUMBERS:

```
2
3.141593
.ØØ3
Ø.Ø1
-3.7E5
4.36E-7
-3.273E+17
```

INTERRUPTED OR STOPPED:

```
ANY "TYPE" STATEMENT.
GO.
CANCEL.
```

OPERATIONS:

```
+ - * / ↑ ()
```

RELATIONS:

```
< > = GE LE NE
```

XP(X)	EXPONENT PART
SQRT(X)	SQUARE ROOT
SIN(X)	SINE
COS(X)	COSINE
LOG(X)	COMMON LOGARITHM
LN(X)	NATURAL LOGARITHM
EXP(X)	EXPONENTIAL
ARCTAN(X)	ARCTANGENT

← IS TYPED AND THE BELL RINGS WHENEVER A USER TYPE-IN IS REQUESTED.
[AND] ARE USED TO DENOTE SUBSCRIPTS.
? TYPED AT THE END OF ANY LINE CAUSES IT TO BE DISREGARDED.
"RUBOUT" DELETES THE PRECEDING CHARACTER AND TYPES ← TO SO INDICATE.
STEP NUMBERS ARE IN THE RANGE 1 TO 9.999999.
VARIABLES ARE THE SINGLE LETTERS A THROUGH Z.
"ALT MODE" INTERRUPTS EXECUTION OF A PROGRAM AT THE COMPLETION OF THE CURRENT STEP; ON A "DEMAND", "ALT MODE" CANCELS EXECUTION.
THE TELETYPE LISTING IS AUTOMATICALLY PAGINATED. "LOG ON" CAUSES A REQUEST FOR THE USER TO TYPE IN THE TITLE WHICH IS TO APPEAR AT THE TOP OF EVERY PAGE.

Most ESI statements may be compounded using semi-colons. Such compound statements are executed as single units:

```
         SET C = A+B; SET E = 2.71828.
     1.2 DEMAND X; DEMAND Y; TO PART 3.
         TYPE STEP 2.3; TYPE PART 4; PAGE; *THIS DUMPS USER MEMORY.
    4.12 TYPE ALL A; STOP.
     6.5 DO PART 2; DO STEP 3.1; DO STEP 6.6.
```

The range of an 'IF' or 'FOR' clause in a compound statement extends to the end of the entire statement, that is, to the terminating period; for example,

```
         IF A = 4, TYPE "YES"; TYPE "       "A.
```

results in (if A = 4)

```
         YES
         4
```

and nothing if A ≠ 4; and

```
     1.3 FOR I = 1(1)2, TYPE I; TYPE SQRT(I).
```

results in

```
        I = 1
  SQRT(I) = 1
        I = 2
  SQRT(I) = 1.414214
```

Certain compounds which do not make logical sense are forbidden:

```
     1.4 TO STEP 9.77; any statement.
    4.22 END; any statement.
         LOG ON; any statement.
         LOG OFF; any statement.
         any statement; LOG ON.
         any statement; LOG OFF.
```

```
*PAGE 8      GAME THEORY

←TYPE ALL PARTS.

 1.ØØ1 *.
 1.ØØ2 *SINGLE PHASE TWO PERSON ZERO SUM GAME SOLVER.
 1.ØØ3 *RMS -- ADR, 9/7/67.
 1.1 DEMAND M; DEMAND N; FOR I=1(1)M, FOR J=1(1)N, DEMAND P[I,J].
 1.3 FOR I=1(1)M, SET Q[I,N+1]=1; SET B[I]=I; SET C[I]=Ø.
 1.35 FOR J=1(1)N, SET Q[M+1,J]=-1; SET D[J]=J; SET A[J]=Ø.
 1.36 SET S=1E1Ø.
 1.37 FOR I=1(1)M, FOR J=1(1)N, IF P[I,J]<S, SET S=P[I,J].
 1.38 IF S>Ø, SET S=Ø.
 1.4 FOR I=1(1)M, FOR J=1(1)N, SET Q[I,J]=P[I,J]-S.
 1.45 SET Q[M+1,N+1]=Ø.
 1.5 SET Z=1; SET Y=-1; SET K=1; SET L=1.
 1.53 FOR J=1(1)N, IF Q[M+1,J] LE Ø, DO PART 4.
 1.54 SET Y=Q[K,L]; SET R[K,L]=Z.
 1.6 FOR I=1(1)M+1, DO PART 6.
 1.61 FOR I=1(1)M+1, FOR J=1(1)N+1, SET Q[I,J]=R[I,J].
 1.7 SET T=A[L]; SET A[L]=B[K]; SET B[K]=T.
 1.73 SET T=C[K]; SET C[K]=D[L]; SET D[L]=T.
 1.8 SET Z=Y.
 1.85 FOR I=1(1)M, IF Q[I,N+1] < Ø, TO STEP 1.5.
 1.86 FOR J=1(1)N, IF Q[M+1,J] < Ø, TO STEP 1.5.
 1.88 FOR I=1(1)M, SET B[I]=Ø.
 1.89 FOR J=1(1)N, SET D[J]=Ø.
 1.9 FOR I=1(1)M, FOR K=1(1)N, IF I=A[K], SET B[I]=Q[M+1,K].
 1.91 FOR J=1(1)N, FOR K=1(1)M, IF J=C[K], SET D[J]=Q[K,N+1].
 1.92 SET Z=Z/Q[M+1,N+1]+S.
 1.93 DO PART 9; TO STEP 1.1.

 4.1 SET F=1; SET G=1E1Ø.
 4.3 FOR I=1(1)M, IF Q[I,J] > Ø, DO PART 5.
 4.4 IF G LE Y, END.
 4.5 SET Y=G; SET K=F; SET L=J.

 5.1 SET V=(-Q[I,N+1]*Q[M+1,J])/Q[I,J];IF V LE G,SET G=V; SET F=I.

 6.1 IF I NE K, TO STEP 6.4.
 6.2 FOR J=1(1)N+1, IF J NE L, SET R[I,J]=Q[I,J].
 6.3 END.
 6.4 FOR J=1(1)N+1,IF J NE L,SET R[I,J]=(Q[I,J]*Y-Q[I,L]*Q[K,J])/Z.
 6.5 SET R[I,L]=-Q[I,L].

 9.1 FOR I=1(1)5,LINE.
 9.15 TYPE "VALUE OF GAME IS "Z.
 9.2 LINE; TYPE "RED ODDS:"; FOR J=1(1)N, TYPE "     "D[J].
 9.5 LINE; TYPE "BLUE ODDS:"; FOR I=1(1)N, TYPE "     "B[I].
 9.8 LINE;TYPE "GAME MATRIX:".
 9.85 FOR I=1(1)M, FOR J=1(1)N, TYPE "     "P[I,J].
 9.9 PAGE.
←F-LOG OFF.
```