## APPENDIX A: HARDWARE REPRESENTATION AND COLLATING SEQUENCE FOR THE LANGUAGE CHARACTER SET

<table>
<thead>
<tr>
<th>SIMULA symbol</th>
<th>VALUE</th>
<th>Punched card code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>decimal</td>
<td>hexadecimal</td>
</tr>
<tr>
<td>&lt;</td>
<td>75</td>
<td>4B</td>
</tr>
<tr>
<td>(</td>
<td>76</td>
<td>4C</td>
</tr>
<tr>
<td>+</td>
<td>77</td>
<td>4D</td>
</tr>
<tr>
<td>&amp;</td>
<td>78</td>
<td>4E</td>
</tr>
<tr>
<td>$</td>
<td>80</td>
<td>50</td>
</tr>
<tr>
<td>*</td>
<td>91</td>
<td>5B</td>
</tr>
<tr>
<td>)</td>
<td>92</td>
<td>5C</td>
</tr>
<tr>
<td>;</td>
<td>93</td>
<td>5D</td>
</tr>
<tr>
<td>-</td>
<td>94</td>
<td>5E</td>
</tr>
<tr>
<td>/</td>
<td>95</td>
<td>5F</td>
</tr>
<tr>
<td>,</td>
<td>96</td>
<td>60</td>
</tr>
<tr>
<td>)</td>
<td>97</td>
<td>61</td>
</tr>
<tr>
<td>,</td>
<td>107</td>
<td>6B</td>
</tr>
<tr>
<td>&lt;</td>
<td>109</td>
<td>6D</td>
</tr>
<tr>
<td>&gt;</td>
<td>110</td>
<td>6E</td>
</tr>
<tr>
<td>:</td>
<td>122</td>
<td>7A</td>
</tr>
<tr>
<td>#</td>
<td>123</td>
<td>7B</td>
</tr>
<tr>
<td>'</td>
<td>125</td>
<td>7D</td>
</tr>
<tr>
<td>=</td>
<td>126</td>
<td>7E</td>
</tr>
<tr>
<td>&quot;</td>
<td>127</td>
<td>7F</td>
</tr>
<tr>
<td>A</td>
<td>193</td>
<td>C1</td>
</tr>
<tr>
<td>B</td>
<td>194</td>
<td>C2</td>
</tr>
<tr>
<td>C</td>
<td>195</td>
<td>C3</td>
</tr>
<tr>
<td>D</td>
<td>196</td>
<td>C4</td>
</tr>
</tbody>
</table>

- no punch
- 12 - 3 - 8
- 12 - 4 - 8
- 12 - 5 - 8
- 12 - 6 - 8
- 12
- 11 - 3 - 8
- 11 - 4 - 8
- 11 - 5 - 8
- 11 - 6 - 8
- 11 - 7 - 8
- 11
- 0 - 1
- 0 - 3 - 8
- 0 - 5 - 8
- 0 - 6 - 8
- 2 - 8
- 3 - 8
- 5 - 8
- 6 - 8
- 7 - 8
- 12 - 1
- 12 - 2
- 12 - 3
- 12 - 4
<table>
<thead>
<tr>
<th>SIMULA symbol</th>
<th>VALUE</th>
<th>Punched card code</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>197</td>
<td>12 - 5</td>
</tr>
<tr>
<td>F</td>
<td>198</td>
<td>12 - 6</td>
</tr>
<tr>
<td>G</td>
<td>199</td>
<td>12 - 7</td>
</tr>
<tr>
<td>H</td>
<td>200</td>
<td>12 - 8</td>
</tr>
<tr>
<td>I</td>
<td>201</td>
<td>12 - 9</td>
</tr>
<tr>
<td>J</td>
<td>209</td>
<td>11 - 1</td>
</tr>
<tr>
<td>K</td>
<td>210</td>
<td>11 - 2</td>
</tr>
<tr>
<td>L</td>
<td>211</td>
<td>11 - 3</td>
</tr>
<tr>
<td>M</td>
<td>212</td>
<td>11 - 4</td>
</tr>
<tr>
<td>N</td>
<td>213</td>
<td>11 - 5</td>
</tr>
<tr>
<td>O</td>
<td>214</td>
<td>11 - 6</td>
</tr>
<tr>
<td>P</td>
<td>215</td>
<td>11 - 7</td>
</tr>
<tr>
<td>Q</td>
<td>216</td>
<td>11 - 8</td>
</tr>
<tr>
<td>R</td>
<td>217</td>
<td>11 - 9</td>
</tr>
<tr>
<td>S</td>
<td>226</td>
<td>0 - 2</td>
</tr>
<tr>
<td>T</td>
<td>227</td>
<td>0 - 3</td>
</tr>
<tr>
<td>U</td>
<td>228</td>
<td>0 - 4</td>
</tr>
<tr>
<td>V</td>
<td>229</td>
<td>0 - 5</td>
</tr>
<tr>
<td>W</td>
<td>230</td>
<td>0 - 6</td>
</tr>
<tr>
<td>X</td>
<td>231</td>
<td>0 - 7</td>
</tr>
<tr>
<td>Y</td>
<td>232</td>
<td>0 - 8</td>
</tr>
<tr>
<td>Z</td>
<td>233</td>
<td>0 - 9</td>
</tr>
<tr>
<td>0</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>241</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>242</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>243</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>244</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>245</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>246</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>247</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>248</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>249</td>
<td>9</td>
</tr>
</tbody>
</table>
N.B. Characters are compared by their value (column 2 or column 3).

The initial value of CHARACTER variables is 0 (punched card code 12-0-1-8-9).

The data character '\' has punched card code 12, the data character '-' (minus) has punch card code 11.
APPENDIX B: THE SYSTEM DEFINED PROCEDURES

Calls to system defined procedures conform to the syntax of calls to declared procedures. The identifier of a standard procedure may be redefined to have another meaning at any block level. The identifier then assumes the new meaning throughout the scope of the block. Standard procedures are available to any SIMULA program.

The system defined procedures detailed below are grouped into the following sections

- Arithmetic functions
- CHARACTER handling
- Random drawing procedures
- Utility procedures

For details of other procedures:

- for TEXT handling see Part 3, section 4
- for Sequencing procedures see Part 3, section 1
- for Procedures local to SIMSET see Part 3, section 2
- for Procedures local to SIMULATION see Part 3, section 3
- for Procedures local to subclasses of CLASS file see Part 3, section 5

A skeleton outline of all system defined procedures and classes is given as APPENDIX C.
Arithmetic functions

Certain identifiers, expressed as procedures are defined by the Simula system for standard arithmetic functions.

ABS(E)  modulus (absolute value) of E

ARCCOS(E) \{ return the principal values of the
ARCSIN(E) \} arc-cosine, arc-sine, arc-tangent of
ARCTAN(E) \ E (E is measured in radians)

COS(E) \ return the cosine, sine tangent of
SIN(E) \ E (E is measured in radians)
TAN(E) \}

COSH(E) \ return the hyperbolic cosine, hyperbolic
SINH(E) \ sine, hyperbolic tangent of E (E is
TANH(E) \ measured in radians)

EXP(E) \ exponential function of E \(e^E\)

LN(E) \ natural logarithm of E (log_2E, or ln E).
If E <= 0, a run time error results.

SQRT(E) \ returns the square root of E if E >= 0.
If E < 0, a run time error results.

The above 13 functions operate on arithmetic arguments. If the type of E is [SHORT]INTEGER or REAL, then the function value is of type REAL. If the type of E is LONG REAL, then the function value is of type LONG REAL.
SIGN(E)  sign of the value of E
        \[
        \begin{cases} 
        1 & \text{if } E > 0 \\
        0 & \text{if } E = 0 \\
        -1 & \text{if } E < 0 
        \end{cases}
        \]

ENTIER(E)  largest whole number less than or equal to E
            (equal to or to the left of E on the real axis).
            e.g.  \( \text{ENTIER}(5.3) = 5 \)
            \( \text{ENTIER}(-4.7) = -5 \)
            \( \text{ENTIER}(1) = 1 \)

The above 2 functions operate upon [LONG] REAL or [SHORT] INTEGER
values of E and yield values of type [SHORT] INTEGER depending
upon their magnitude.
MOD(M,N)  M modulo N, that is
          M-ENTIER(M/N)*N

        e.g. MOD (7,3)   is 1
             MOD (-48,5) is 2

The function operates on [SHORT]INTEGER arguments, [LONG]REAL arguments being rounded. The result is [SHORT]INTEGER.
CHARACTER handling

Two CHARACTER subsets are defined by the standard procedures:

BOOLEAN PROCEDURE DIGIT(C); CHARACTER C;

which is TRUE if C is a digit, FALSE otherwise.

BOOLEAN PROCEDURE LETTER(C); CHARACTER C;

which is TRUE if C is a capital letter, FALSE otherwise.

The collating sequence defines a one-one mapping between the INTEGERS and the internal CHARACTER representation (see APPENDIX A).

INTEGER PROCEDURE RANK(C); CHARACTER C;

returns a value in the range 0 through 255.

CHARACTER PROCEDURE CHAR(N); INTEGER N;

if the parameter value is not in the range 0 through 255, a run time error results. Otherwise the procedure returns the CHARACTER with value N.

Examples:

RANK('+') = 78
RANK('A') = 193
CHAR(249) = '9'
CHAR(#40) = '٠'
RANK(CHAR(127)) = 127
DIGIT(C) = RANK(C) >= 240 AND RANK(C) <= 249
DIGIT(C) = C >= '0' AND C <= '9'
LETTER(C) = (C >= 'A' AND C <= 'I')
            OR (C >= 'J' AND C <= 'R')
            OR (C >= 'S' AND C <= 'Z')
Random drawing procedures

The random drawing procedures produce, in successive calls, a stream of random numbers taken from a specified distribution. As a side effect, the procedures update the stream variable U (which must always be an INTEGER variable) thus advancing the specified stream by one step.

**BOOLEAN PROCEDURE DRAW(A,U); NAME U; REAL A; INTEGER U;**

If 0 < A < 1, the value is TRUE with probability A, FALSE with probability 1-A.

If A <= 0, the value is always FALSE.

If A >= 1, the value is always TRUE.

**INTEGER PROCEDURE RANDINT(A,B,U); NAME U; INTEGER A,B,U;**

If A <= B, the value is one of the INTEGERS A, A+1, A+2, ..., B-1, B

with equal probability.

If A > B, a run time error results.

**REAL PROCEDURE UNIFORM(A,B,U); NAME U; REAL A,B; INTEGER U;**

If A < B, the value is uniformly distributed between A and B.

If A >= B, a run time error results.
REAL PROCEDURE NORMAL(A,B,U); NAME U; REAL A,B; INTEGER U;

The value is normally distributed with mean "A" and standard deviation "B". An approximation function is used for the normal distribution.

REAL PROCEDURE NEGEXP(A,U); NAME U; REAL A; INTEGER U;

If A > 0, the value is drawn from a negative exponential distribution with mean "1/A". This is the same as a random waiting time in a Poisson distributed arrival pattern with an expected number of arrivals per time unit equal to "A".

If A <= 0, a run time error results.

INTEGER PROCEDURE POISSON(A,U); NAME U; REAL A; INTEGER U;

The value is a drawing from the Poisson distribution with parameter "A".

If A > 20.0, the value is approximated by

\[ \text{ENTIER}(0.5 + \text{NORMAL}(A,\sqrt{A}),U) \]

If A < 0, the value is zero.

REAL PROCEDURE ERLANG(A,B,U); NAME U; REAL A,B; INTEGER U;

If A <= 0 or B <= 0, a run time error results.

If A > 0 and B > 0, then the value is a drawing from the Erlang distribution with mean "1/A" and standard deviation "1/(A*\sqrt{B})".
INTEGER PROCEDURE DISCRETE(A,U); NAME U; REAL ARRAY A;
    INTEGER U;

The one dimensional REAL ARRAY A, augmented by a unit element one to the right, is interpreted as a step function of the subscript, defining a discrete (cumulative) distribution function. The function value is an INTEGER in the range

"A.lower bound" through "A.upper bound+1"

It is defined as the smallest I such that

A(I) > U

where U is a basic drawing and A(upper bound+1) = 1.0.

REAL PROCEDURE LINEAR(A,B,U); NAME U; REAL ARRAY A,B;
    INTEGER U;

The value is a drawing from a cumulative distribution function F, which is obtained by linear interpolation in a non-equidistant table defined by A and B, such that

A(I) = F(B(I))

It is assumed that A and B are one dimensional REAL ARRAYS with the same bounds, that the first and last elements of A are equal to 0.0 and 1.0 respectively, and that

\[
\begin{align*}
A(I) &\geq A(J) \\
B(I) &\geq B(J)
\end{align*}
\]

for I > J

INTEGER PROCEDURE HISTD(A,U); NAME U; REAL ARRAY A;
    INTEGER U;

The value is an INTEGER in the range "A.lower bound" through "A.upper bound", where A is a one dimensional REAL ARRAY interpreted as a histogram defining the relative frequencies of values.
Utility procedures

PROCEDURE HISTO(A,B,C,D); {INTEGER} ARRAY A,B;
{REAL} C,D;

A call on HISTO updates a histogram defined by the one dimensional ARRAYS (INTEGER or REAL) A,B according to observation C with weight D. A(I) is incremented by D, where J is the smallest INTEGER such that C <= B(I). It is assumed that the length of A is one greater than the length of B. The last element of A corresponds to those observations which are greater than all the elements of B.

PROCEDURE LOWTEN(C); CHARACTER C;

Without use of LOWTEN, the CHARACTER 'E' represents the exponent sign in any numeric item to be edited or de-edited. A call on "LOWTEN" with actual parameter "EXPSIGN" will replace 'E' by the value of EXPSIGN in future editing and de-editing.
APPENDIX C: SKELETON OF THE SYSTEM CLASSES AND SYSTEM PROCEDURES

Contents:

SYSTEM DEFINED PROCEDURES
ARITHMETIC FUNCTIONS
CHARACTER HANDLING
TEXT HANDLING
SEQUENCING PROCEDURES
RANDOM DRAWING PROCEDURES
UTILITY PROCEDURES

SYSTEM DEFINED CLASSES
CLASS SIMSET
CLASS SIMULATION
SUBCLASSES of file

PROGRAM ENVIRONMENT
SYSTEM DEFINED PROCEDURES

ARITHMETIC FUNCTIONS

REAL PROCEDURE ABS(X); REAL X;
REAL PROCEDURE ACOSH(X); REAL X;
REAL PROCEDURE ACOS(X); REAL X;
REAL PROCEDURE ARCCOSH(X); REAL X;
REAL PROCEDURE ARCOS(X); REAL X;
REAL PROCEDURE ATAN(X); REAL X;
REAL PROCEDURE COSH(X); REAL X;
REAL PROCEDURE SINH(X); REAL X;
REAL PROCEDURE TANH(X); REAL X;
REAL PROCEDURE SQRT(X); REAL X;
REAL PROCEDURE TANH(X); REAL X;
REAL PROCEDURE TANH(X); REAL X;

N.B. If the actual parameter on a call to any of the above procedures is LONG REAL, then a LONG REAL result will be returned.

INTEGER PROCEDURE ENTIER(X); REAL X;
INTEGER PROCEDURE MOD(X,Y); INTEGER X,Y;
INTEGER PROCEDURE SIGN(X); REAL X;
CHARACTER HANDLING

BOOLEAN PROCEDURE DIGIT(C); CHARACTER C;
BOOLEAN PROCEDURE LETTER(C); CHARACTER C;
CHARACTER PROCEDURE CHAR(N); INTEGER N;
INTEGER PROCEDURE RANK(C); CHARACTER C;

TEXT HANDLING

TEXT OBJECT GENERATION

TEXT PROCEDURE COPY(T); VALUE T; TEXT T;
TEXT PROCEDURE BLANKS(N); INTEGER N;

TEXT ATTRIBUTES

CHARACTER PROCEDURE GETCHAR;
INTEGER PROCEDURE GETFRAC;
INTEGER PROCEDURE GETINT;
REAL PROCEDURE GETREAL;
INTEGER PROCEDURE LENGTH;
TEXT PROCEDURE MAIN;
BOOLEAN PROCEDURE MORE;
INTEGER PROCEDURE POS;
PROCEDURE PUTCHAR(C); CHARACTER C;
PROCEDURE PUTFIX(X,N); REAL X; INTEGER N;
PROCEDURE PUTFRAC(I,N); INTEGER I,N;
PROCEDURE PUTINT(I); INTEGER I;
PROCEDURE PUTREAL(X,N); REAL X; INTEGER N;
PROCEDURE SETPOS(N); INTEGER N;
TEXT PROCEDURE STRIP;
TEXT PROCEDURE SUB(I,N); INTEGER I,N;
SEQUENCING PROCEDURES

PROCEDURE CALL(X); REF(anyclass)X;
PROCEDURE DETACH;
PROCEDURE RESUME(Y); REF(anyclass)X;

RANDOM DRAWING PROCEDURES

INTEGER PROCEDURE DISCRETE(A,U); NAME U; ARRAY A; INTEGER U;
BOOLEAN PROCEDURE DRAW(A,U); NAME U; REAL A; INTEGER U;
REAL PROCEDURE ERLANG(A,B,U); NAME U; REAL A,B; INTEGER U;
INTEGER PROCEDURE HISTD(A,U); NAME U; ARRAY A; INTEGER U;
REAL PROCEDURE LINEAR(A,B,U); NAME U; ARRAY A,B; INTEGER U;
REAL PROCEDURE NEGEXP(A,U); NAME U; REAL A; INTEGER U;
REAL PROCEDURE NORMAL(A,B,U); NAME U; REAL A,B; INTEGER U;
INTEGER PROCEDURE POISSON(A,U); NAME U; REAL A; INTEGER U;
INTEGER PROCEDURE RANDINT(A,B,U); NAME U; INTEGER A,B,U;
REAL PROCEDURE UNIFORM(A,B,U); NAME U; REAL A,B; INTEGER U;

UTILITY PROCEDURES

PROCEDURE HISTO(A,B,C,D); ARRAY A,B; REAL C,D;
PROCEDURE LOWTEN(C); CHARACTER C;
SYSTEM DEFINED CLASSES

CLASS SIMSET:

CLASS SIMSET:
BEGIN CLASS LINKAGE;
BEGIN REF(LINK) PROCEDURE SUC;
REF(LINK) PROCEDURE PRED;
REF(LINKAGE) PROCEDURE PREV;
END ***LINKAGE*** ;

LINKAGE CLASS LINK;
BEGIN PROCEDURE OUT;
PROCEDURE INTO(H); REF(HEAD)H;
PROCEDURE PRECEDE(X); REF(LINKAGE)X;
PROCEDURE FOLLOW(X); REF(LINKAGE)X;
END ***LINK*** ;

LINKAGE CLASS HEAD;
BEGIN REF(LINK) PROCEDURE FIRST;
REF(LINK) PROCEDURE LAST;
PROCEDURE CLEAR;
BOOLEAN PROCEDURE EMPTY;
INTEGER PROCEDURE CARDINAL;
END ***HEAD*** ;

END ***SIMSET*** ;
CLASS SIMULATION

SIMSET CLASS SIMULATION;
BEGIN LINK CLASS PROCESS;
BEGIN BOOLEAN PROCEDURE IDLE;
BOOLEAN PROCEDURE TERMINATED;
REAL PROCEDURE EVTME;
REF(PROCESS) PROCEDURE NEXTEV;
END ***PROCESS*** ;

REF(PROCESS) PROCEDURE CURRENT;
LONG REAL PROCEDURE TIME;
PROCEDURE HOLD(T); REAL T;
PROCEDURE PASSIVATE;
PROCEDURE WAIT(Q); REF(HEAD)Q;
PROCEDURE CANCEL(X); REF(PROCESS)X;
PROCEDURE ACCUM(A,B,C,D); NAME A,B,C;
REAL A,B,C,D;

REF(main program)MAIN;
<ACTIVATION-statements>
END;
**SUBCLASSES of file**

```plaintext
CLASS file(name); VALUE name; TEXT name;
    VIRTUAL : PROCEDURE OPEN, CLOSE;
BEGIN  TEXT IMAGE;
    INTEGER PROCEDURE LENGTH;
    BOOLEAN PROCEDURE MORE;
    INTEGER PROCEDURE POS;
    PROCEDURE SETPOS(I); INTEGER I;
END ***file*** ;

file CLASS INFILE; VIRTUAL : BOOLEAN PROCEDURE ENDFILE;
    PROCEDURE INIMAGE;
BEGIN
    PROCEDURE CLOSE;
    BOOLEAN PROCEDURE ENDFILE;
    CHARACTER PROCEDURE INCHAR;
    PROCEDURE INIMAGE;
    INTEGER PROCEDURE ININT;
    INTEGER PROCEDURE INFRAC;
    REAL PROCEDURE INREAL;
    TEXT PROCEDURE INTEXT(N); INTEGER N;
    BOOLEAN PROCEDURE LASTITEM;
    PROCEDURE OPEN(T); TEXT T;
END ***INFILE*** ;
```
file CLASS OUTFILE; VIRTUAL : PROCEDURE OUTIMAGE;
BEGIN
PROCEDURE CLOSE;
PROCEDURE OPEN(T); TEXT T;
PROCEDURE OUTCHAR(C); CHARACTER C;
PROCEDURE OUTFIX(R,N,W); REAL R; INTEGER N,W;
PROCEDURE OUTFRACT(I,N,W); INTEGER I,N,W;
PROCEDURE OUTIMAGE;
PROCEDURE OUTINT(I,W); INTEGER I,W;
PROCEDURE OUTREAL(X,N,W); REAL R; INTEGER N,W;
PROCEDURE OUTTEXT(T); VALUE T; TEXT T;
END ***OUTFILE*** ;

OUTFILE CLASS PRINTFILE;
BEGIN
PROCEDURE CLOSE;
PROCEDURE EJECT(N); INTEGER N;
INTEGER PROCEDURE LINE;
PROCEDURE LINESPERPAGE(N); INTEGER N;
PROCEDURE OPEN(T); TEXT T;
PROCEDURE OUTIMAGE;
PROCEDURE SPACING(N); INTEGER N;
END ***PRINTFILE*** ;
file CLASS DIRECTFILE; VIRTUAL : PROCEDURE LOCATE OUTIMAGE, .INIMAGE;
  BOOLEAN PROCEDURE ENDFILE;
BEGIN
  PROCEDURE CLOSE;
  BOOLEAN PROCEDURE ENDFILE;
  CHARACTER PROCEDURE INCHAR;
  INTEGER PROCEDURE INFRAC;
  PROCEDURE INIMAGE;
  INTEGER PROCEDURE ININT;
  REAL PROCEDURE INREAL;
  TEXT PROCEDURE INTTEXT(N); INTEGER N;
  BOOLEAN PROCEDURE LASTITEM;
  PROCEDURE LOCATE(I); INTEGER I;
  INTEGER PROCEDURE LOCATION;
  PROCEDURE OPEN(T); TEXT T;
  PROCEDURE OUTCHAR(C); CHARACTER C;
  PROCEDURE OUTFIX(X,N,W); REAL X; INTEGER N,W;
  PROCEDURE OUTFRAC(I,N,W); INTEGER I,N,W;
  PROCEDURE OUTIMAGE;
  PROCEDURE OUTINT(I,W); INTEGER I,W;
  PROCEDURE OUTREAL(X,N,W); REAL X; INTEGER N,W;
  PROCEDURE OUTTEXT(T); VALUE T; TEXT T;
END ***DIRECTFILE*** ;
PROGRAM ENVIRONMENT

A user's program behaves as though enclosed as below:

basicio(132) BEGIN   INSPECT SYSIN DO
                     INSPECT SYSOUT DO
                     <program>
                     END

where "basicio" is defined by

CLASS basicio(linelen); INTEGER linelen;
BEGIN   REF(INFILE) PROCEDURE SYSIN;
         REF(OUTFILE) PROCEDURE SYSOUT;
         file CLASS INFILE;
         file CLASS OUTFILE;
OUTFILE CLASS PRINTFILE;
         file CLASS DIRECTFILE;
         sysin :- NEW INFILE("SYSIN");
         SYSIN.OPEN(BLANKS(80));
         sysout :- NEW OUTFILE("SYSOUT");
         SYSOUT.OPEN(BLANKS(132));
         INNER;
         SYSIN.CLOSE;
         SYSOUT.CLOSE;
END ***basicio*** ;
INDEX

******************************
* N.B. THE SECTIONS THAT THE INDEXED *
* WORD APPEAR IN ARE GIVEN ON THE RIGHT*
* AS A LIST IN THE FORMAT:        *
*   <PART NUMBER>;<SECTION NUMBER> *
* E.G. 2:7.2 MEANS PART 2,SECTION 7.2 *
******************************

ACCESSIBLE  3:1.0
ACCESSING    1:3.0
ACCUM        3:3.0
ACTIVATE     2:2.2, 3:3.0
ACTIVATION STATEMENTS  2:7.2
ACTIVATION-STATEMENT  2:7.2
ACTIVE       1:4.0, 3:1.0
ACTUAL-FORMAL PARAMETER CORRESPONDENCE  2:6.2
ACTUAL-PARAMETER  1:3.0, 2:5.4, 2:6.2
ACTUAL-PARAMETER-LIST  2:6.2
ACTUAL-PARAMETER-PART  2:6.7, 2:7.1
ADDITION SIGN (+)  2:2.2
AFTER        2:2.2, 3:3.0
ALPHABETIC CHARACTER  2:2.1
ALPHA NUMERIC CHARACTER  2:2.1
ALTERNATIVES  2:1.5
AND          1:2.0, 2:2.2, 2:2.2
              2:6.4
ARITHMETIC EXPRESSIONS  2:6.3
ARITHMETIC OPERATOR PRECEDENCE  2:6.3
ARITHMETIC OPERATORS AND TYPES  2:6.3
ARITHMETIC TYPE CONVERSION  2:5.1
ARITHMETICS OF (LONG)REAL QUANTITIES  2:6.3
ARITHMETIC-CONSTANTS  2:4.2
ARITHMETIC-CONSTANTS, USE OF  2:6.3
ARITHMETIC-CONSTANT  2:6.0
ARITHMETIC-EXPRESSIONS  2:6.1, 2:6.3, 2:7.2
ARITHMETIC-EXPRESSION  2:6.3
ARITHMETIC-FUNCTION-DESIGNATOR  2:2.2
ARITHMETIC-OPERATORS  2:6.3
ARITHMETIC-OPERATOR  2:6.3
ARITHMETIC-PRIMARY  2:6.3
ARITHMETIC-VARIABLE  2:6.3
ARRAY         1:2.0, 2:2.2, 2:5.2
              2:5.4, 2:5.5
ARRAY COMPONENT  2:5.2, 2:6.1
ARRAY DECLARATIONS  2:5.2
ARRAY DIMENSION    2:5.2
ARRAY SEGMENT      2:5.2
ARRAY-DECLARATION  2:5.0, 2:5.2
ARRAY-IDENTIFIER  2:5.2, 2:6.2
ARRAY,INITIALISATION OF  2:5.2
ASSIGNABLE RANGE,BOOLEAN  2:5.1
ASSIGNABLE RANGE,CHARACTER  2:5.1
ASSIGNABLE RANGE,INTEGER  2:5.1
ASSIGNABLE RANGE,REAL  2:5.1
ASSIGNABLE RANGE,REF  2:5.1
ASSIGNABLE RANGE,SHORT INTEGER  2:5.1
ASSIGNABLE RANGE,TEXT  2:6.1
ASSIGNMENT  2:7.2
ASSIGNMENTS  1:1.0, 2:7.2
ASSIGNMENT-STATEMENT  2:2.1
ASTERISK (*)  2:2.2, 3:3.0
AT  3:1.0
ATTACHED STATE  1:3.0, 2:5.5, 2:6.1
ATTRIBUTE  2:5.5
ATTRIBUTE REDEFINITION (VIRTUAL)  2:3.0
BASIC BINDING RULES  2:1.0, 2:1.2, 2:2.0
BASIC SYMBOL  2:2.1
BASIC SYMBOL SET  2:2.0
BASIC SYMBOLS AND SYNTACTIC VARIABLES  2:2.2
BASIC SYMBOLS  3:5.0
BASICIO  2:2.2
BECOMES SIGN (:=)  2:7.2
BECOMES (:=)  2:2.2, 3:3.0
BEFORE  2:2.2, 2:5.5, 2:7.1
BEGIN  1:1.0
BINDING RULE  2:3.0
BINDING RULES,BASIC  2:7.2
BINDING RULES,CONNECTION  2:6.1
BINDING RULES,REMOTE ACCESSING  2:5.5
BINDING RULES,VIRTUAL QUANTITIES  3:4.0
BLANK  2:2.1
BLANK (w)  3:4.0
BLANKS  2:4.2
BLANK,IN CHARACTER-CONSTANT  2:4.2
BLANK,IN TEXT-CONSTANT  1:1.0, 2:7.0, 2:7.1
BLOCK  2:7.2
BLOCK HEAD  2:5.0, 2:7.0
BLOCK INSTANCE
BLOCK PREFIX
BLOCKS AND COMPOUND STATEMENTS
BLOCKS AND STATEMENTS
BOOLEAN
BOOLEAN OPERATORS
BOOLEAN-CONSTANTS
BOOLEAN-EXPRESSIONS
BOOLEAN-FUNCTION-DESIGNATOR
BOOLEAN-OPERATOR
BOOLEAN-PRIMARY
BOOLEAN-VARIABLE
BRACES
BRACKETS
BUFFER
CALL
CALL BY NAME
CALL BY REFERENCE
CALL BY VALUE
CANCEL
CAR WASH SIMULATION
CARD READER
CHARACTER
CHARACTER ACCESS (TO TEXTS)
CHARACTER ARRAY
CHARACTER EXPRESSIONS
CHARACTER QUOTE ("")
CHARACTER STRING
CHARACTER VALUES; COMPARISON
CHARACTER-CONSTANTS
CHARACTER-CONSTANT
CHARACTER-EXPRESSIONS
CHARACTER-EXPRESSION
CHARACTER-FUNCTION-DESIGNATOR
CHARACTER-VARIABLE
CLASS
CLASS BASICIO
CLASS CAR
CLASS CARD
CLASS CIRCLE
CLASS DECK
CLASS DECLARATION
CLASS DECLARATIONS
<table>
<thead>
<tr>
<th>Command</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS DIRECTFILE</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CLASS FILE</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CLASS HAND</td>
<td>3:2.0</td>
</tr>
<tr>
<td>CLASS HEAD</td>
<td>1:4.0, 3:2.0</td>
</tr>
<tr>
<td>CLASS HIERARCHIES</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS INFILE</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS INSTANCE</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CLASS LINE</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS LINK</td>
<td>3:1.0</td>
</tr>
<tr>
<td>CLASS LINKAGE</td>
<td>3:1.0</td>
</tr>
<tr>
<td>CLASS OUTFILE</td>
<td>3:1.0</td>
</tr>
<tr>
<td>CLASS PARAMETERS (TABLE)</td>
<td>1:3.0</td>
</tr>
<tr>
<td>CLASS PLAYER</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CLASS POINT</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS PRINTFILE</td>
<td>1:4.0</td>
</tr>
<tr>
<td>CLASS PROCESS</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS ROW</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS ROW1</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS SICKP</td>
<td>3:3.0</td>
</tr>
<tr>
<td>CLASS SIMSET</td>
<td>3:2.0, 3:3.0</td>
</tr>
<tr>
<td>CLASS SIMULATION</td>
<td>1:4.0, 2:7.2, 3:3.0</td>
</tr>
<tr>
<td>CLASS STACKABLE</td>
<td>1:3.0</td>
</tr>
<tr>
<td>CLASS TREATMENT</td>
<td>3:3.0</td>
</tr>
<tr>
<td>CLASS WASH</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLASS-BODY</td>
<td>1:4.0</td>
</tr>
<tr>
<td>CLASS-DECLARATION</td>
<td>3:3.0</td>
</tr>
<tr>
<td>CLASS-IDENTIFIER</td>
<td>2:5.5, 2:6.7</td>
</tr>
<tr>
<td>CLASS-DECLARATION</td>
<td>2:5.5, 2:6.7</td>
</tr>
<tr>
<td>CLASS-SPEC-PART</td>
<td>2:7.1</td>
</tr>
<tr>
<td>CLOSE</td>
<td>2:5.5</td>
</tr>
<tr>
<td>CLOSED</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CODING FORM</td>
<td>3:5.0</td>
</tr>
<tr>
<td>CODING SIMULA PROGRAMS</td>
<td>2:7.2</td>
</tr>
<tr>
<td>COLLATING SEQUENCE</td>
<td>2:2.6</td>
</tr>
<tr>
<td>COLON</td>
<td>2:2.6</td>
</tr>
<tr>
<td>COLON (:)</td>
<td>2:2.6</td>
</tr>
<tr>
<td>COMMA (,)</td>
<td>2:2.6</td>
</tr>
<tr>
<td>COMMENT</td>
<td>1:1.0</td>
</tr>
<tr>
<td>COMMENT CONVENTIONS</td>
<td>2:1.2, 2:6.1, 2:6.2</td>
</tr>
<tr>
<td>COMPILATE TIME AND RUN TIME</td>
<td>2:7.2</td>
</tr>
<tr>
<td>COMPILATE TIME</td>
<td>2:7.2</td>
</tr>
<tr>
<td>COMPOUND TAIL</td>
<td>1:2.0, 2:7.1, 2:7.1</td>
</tr>
</tbody>
</table>
CONDITION

CONDITIONAL STATEMENTS
CONDITIONAL-STATEMENT
CONDITIONS
CONNECTION BLOCK
CONNECTION STATEMENTS
CONNECTION-STATEMENT
CONSTANT
CONSTANTS
CONTROL CARD
CONTROLLED-STATEMENT
CONTROLLED-VARIABLE
COPY
CURRENT
CURRENT POSITION INDICATOR (TEXT)
DATA CHARACTER SET
DATA SET
UDNAME
DECIMAL DIGIT
DECIMAL-CONSTANTS
DECIMAL-DIGIT
DECIMAL-DIGITS
DECLARATIONS
DECLARATION
DECLARATORS
DEFAULT ACTIONS (AFTER/BEFORE)
DEFAULT ACTIONS (AT/DELAY)
DEFAULT MODE OF PARAMETER TRANSMISSION
DEFAULT VALUE OF A FUNCTION-DESIGNATOR
DELAY
DELIMITERS
ENOTES SIGN (\-)
ENOTES (\-)
DESIGNATIONAL EXPRESSIONS
DESIGNATIONAL-EXPRESSION
DESIGNATIONAL-EXPRESSIONS
DETACH
DETACHLD STATE
DE-EDITING PROCEDURES
DIGIT
DIGITS
DIRECT ACCESS FILES
<table>
<thead>
<tr>
<th>Field</th>
<th>Section</th>
<th>Page</th>
<th>Level</th>
<th>Date</th>
<th>Originator</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>I</td>
<td>7</td>
<td>0</td>
<td>15/4-71</td>
<td>GMB</td>
</tr>
<tr>
<td>FIXED FIELD TEXT EDITING</td>
<td>3:5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOLLOW</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR</td>
<td>3:2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR LIST ELEMENTS</td>
<td>2:2.2</td>
<td>2:7.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR STATEMENTS</td>
<td>2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAL-ACTUAL PARAMETER CORRESPONDENCE</td>
<td>2:6.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAL-PARAMETER</td>
<td>2:3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAL-PARAMETER-PART</td>
<td>2:5.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORTRAN CODING FORM</td>
<td>2:2.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR-RIGHT-PART</td>
<td>2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR-STATEMENT</td>
<td>1:1.0, 2:7.2, 2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOR-STATEMENT,LOCAL LABELS</td>
<td>2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUNCTION DESIGNATORS</td>
<td>2:6.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FUNCTION-DECLARATION</td>
<td>2:5.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GE (&gt;=)</td>
<td>2:2.2, 2:6.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEOMETRICAL APPLICATIONS</td>
<td>2:6.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETCHAR</td>
<td>1:3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETFRAC</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETINT</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETREAL</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GO</td>
<td>2:2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOTO</td>
<td>2:2.2, 2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOTO EXIT</td>
<td>2:5.5, 3:1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOTO STATEMENTS</td>
<td>2:7.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOTO-STATEMENT</td>
<td>1:1.0, 2:7.2, 3:1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GREATER THAN SIGN (&gt;),</td>
<td>2:2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GREATER THAN (&gt;)</td>
<td>2:2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GREATER THAN OR EQUAL TO SIGN (&gt;=),</td>
<td>2:2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP MARKERS</td>
<td>2:1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUPED-ITEM</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUPS</td>
<td>3:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT (&gt;)</td>
<td>2:2.2, 2:6.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HASH SIGN (#)</td>
<td>2:4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HASH (#)</td>
<td>2:2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEAD</td>
<td>3:2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEXADECIMAL-CONSTANTS</td>
<td>2:4.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEXADECIMAL-CONSTANT</td>
<td>1:4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HISTORY OF A MODEL</td>
<td>3:3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOLD</td>
<td>2:3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDENTIFIERS AND KEY WORDS</td>
<td>2:2.0, 2:3.0, 2:5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDENTIFIER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
JOB CONTROL CARDS
KEY WORD
KEY WORDS
KEY-WORD-CONSTANTS
LABEL
LABELS LOCAL TO A FOR-STATEMENT
LABEL-IDENTIFIER
LANGUAGE CHARACTER SET
LASTITEM
LE (<=)
LEFT PARENTHESIS (())

LENGTH
LENGTH AND MAIN
LENGTH OF A TEXT VALUE
LENGTH OF IDENTIFIERS
LESS THAN OR EQUAL TO SIGN (<=)
LESS THAN SIGN (<)
LESS THAN (<)
LETTER
LINE
LINE PRINTER
LINELENGTH
LINESPERPAGE
LINK
LINKAGE
LIST PROCESSING
LOCAL OBJECTS
LOCAL QUANTITY
LOCAL SEQUENCE CONTROL (LSC)
LOCAL-OBJECT
LOCATE
LOCATION
LOGICAL AND (AND)
LOGICAL EQUIVALENCE (EQV)
LOGICAL IMPLICATION (IMP)
LOGICAL INCLUSIVE OR (OR)
LOGICAL NEGATION (NOT, !)
LOGICAL-OPERATORS
LONG
LONG REAL
LONG REAL-CONSTANT  2:4.2
LOWER SUBSCRIPT BOUND  2:5.2
LOWER-BOUND  2:5.2
LOWTEN  3:4.0
LSC  3:1.0
LT (<)  2:2.2, 2:16.4
MAGIC BOX (TEXT)  3:4.0
MAIN  3:3.0, 3:4.0
MAIN-BLOCK  2:7.1
MAIN-PART  2:5.5
MATCHING ATTRIBUTE (VIRTUAL)  2:5.5
MATCHING QUALIFICATION  2:5.4
METHOD OF SYNTAX SPECIFICATION  2:1.0
MINUS (-)  2:2.1
MODE  2:5.4
MODE-PART  2:5.4
MORE  3:4.0, 3:5.0
MULTIPLE ASSIGNMENT  1:2.0
MULTIPLICATION SIGN (*)  2:2.2
MULTIPLY (*)  2:2.1
NAME  1:3.0, 2:12.2, 2:5.4
NE (=)  2:2.2, 2:16.4
NEW  2:2.2, 2:6.7
NEXTEV  3:3.0
NONE  1:3.0, 2:12.2, 2:4.2
NON-EMPTY PREFIX AND EMPTY VIRTUAL PART  2:6.1, 2:16.7
NOT  2:5.5
NOT EQUAL TO SIGN (=)  2:2.2
NOT (-)  2:2.2, 2:12.2, 2:6.4
NOTEXT  2:6.4
NUMBERING A CROSSWORD PUZZLE  2:2.2, 2:16.8, 3:4.0
NUMERIC ITEM  1:2.0
NUMERIC-TEXT-VALUES  3:4.0
OBJECT  3:4.0
OBJECT EXPRESSIONS  2:5.5, 2:16.7
OBJECT GENERATORS  2:6.7
OBJECT PROGRAM  2:6.7, 2:7.2
OBJECT-ELEMENT  1:1.0
OBJECT-EXPRESSIONS  2:7.2
OBJECT-EXPRESSION  2:6.0
OBJECT-GENERATOR  2:6.7, 2:16.7, 2:7.2
OBJECT-REFERENCE  2:7.2
OBJECT-REFERENCE-CONSTANT  2:4.2, 2:14.2
OBJECT-REFERENCE  2:7.2
ONE DIMENSIONAL ARRAY
OPEN
OPERAND
OPERATOR
OPERATORS
OPTIONS
OR
OTHERWISE
OUT
OUTCHAR
OUTER
OUTER BLOCK
OUTFILE
OUTFIX
OUTFRAC
OUTIMAGE
OUTINT
OUTREAL
OUTTEXT
PARAMETER
PARAMETER TRANSMISSION
PARAMETERS TO CLASSES
PARAMETERS TO PROCEDURES
PARAMETER-PART
PASSIVATE
PERIOD (*)
PLUS (+)
POS
POWER SIGN (**)
PRECEDE
PRECEDENCE OF OPERATORS
PRED
PREFIX CHAIN
PREFIX SEQUENCE
PREFIXED BLOCK
PREFIXED-BLOCK
PREV
PRINTFILE
PRIOR
PROCEDURE
PROCEDURE ADD (REF(POINT))
PROCEDURE CALL_BY_VALUE
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCEDURE CHAR</td>
<td>2:6.5</td>
</tr>
<tr>
<td>PROCEDURE COMPRESS</td>
<td>3:4.0</td>
</tr>
<tr>
<td>PROCEDURE DECLARATIONS</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE DIGIT</td>
<td>2:6.5</td>
</tr>
<tr>
<td>PROCEDURE FACTORIAL (INTEGER)</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE INFECT</td>
<td>3:3.0</td>
</tr>
<tr>
<td>PROCEDURE LETTER</td>
<td>2:6.5</td>
</tr>
<tr>
<td>PROCEDURE NORM (REAL)</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE NULLEN (INTEGER)</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE OUTCOLUMN</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE PARAMETERS (TABLE)</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE PLACE</td>
<td>3:2.0</td>
</tr>
<tr>
<td>PROCEDURE RANK</td>
<td>2:6.5</td>
</tr>
<tr>
<td>PROCEDURE SELECT</td>
<td>2:7.2, 2:7.2</td>
</tr>
<tr>
<td>PROCEDURE STATEMENTS</td>
<td>3:2.0</td>
</tr>
<tr>
<td>PROCEDURE SWAP</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE THESETHEADDOT (REF(HEAD))</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE TREERAVENSE</td>
<td>3:2.0</td>
</tr>
<tr>
<td>PROCEDURE BODY</td>
<td>1:3.0, 2:3.0, 2:5.4</td>
</tr>
<tr>
<td>PROCESS DECLARATION</td>
<td>1:3.0, 2:5.0, 2:5.4</td>
</tr>
<tr>
<td>PROCEDURE-HEADING</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PROCEDURE-IDENTIFIER</td>
<td>2:7.2, 2:7.2, 2:7.2</td>
</tr>
<tr>
<td>PROCEDURE-STATEMENT</td>
<td>3:3.0</td>
</tr>
<tr>
<td>PROCESS OBJECT (TABLE OF STATES)</td>
<td>3:3.0</td>
</tr>
<tr>
<td>PROCESS-EXPRESSION</td>
<td>3:3.0</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>2:7.0, 2:7.1</td>
</tr>
<tr>
<td>PROGRAM CONTENT</td>
<td>2:2.0</td>
</tr>
<tr>
<td>PROGRAM SEQUENCE CONTROL</td>
<td>3:1.0</td>
</tr>
<tr>
<td>PROGRAM STRUCTURE</td>
<td>3:1.0</td>
</tr>
<tr>
<td>PROPER-PROCEDURE-DECLARATION</td>
<td>2:5.4</td>
</tr>
<tr>
<td>PSC</td>
<td>3:1.0</td>
</tr>
<tr>
<td>PUTCHAR</td>
<td>3:4.0</td>
</tr>
<tr>
<td>PUTFIX</td>
<td>3:4.0, 3:5.0</td>
</tr>
<tr>
<td>PUTINT</td>
<td>3:4.0, 3:5.0</td>
</tr>
<tr>
<td>PUTREAL</td>
<td>3:4.0, 3:5.0</td>
</tr>
<tr>
<td>QUA</td>
<td>2:2.2, 2:6.7</td>
</tr>
<tr>
<td>QUALIFICATION</td>
<td>1:3.0, 2:6.7</td>
</tr>
<tr>
<td>QUALIFICATION OF ACTUAL PARAMETER</td>
<td>2:5.4</td>
</tr>
<tr>
<td>QUALIFYING CLASS</td>
<td>2:5.1</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>2:3.0</td>
</tr>
<tr>
<td>QUASI-PARALLEL SYSTEM</td>
<td>2:7.1</td>
</tr>
<tr>
<td>R (REAL IN HEXADECIMAL-CONSTANTS)</td>
<td>2:4.2</td>
</tr>
<tr>
<td>RAISED-TO-THE-POWER-OF SIGN (**)</td>
<td>2:2.2</td>
</tr>
</tbody>
</table>
RANGE OF VALUES, BOOLEAN VARIABLES 2:5.1
RANGE OF VALUES, CHARACTER VARIABLES 2:5.1
RANGE OF VALUES, INTEGER VARIABLES 2:5.1
RANGE OF VALUES, INTEGER-CONSTANT 2:4.2
RANGE OF VALUES, LONG REAL VARIABLES 2:5.1
RANGE OF VALUES, LONG REAL-CONSTANT 2:4.2
RANGE OF VALUES, REAL VARIABLES 2:5.1
RANGE OF VALUES, REAL-CONSTANT 2:4.2
RANGE OF VALUES, REF VARIABLES 2:5.1
RANGE OF VALUES, SHORT INTEGER VARIABLES 2:5.1
RANGE OF VALUES, SHORT INTEGER-CONSTANT 2:4.2
RANGE OF VALUES, TEXT VARIABLES 2:5.1
RANK 2:6.4
REACTIVATE 2:2.2, 3:3.0
REAL 2:2.2, 2:4.1, 2:5.1
2:5.5, 2:6.3
2:4.2
REAL-CONSTANT 3:4.0
REAL-ITEM 3:5.0
RECORD 2:5.4
RECURSION 2:2.2, 2:4.1, 2:5.1
2:5.5
REF 2:6.4
REFERENCE COMPARATORS 2:2.2
REFERENCE EQUAL (==) 2:2.2, 2:6.4
REFERENCE NOT EQUAL (=/=) 1:3.0
REFERENCE VARIABLE 2:7.2
REFERENCE-ASSIGNMENT 2:2.2
REFERENCE-COMPARATORS 2:4.2
REFERENCE-CONSTANT 2:6.4
REFERENCE-EQUAL (==) 2:4.1, 2:5.1
REFERENCE-TYPE 2:6.4
RELATION 2:12.2, 2:2.2
2:6.4
RELATIONAL-OPERATORS 1:3.0, 2:6.1, 2:7.2
RELATIONS 2:6.2
REMOTE ACCESSING 2:6.1
REMOTE-IDENTIFIER 2:11.7
REMOTE-VARIABLE 2:5.5, 3:1.0, 3:3.0
REPETITION 2:2.2, 2:5.2, 2:5.4
2:5.5, 2:6.1, 2:6.3
2:6.4, 2:6.5, 2:6.6
2:6.7, 2:6.8
RESUME 1:2.0
RIGHT PARENTHESIS ( ) 1:1.0, 1:3.0

RTS BLOCK
RTS INFO
RUN TIME
RUN TIME ERROR
RUN TIME STRUCTURE
RUN TIME SYSTEM
RUN TIME SYSTEM INFORMATION
SCOPE
SECURITY
SEMANTICS
SEMICOLON
SEMICOLON (;)
SEPARATORS
SEQUENTIAL FILE ORGANISATION
SEQUENTIAL-OPERATORS
SET
SETPOS
SHORT INTEGER
SHORT INTEGER-CONSTANT
SIGN-PART
SIMPLE VARIABLE
SIMPLE-ARITHMETIC EXPRESSION
SIMPLE-CHARACTER-EXPRESSION
SIMPLE-DESIGNATIONAL-EXPRESSION
SIMPLE-OBJECT-EXPRESSION
SIMPLE-REFERENCE-EXPRESSION
SIMPLE-TEXT-EXPRESSION
SIMPLE-VALUE EXPRESSION
SIMPLE-VARIABLE
SIMSET
SIMSET, EXAMPLE ON THE USE OF
SIMSET, PREFIX TO SIMULATION
SIMULATION OF A QUEUING SITUATION
SOURCE PROGRAM
SPACE ( )
SPACING
SPECIAL CHARACTER
SPECIFIERS
SPEC-PART
SPLIT-BODY
SQUARE BRACKETS
STACK
STACK OF ATTACHED BLOCKS
STACK OF BLOCKS

1: 0
1:3, 3:5
3:1, 0
1:0
1:1, 0
2:3, 0
1:3, 2:5
0
2:1, 0
1:1, 0
2:2, 1, 2:5.5, 2:5.5
2:5, 5
2:2, 2
3:5, 0
2:2, 2
3:2, 0
3:4, 0, 3:5, 0
2:2, 2, 2:4, 1, 2:5, 1
2:5, 5, 2:6, 3
2:4, 2
3:4, 0
2:7, 2
2:6, 3
2:6, 5
2:6, 6
2:6, 1, 2:6, 4, 2:6, 7
2:6, 7, 2:7, 2
2:6, 4, 2:6, 7
2:6, 1, 2:6, 4, 2:6, 8
2:6, 4
2:6, 1
3:2, 0
3:2, 0
3:3, 0
1:4, 0
1:0
2:2, 1
3:5, 0
2:1, 0, 2:2, 1
2:2, 2
2:5, 4
2:5, 5
2:1, 6
1:3, 0
3:1, 0
3:1, 0
STANDARD SYSTEM PROCEDURES
STATEMENT

STATEMENTS
STATEMENT-BRACKETS
STEP
STRIP
STRUCTURE OF A QUANTITY
SUB
SUBCLASS
SUBSCRIPT
SUBSCRIPT BOUND
SUBSCRIPT BOUNDS
SUBSCRIPTED VARIABLE
SUBSCRIPTED-VARIABLE
SUBSCRIPT-LIST
SUBSCRIPT,SWITCH
SUBTEXTS
SUBTRACTION SIGN (-)
SUC
SWITCH

SWITCH DECLARATIONS
SWITCH-DECLARATION
SWITCH-DESIGNATOR
SWITCH-IDENTIFIER
SWITCH-LIST
SYNTACTIC UNIT
SYNTACTIC VARIABLE
SYNTAX
SYSIN
SYSOUT
TERMINATED
TERMINATED STATE
TEXT

TEXT ASSIGNMENT
TEXT EDITING AND DE-EDITING
TEXT EXPRESSIONS AND TEXT VALUES
TEXT GENERATION
TEXT HANDLING FACILITIES
TEXT HANDLING
TEXT OBJECT
TEXT QUOTE (△)
TEXT REFERENCE
TEXT VALUES, COMPARISON
TEXTUAL LINK
TEXTUALLY ENCLOSING BLOCK
TEXT-
TEXT-CONSTANT
TEXT-EXPRESSION
TEXT-FUNCTION-DESIGNATOR
TEXT-REFERENCE-ASSIGNMENT
TEXT-VALUE
TEXT-VALUE-ASSIGNMENT
TEXT-VARIABLE
THE CONTROLLED VARIABLE
THE HIGHEST MARK
THE SYNTAX AND SEMANTICS OF SIMULA
THE SYSTEM CLASS BASICIO
THE SYSTEM CLASS SIMSET
THE SYSTEM CLASS SIMULATION
THE USE OF BLANKS
THEN
THIS
THREE DIMENSIONAL ARRAY
THREE DOTS
TIME
TIME AXIS
TIMING CLAUSE
TO
TRACE OF A MODEL
TRUE
TWO DIMENSIONAL ARRAY
TWO WAY LIST
TYING A DDNAME TO A FILE
TYPE
TYPE ARRAY
TYPE CONVERSION (ARITHMETIC)
TYPE DECLARATIONS
TYPE OF A QUANTITY
TYPE OF ARITHMETIC EXPRESSIONS
TYPES
TYPES AND CONSTANTS
TYPE-DEKLARATION
UNDERSCORE (_)
UNTIL
UPPER SUBSCRIPT BOUND
UPPER-BOUND

Section: I
Page: 16
Level: 0
Date: 15/4-71
Originator: GMB