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

| SIMULA symbol | VALUE | | Punched card code |
|---|---|---|---|
| | decimal | hexadecimal | |
| ␣ | 64 | 40 | no punch |
| . | 75 | 4B | 12 - 3 - 8 |
| < | 76 | 4C | 12 - 4 - 8 |
| ( | 77 | 4D | 12 - 5 - 8 |
| + | 78 | 4E | 12 - 6 - 8 |
| & | 80 | 50 | 12 |
| $ | 91 | 5B | 11 - 3 - 8 |
| * | 92 | 5C | 11 - 4 - 8 |
| ) | 93 | 5D | 11 - 5 - 8 |
| ; | 94 | 5E | 11 - 6 - 8 |
| ¬ | 95 | 5F | 11 - 7 - 8 |
| - | 96 | 60 | 11 |
| / | 97 | 61 | 0 - 1 |
| , | 107 | 6B | 0 - 3 - 8 |
| _ | 109 | 6D | 0 - 5 - 8 |
| > | 110 | 6E | 0 - 6 - 8 |
| : | 122 | 7A | 2 - 8 |
| # | 123 | 7B | 3 - 8 |
| ' | 125 | 7D | 5 - 8 |
| = | 126 | 7E | 6 - 8 |
| " | 127 | 7F | 7 - 8 |
| A | 193 | C1 | 12 - 1 |
| B | 194 | C2 | 12 - 2 |
| C | 195 | C3 | 12 - 3 |
| D | 196 | C4 | 12 - 4 |

| SIMULA symbol | VALUE | | Punched card code |
|---|---|---|---|
| | decimal | hexadecimal | |
| E | 197 | C5 | 12 - 5 |
| F | 198 | C6 | 12 - 6 |
| G | 199 | C7 | 12 - 7 |
| H | 200 | C8 | 12 - 8 |
| I | 201 | C9 | 12 - 9 |
| J | 209 | D1 | 11 - 1 |
| K | 210 | D2 | 11 - 2 |
| L | 211 | D3 | 11 - 3 |
| M | 212 | D4 | 11 - 4 |
| N | 213 | D5 | 11 - 5 |
| O | 214 | D6 | 11 - 6 |
| P | 215 | D7 | 11 - 7 |
| Q | 216 | D8 | 11 - 8 |
| R | 217 | D9 | 11 - 9 |
| S | 226 | E1 | 0 - 2 |
| T | 227 | E2 | 0 - 3 |
| U | 228 | E3 | 0 - 4 |
| V | 229 | E4 | 0 - 5 |
| W | 230 | E5 | 0 - 6 |
| X | 231 | E6 | 0 - 7 |
| Y | 232 | E7 | 0 - 8 |
| Z | 233 | E8 | 0 - 9 |
| 0 | 240 | F0 | 0 |
| 1 | 241 | F1 | 1 |
| 2 | 242 | F2 | 2 |
| 3 | 243 | F3 | 3 |
| 4 | 244 | F4 | 4 |
| 5 | 245 | F5 | 5 |
| 6 | 246 | F6 | 6 |
| 7 | 247 | F7 | 7 |
| 8 | 248 | F8 | 8 |
| 9 | 249 | F9 | 9 |

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)<br>ARCSIN(E)<br>ARCTAN(E) | return the principal values of the arc-cosine, arc-sine, arc-tangent of E (E is measured in radians) |
| COS(E)<br>SIN(E)<br>TAN(E) | return the cosine, sine tangent of E (E is measured in radians) |
| COSH(E)<br>SINH(E)<br>TANH(E) | return the hyperbolic cosine, hyperbolic sine, hyperbolic tangent of E (E is measured in radians) |
| EXP(E) | exponential function of E ($e^E$) |
| LN(E) | natural logarithm of E ($\log_2 E$, 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. ENTIER(5.3) = 5
ENTIER(-4.7) = -5
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)  = '␣'
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

```
ENTIER(0.5 + 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

$$\left.\begin{array}{l} A(I) >= A(J) \\ B(I) >= B(J) \end{array}\right\} \text{ for } I > J$$

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 / REAL}ARRAY A,B;
{INTEGER / 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 ARCCOS(X); REAL X;
REAL PROCEDURE ARCSIN(X); REAL X;
REAL PROCEDURE ARCTAN(X); REAL X;
REAL PROCEDURE COS(X); REAL X;
REAL PROCEDURE COSH(X); REAL X;
REAL PROCEDURE EXP(X); REAL X;
REAL PROCEDURE LN(X); REAL X;
REAL PROCEDURE SIN(X); REAL X;
REAL PROCEDURE SINH(X); REAL X;
REAL PROCEDURE SQRT(X); REAL X;
REAL PROCEDURE TAN(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 EVTIME;
                     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

```
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 OUTFRAC(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 INTEXT(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(linelength); INTEGER linelength;
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
ALPHANUMERIC 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 2:4.2
ARITHMETIC-CONSTANT 2:6.3
ARITHMETIC-EXPRESSIONS 2:6.0
ARITHMETIC-EXPRESSION 2:6.1, 2:6.3, 2:7.2
ARITHMETIC-FUNCTION-DESIGNATOR 2:6.3
ARITHMETIC-OPERATORS 2:2.2
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,LONG REAL 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 2:5.1
ASSIGNMENT 2:6.1
ASSIGNMENTS 2:7.2
ASSIGNMENT-STATEMENT 1:1.0, 2:7.2
ASTERISK (*) 2:2.1
AT 2:2.2, 3:3.0
ATTACHED STATE 3:1.0
ATTRIBUTE 1:3.0, 2:5.5, 2:6.1
ATTRIBUTE REDEFINITION (VIRTUAL) 2:5.5
BASIC BINDING RULES 2:3.0
BASIC SYMBOL 2:1.0, 2:1.2, 2:2.0
BASIC SYMBOL SET 2:2.1
BASIC SYMBOLS AND SYNTACTIC VARIABLES 2:2.0
BASIC SYMBOLS 2:2.2
BASICIO 3:5.0
BECOMES SIGN (:=) 2:2.2
BECOMES (:=) 2:7.2
BEFORE 2:2.2, 3:3.0
BEGIN 2:2.2, 2:5.5, 2:7.1
BINDING RULE 1:1.0
BINDING RULES,BASIC 2:3.0
BINDING RULES,CONNECTION 2:7.2
BINDING RULES,REMOTE ACCESSING 2:6.1
BINDING RULES,VIRTUAL QUANTITIES 2:5.5
BLANK 3:4.0
BLANK (␣) 2:2.1
BLANKS 3:4.0
BLANK,IN CHARACTER-CONSTANT 2:4.2
BLANK,IN TEXT-CONSTANT 2:4.2
BLOCK 1:1.0, 2:7.0, 2:7.1
2:7.2
BLOCK HEAD 2:5.0, 2:7.0

BLOCK INSTANCE 1:1.0
BLOCK PREFIX 3:2.0
BLOCKS AND COMPOUND STATEMENTS 2:7.1
BLOCKS AND STATEMENTS 2:7.0
BOOLEAN 2:2.2, 2:4.1, 2:5.1 2:5.5
BOOLEAN OPERATORS 2:6.4
BOOLEAN-CONSTANTS 2:4.2, 2:4.2
BOOLEAN-EXPRESSIONS 2:6.4
BOOLEAN-FUNCTION-DESIGNATOR 2:6.4
BOOLEAN-OPERATOR 2:6.4
BOOLEAN-PRIMARY 2:6.4
BOOLEAN-VARIABLE 2:6.4
BRACES 2:1.4
BRACKETS 2:2.2
BUFFER 3:5.0
CALL 2:5.5, 3:1.0
CALL BY NAME 2:5.4, 2:5.5, 2:7.2
CALL BY REFERENCE 2:5.4, 2:5.5
CALL BY VALUE 2:5.4, 2:5.5
CANCEL 3:3.0
CAR WASH SIMULATION 1:4.0
CARD READER 3:5.0
CHARACTER 2:2.2, 2:4.1, 2:5.1 2:5.5
CHARACTER ACCESS (TO TEXTS) 3:4.0
CHARACTER ARRAY 3:4.0
CHARACTER EXPRESSIONS 2:6.5
CHARACTER QUOTE (>) 2:2.1, 2:4.2
CHARACTER STRING 3:4.0
CHARACTER VALUES,COMPARISON 2:6.4
CHARACTER-CONSTANTS 2:4.2
CHARACTER-CONSTANT 2:4.2, 2:6.5
CHARACTER-EXPRESSIONS 2:6.0
CHARACTER-EXPRESSION 2:6.5
CHARACTER-FUNCTION-DESIGNATOR 2:6.5
CHARACTER-VARIABLE 2:6.5
CLASS 1:3.0, 2:2.2, 2:5.5
CLASS BASICIO 3:5.0
CLASS CAR 1:4.0
CLASS CARD 3:2.0
CLASS CIRCLE 1:3.0
CLASS DECK 3:2.0
CLASS DECLARATION 1:3.0
CLASS DECLARATIONS 2:5.5

CLASS DIRECTFILE 3:5.0
CLASS FILE 3:5.0
CLASS HAND 3:2.0
CLASS HEAD 1:4.0, 3:2.0
CLASS HIERARCHIES 2:5.5
CLASS INFILE 3:5.0
CLASS INSTANCE 2:5.5
CLASS LINE 1:3.0
CLASS LINK 1:4.0, 3:2.0
CLASS LINKAGE 3:2.0
CLASS OUTFILE 3:5.0
CLASS PARAMETERS (TABLE) 2:5.5
CLASS PLAYER 3:1.0
CLASS POINT 1:3.0
CLASS PRINTFILE 3:5.0
CLASS PROCESS 1:4.0
CLASS ROW 2:5.5
CLASS ROW1 2:5.5
CLASS SICKP 3:3.0
CLASS SIMSET 3:2.0, 3:3.0
CLASS SIMULATION 1:4.0, 2:7.2, 3:3.0
CLASS STACKABLE 1:3.0
CLASS TREATMENT 3:3.0
CLASS WASH 1:4.0
CLASS-BODY 1:3.0, 2:3.0, 2:5.0
2:5.5, 2:5.5, 2:6.7
CLASS-DECLARATION 2:5.0, 2:5.5
CLASS-IDENTIFIER 2:5.5, 2:6.4, 2:6.7
2:7.1
CLASS-SPEC-PART 2:5.5
CLOSE 3:5.0
CLOSED 2:7.2
CODING FORM 2:2.6
CODING SIMULA PROGRAMS 2:2.6
COLLATING SEQUENCE 2:2.3
COLON 1:1.0
COLON (:) 2:2.1, 2:5.5, 2:7.2
COMMA (,) 2:2.1, 2:6.1, 2:6.2
2:7.2
COMMENT 1:2.0, 2:2.2
COMMENT CONVENTIONS 2:2.5
COMPILE TIME AND RUN TIME 1: .0
COMPILE TIME 1: .0
COMPOUND TAIL 2:7.0
COMPOUND-STATEMENT 1:2.0, 2:7.1, 2:7.1

| | |
|---|---|
| | 2:7.2 |
| CONDITION | 2:6.0, 2:6.0, 2:6.3 2:6.4, 2:7.2 |
| CONDITIONAL STATEMENTS | 2:7.2 |
| CONDITIONAL-STATEMENT | 1:1.0, 2:7.2, 2:7.2 |
| CONDITIONS | 2:6.4 |
| CONNECTION BLOCK | 2:6.7 |
| CONNECTION STATEMENTS | 2:7.2 |
| CONNECTION-STATEMENT | 2:7.2, 2:7.2 |
| CONSTANT | 2:2.0, 2:4.0, 2:6.0 |
| CONSTANTS | 2:4.2 |
| CONTROL CARD | 2:2.2 |
| CONTROLLED-STATEMENT | 2:7.2 |
| CONTROLLED-VARIABLE | 2:7.2 |
| COPY | 3:4.0 |
| CURRENT | 3:3.0 |
| CURRENT POSITION INDICATOR (TEXT) | 3:5.0 |
| DATA CHARACTER SET | 2:2.3 |
| DATA SET | 3:5.0 |
| DDNAME | 3:5.0 |
| DECIMAL DIGIT | 2:2.1 |
| DECIMAL-CONSTANTS | 2:4.2 |
| DECIMAL-DIGIT | 2:4.2 |
| DECIMAL-DIGITS | 2:4.2, 2:4.2 |
| DECLARATIONS | 2:5.0 |
| DECLARATION | 2:5.0, 2:5.5, 2:7.1 |
| DECLARATORS | 2:2.2 |
| DEFAULT ACTIONS (AFTER/BEFORE) | 3:3.0 |
| DEFAULT ACTIONS (AT/DELAY) | 3:3.0 |
| DEFAULT MODE OF PARAMETER TRANSMISSION | 2:5.4 |
| DEFAULT VALUE OF A FUNCTION-DESIGNATOR | 2:5.4 |
| DELAY | 2:2.2, 3:3.0 |
| DELIMETERS | 2:2.2 |
| DENOTES SIGN (:-) | 2:2.2 |
| DENOTES (:-) | 2:7.2 |
| DESIGNATIONAL EXPRESSIONS | 2:6.6 |
| DESIGNATIONAL-EXPRESSION | 2:5.3, 2:6.6, 2:7.2 |
| DESIGNATIONAL-EXPRESSIONS | 2:6.0 |
| DETACH | 2:5.5, 2:6.7, 3:1.0 3:3.0 |
| DETACHED STATE | 2:6.7, 3:1.0 |
| DE-EDITING PROCEDURES | 3:4.0 |
| DIGIT | 2:3.0 |
| DIGITS | 3:4.0 |
| DIRECT ACCESS FILES | 3:5.0 |

DIRECT ACTIVATION 3:3.0
DIRECTFILE 3:5.0
DIVIDE (/) 2:2.1
DIVISION SIGN (/) 2:2.2
DO 2:2.2, 2:7.2, 2:7.2
DOLLAR ($) 2:2.1
DOT NOTATION 1:3.0
DOT (.) 2:2.1, 2:4.2, 2:6.1
2:7.2
DUMMY STATEMENTS 2:7.2
DUMMY-STATEMENT 2:5.5, 2:7.2, 2:7.2
EDITING PROCEDURES 3:4.0
EJECT 3:5.0
ELSE 2:2.2, 2:6.3, 2:6.4
2:6.5, 2:6.7, 2:6.8
EMPTY PART AND VIRTUAL PART 2:5.5
EMPTY TEXT VALUE (NOTEXT) 2:2.2
END 2:2.2, 2:5.5, 2:7.1
END OF FILE 3:5.0
ENDFILE 3:5.0
EPIDEMIC 3:3.0
EQ (=) 2:2.2, 2:6.4
EQUAL TO SIGN (=) 2:2.2
EQUAL (W.R.T. QUALIFICATION) 2:5.5
EQUALS (=) 2:2.1
EQUIVALENCE (EQV) 2:2.2
EQV 2:2.2, 2:2.2, 2:6.4
EVENT NOTICE 3:3.0
EVTIME 3:3.0
EXAMPLE ON THE USE OF FILES 3:5.0
EXAMPLE ON THE USE OF SIMULATION 3:3.0
EXIT FROM A BLOCK 2:5.0
EXPONENT 2:4.2
EXPONENT SIGN IN DECIMAL-CONSTANT 2:4.2
EXPONENT SIGN (&) 2:4.2
EXPONENT (&) 2:2.1
EXPRESSION 2:6.2, 2:7.2
EXPRESSIONS 2:6.0
EXTERNAL DATA 3:5.0
EXTERNAL DECLARATIONS 2:5.6
EXTERNAL FILE 3:5.0
EXTERNAL RECORD 3:5.0
EXTERNAL-DECLARATION 2:5.0, 2:5.6
E,EXPONENT SIGN 3:4.0
FALSE 2:2.2, 2:4.2, 2:6.4

FIELD 3:5.0
FILE 3:5.0, 3:5.0
FIXED FIELD TEXT EDITING 3:4.0
FOLLOW 3:2.0
FOR 2:2.2, 2:7.2
FOR LIST ELEMENTS 2:7.2
FOR STATEMENTS 2:7.2
FORMAL-ACTUAL PARAMETER CORRESPONDENCE 2:6.2
FORMAL-PARAMETER 2:3.0
FORMAL-PARAMETER-PART 2:5.4
FORTRAN CODING FORM 2:2.6
FOR-RIGHT-PART 2:7.2
FOR-STATEMENT 1:1.0, 2:7.2, 2:7.2
FOR-STATEMENT,LOCAL LABELS 2:7.2
FUNCTION DESIGNATORS 2:6.2
FUNCTION-DECLARATION 2:5.4
FUNCTION-DESIGNATOR 2:6.0, 2:6.2, 2:6.7
GE (>=) 2:2.2, 2:6.4
GENERATED OBJECT 2:6.7
GEOMETRICAL APPLICATIONS 1:3.0
GETCHAR 3:4.0
GETFRAC 3:4.0
GETINT 3:4.0
GETREAL 3:4.0
GO 2:2.2
GOTO 2:2.2, 2:7.2
GOTO EXIT 2:5.5, 3:1.0
GOTO STATEMENTS 2:7.2
GOTO-STATEMENT 1:1.0, 2:7.2, 3:1.0
GREATER THAN SIGN (>) 2:2.2
GREATER THAN (>) 2:2.1
GREATER THEN OR EQUAL TO SIGN (>=) 2:2.2
GROUP MARKERS 2:1.4
GROUPED-ITEM 3:4.0
GROUPS 3:4.0
GT (>) 2:2.2, 2:6.4
HASH SIGN (#) 2:4.2
HASH (#) 2:2.1
HEAD 3:2.0
HEXADECIMAL-CONSTANTS 2:4.2
HEXADECIMAL-CONSTANT 2:4.2
HISTORY OF A MODEL 1:4.0
HOLD 3:3.0
IDENTIFIERS AND KEY WORDS 2:3.0
IDENTIFIER 2:2.0, 2:3.0, 2:5.0

| | |
|---|---|
| | 2:6.1, 2:6.2 |
| IDENTIFIER CLASH | 3:1.0 |
| IDENTIFIERS | 2:3.0 |
| IDENTIFIER-LIST | 2:5.0, 2:5.1, 2:5.2 |
| | 2:5.4, 2:5.5 |
| IDLE | 3:3.0 |
| IF | 2:2.2, 2:6.0, 2:7.2 |
| IF-CLAUSE | 2:6.0, 2:6.3, 2:6.4 |
| | 2:6.5, 2:6.7, 2:6.8 |
| | 2:7.2 |
| IF-STATEMENT | 2:7.2, 2:7.2 |
| IMAGE | 3:5.0, 3:5.0 |
| IMP | 2:2.2, 2:2.2, 2:6.4 |
| IMPLICATION (IMP) | 2:2.2 |
| IN | 2:2.2, 2:6.4 |
| INCHAR | 3:5.0 |
| INCLUDE | 2:5.5 |
| INFILE | 3:5.0 |
| INFRAC | 3:5.0 |
| INIMAGE | 3:5.0 |
| ININT | 3:5.0 |
| INITIAL VALUE | 1:1.0, 2:4.0 |
| INITIAL VALUE,BOOLEAN | 2:5.1 |
| INITIAL VALUE,CHARACTER | 2:5.1 |
| INITIAL VALUE,INTEGER | 2:5.1 |
| INITIAL VALUE,LONG REAL | 2:5.1 |
| INITIAL VALUE,REAL | 2:5.1 |
| INITIAL VALUE,REF | 2:5.1 |
| INITIAL VALUE,SHORT INTEGER | 2:5.1 |
| INITIAL VALUE,TEXT | 2:5.1 |
| INNER | 2:2.2, 2:5.4, 2:5.5 |
| | 2:5.5, 2:7.2, 3:5.0 |
| INNER ACCESSIBILITY (VIRTUAL) | 2:5.5 |
| INNER BLOCK | 1:2.0 |
| INREAL | 3:5.0, 3:5.0 |
| INSPECT | 1:3.0, 2:2.2, 2:7.2 |
| INSTANTANEOUS QUALIFICATION | 2:6.7 |
| INTEGER | 2:2.2, 2:4.1, 2:5.1 |
| | 2:5.5, 2:6.3 |
| INTEGER DIVISION SIGN (//) | 2:2.2 |
| INTEGER-CONSTANT | 2:4.2 |
| INTEGER-ITEM | 3:4.0 |
| INTEXT | 3:5.0 |
| INTO | 3:2.0 |
| IS | 2:2.2, 2:6.4 |

JOB CONTROL CARDS 2:2.6
KEY WORD 2:1.0, 2:3.0
KEY WORDS 2:2.2
KEY-WORD-CONSTANTS 2:2.2
LABEL 2:2.2, 2:3.0, 2:5.4 2:5.5, 2:7.2, 2:7.2
LABELS LOCAL TO A FOR-STATEMENT 2:7.2
LABEL-IDENTIFIER 2:6.2, 2:6.6, 2:7.2
LANGUAGE CHARACTER SET 2:2.0, 2:2.1
LASTITEM 3:5.0
LE (<=) 2:2.2, 2:6.4
LEFT PARENTHESIS (() 2:2.1, 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
LENGTH 3:4.0, 3:5.0
LENGTH AND MAIN 3:4.0
LENGTH OF A TEXT VALUE 3:4.0
LENGTH OF IDENTIFIERS 2:3.0
LESS THAN OR EQUAL TO SIGN (<=) 2:2.2
LESS THAN SIGN (<) 2:2.2
LESS THAN (<) 2:2.1
LETTER 2:3.0
LINE 3:5.0
LINE PRINTER 3:5.0
LINELENGTH 3:5.0
LINESPERPAGE 3:5.0
LINK 3:2.0
LINKAGE 3:2.0
LIST PROCESSING 3:2.0
LOCAL OBJECTS 2:6.7
LOCAL QUANTITY 2:7.1
LOCAL SEQUENCE CONTROL (LSC) 3:1.0
LOCAL-OBJECT 2:6.7
LOCATE 3:5.0
LOCATION 3:5.0
LOGICAL AND (AND) 2:2.2
LOGICAL EQUIVALENCE (EQV) 2:2.2
LOGICAL IMPLICATION (IMP) 2:2.2
LOGICAL INCLUSIVE OR (OR) 2:2.2
LOGICAL NEGATION (NOT,¬) 2:2.2
LOGICAL-OPERATORS 2:2.2, 2:2.2
LONG 2:5.1
LONG REAL 2:2.2, 2:4.1, 2:5.5 2:6.3

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, 2:6.4
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, 2:2.2, 2:5.4
NE (¬=) 2:2.2, 2:6.4
NEW 2:2.2, 2:6.7
NEXTEV 3:3.0
NONE 1:3.0, 2:2.2, 2:4.2
2:6.1, 2:6.7
NON-EMPTY PREFIX AND EMPTY VIRTUAL PART 2:5.5
NOT 2:2.2
NOT EQUAL TO SIGN (¬=) 2:2.2
NOT (¬) 2:2.1, 2:2.2, 2:6.4
2:6.4
NOTEXT 2:2.2, 2:6.8, 3:4.0
NUMBERING A CROSSWORD PUZZLE 1:2.0
NUMERIC ITEM 3:4.0
NUMERIC-TEXT-VALUES 3:4.0
OBJECT 2:5.5, 2:6.7
OBJECT EXPRESSIONS 2:6.7
OBJECT GENERATORS 2:6.7, 2:7.2
OBJECT PROGRAM 1: .0
OBJECT-ELEMENT 2:7.2
OBJECT-EXPRESSIONS 2:6.0
OBJECT-EXPRESSION 2:6.7, 2:6.7, 2:7.2
OBJECT-GENERATOR 2:7.2
OBJECT-REFERENCE-CONSTANT 2:4.2, 2:4.2
OBJECT-REFERENCE 2:7.2

ONE DIMENSIONAL ARRAY 2:5.2
OPEN 2:7.2, 3:5.0
OPERAND 2:6.0
OPERATOR 2:6.0
OPERATORS 2:2.2
OPTIONS 2:1.6
OR 1:2.0, 2:2.2, 2:2.2 2:6.4
OTHERWISE 2:2.2, 2:7.2
OUT 3:2.0
OUTCHAR 3:5.0
OUTER 2:5.5
OUTER BLOCK 1:2.0
OUTFILE 3:5.0
OUTFIX 3:5.0
OUTFRAC 3:5.0
OUTIMAGE 3:5.0, 3:5.0
OUTINT 3:5.0
OUTREAL 3:5.0
OUTTEXT 3:5.0
PARAMETER 2:5.4
PARAMETER TRANSMISSION 2:5.4
PARAMETERS TO CLASSES 2:5.4
PARAMETERS TO PROCEDURES 2:5.4
PARAMETER-PART 2:5.5
PASSIVATE 3:3.0
PASSIVE 1:4.0, 3:1.0
PERIOD (.) 2:2.1
PLUS (+) 2:2.1
POS 3:4.0, 3:5.0
POWER SIGN (**) 2:2.2
PRECEDE 3:2.0
PRECEDENCE OF OPERATORS 2:6.4
PRED 3:2.0
PREFIX CHAIN 2:5.5
PREFIX SEQUENCE 2:5.5
PREFIXED BLOCK 1:3.0, 2:7.1
PREFIXED-BLOCK 2:5.5, 2:7.1
PREV 3:2.0
PRINTFILE 3:5.0
PRIOR 2:2.2, 3:3.0
PROCEDURE 1:3.0, 2:2.2, 2:5.4 2:5.5
PROCEDURE ADD (REF(POINT)) 2:5.4
PROCEDURE CALL_BY_VALUE 2:5.4

PROCEDURE CHAR 2:6.5
PROCEDURE COMPRESS 3:4.0
PROCEDURE DECLARATIONS 2:5.4
PROCEDURE DIGIT 2:6.5
PROCEDURE FACTORIAL (INTEGER) 2:5.4
PROCEDURE INFECT 3:3.0
PROCEDURE LETTER 2:6.5
PROCEDURE NORM (REAL) 2:5.4
PROCEDURE NULLREF (INTEGER) 2:5.4
PROCEDURE OUTCOLUMN 2:5.4
PROCEDURE PARAMETERS (TABLE) 2:5.4
PROCEDURE PLACE 3:2.0
PROCEDURE RANK 2:6.5
PROCEDURE SELECT 3:2.0
PROCEDURE STATEMENTS 2:7.2, 2:7.2
PROCEDURE SWAP 2:5.4
PROCEDURE THESETHEADOF (REF(HEAD)) 3:2.0
PROCEDURE TREETRAVERSE 2:5.4
PROCEDURE-BODY 1:3.0, 2:3.0, 2:5.4
PROCEDURE-DECLARATION 1:3.0, 2:5.0, 2:5.4
PROCEDURE-HEADING 2:5.4
PROCEDURE-IDENTIFIER 2:5.4, 2:6.2, 2:7.2
2:7.2
PROCEDURE-STATEMENT 2:7.2, 2:7.2, 2:7.2
PROCESS 3:3.0
PROCESS OBJECT (TABLE OF STATES) 3:3.0
PROCESS-EXPRESSION 3:3.0
PROGRAM 2:7.0, 2:7.1
PROGRAM CONTENT 2:2.0
PROGRAM SEQUENCE CONTROL 3:1.0
PROGRAM STRUCTURE 3:1.0
PROPER-PROCEDURE-DECLARATION 2:5.4
PSC 3:1.0
PUTCHAR 3:4.0
PUTFIX 3:4.0, 3:5.0
PUTINT 3:4.0, 3:5.0
PUTREAL 3:4.0, 3:5.0
QUA 2:2.2, 2:6.7
QUALIFICATION 1:3.0, 2:6.7
QUALIFICATION OF ACTUAL PARAMETER 2:5.4
QUALIFYING CLASS 2:5.1
QUANTITY 2:3.0
QUASI-PARALLEL SYSTEM 2:7.1
R (REAL IN HEXADECIMAL-CONSTANTS ) 2:4.2
RAISED-TO-THE-POWER-OF SIGN (**) 2:2.2

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
REAL-CONSTANT 2:4.2
REAL-ITEM 3:4.0
RECORD 3:5.0
RECURSION 2:5.4
REF 2:2.2, 2:4.1, 2:5.1
2:5.5
REFERENCE COMPARATORS 2:6.4
REFERENCE EQUAL (==) 2:2.2
REFERENCE NOT EQUAL (=/=) 2:2.2, 2:6.4
REFERENCE VARIABLE 1:3.0
REFERENCE-ASSIGNMENT 2:7.2
REFERENCE-COMPARATORS 2:2.2
REFERENCE-CONSTANT 2:4.2
REFERENCE-EQUAL (==) 2:6.4
REFERENCE-TYPE 2:4.1, 2:5.1
RELATION 2:6.4
RELATIONAL-OPERATORS 2:2.2, 2:2.2
RELATIONS 2:6.4
REMOTE ACCESSING 1:3.0, 2:6.1, 2:7.2
REMOTE-IDENTIFIER 2:6.2
REMOTE-VARIABLE 2:6.1
REPETITION 2:1.7
RESUME 2:5.5, 3:1.0, 3:3.0
RIGHT PARENTHESIS ()) 2:2.1, 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
RTS BLOCK 1:2.0
RTS INFO 1:1.0, 1:3.0

RUN TIME 1: .0
RUN TIME ERROR 1:3.0, 3:5.0
RUN TIME STRUCTURE 3:1.0
RUN TIME SYSTEM 1: .0
RUN TIME SYSTEM INFORMATION 1:1.0
SCOPE 2:3.0
SECURITY 1:3.0, 2:5.0
SEMANTICS 2:1.0
SEMICOLON 1:1.0
SEMICOLON (;) 2:2.1, 2:5.4, 2:5.5
2:5.5
SEPARATORS 2:2.2
SEQUENTIAL FILE ORGANISATION 3:5.0
SEQUENTIAL-OPERATORS 2:2.2
SET 3:2.0
SETPOS 3:4.0, 3:5.0
SHORT INTEGER 2:2.2, 2:4.1, 2:5.1
2:5.5, 2:6.3
SHORT INTEGER-CONSTANT 2:4.2
SIGN-PART 3:4.0
SIMPLE VARIABLE 2:7.2
SIMPLE-ARITHMETIC EXPRESSION 2:6.3
SIMPLE-CHARACTER-EXPRESSION 2:6.5
SIMPLE-DESIGNATIONAL-EXPRESSION 2:6.6
SIMPLE-OBJECT-EXPRESSION 2:6.1, 2:6.4, 2:6.7
2:6.7, 2:7.2
SIMPLE-REFERENCE-EXPRESSION 2:6.4, 2:6.7
SIMPLE-TEXT-EXPRESSION 2:6.1, 2:6.4, 2:6.8
SIMPLE-VALUE EXPRESSION 2:6.4
SIMPLE-VARIABLE 2:6.1
SIMSET 3:2.0
SIMSET,EXAMPLE ON THE USE OF 3:2.0
SIMSET,PREFIX TO SIMULATION 3:3.0
SIMULATION OF A QUEUING SITUATION 1:4.0
SOURCE PROGRAM 1: .0
SPACE (␣) 2:2.1
SPACING 3:5.0
SPECIAL CHARACTER 2:1.0, 2:2.1
SPECIFICATORS 2:2.2
SPEC-PART 2:5.4
SPLIT-BODY 2:5.5
SQUARE BRACKETS 2:1.6
STACK 1:3.0
STACK OF ATTACHED BLOCKS 3:1.0
STACK OF BLOCKS 3:1.0

STANDARD SYSTEM PROCEDURES 1:2.0
STATEMENT 1:1.0, 2:5.4, 2:5.5 2:7.1, 2:7.2, 2:7.2
STATEMENTS 2:7.2
STATEMENT-BRACKETS 2:2.2
STEP 2:2.2, 2:7.2
STRIP 3:4.0
STRUCTURE OF A QUANTITY 2:5.0
SUB 3:4.0
SUBCLASS 2:5.1, 2:5.5
SUBSCRIPT 2:6.1
SUBSCRIPT BOUND 2:5.2
SUBSCRIPT BOUNDS 2:6.1
SUBSCRIPTED VARIABLE 2:5.2
SUBSCRIPTED-VARIABLE 2:6.1
SUBSCRIPT-LIST 2:6.1
SUBSCRIPT,SWITCH 2:6.6
SUBTEXTS 3:4.0
SUBTRACTION SIGN (-) 2:2.2
SUC 3:2.0
SWITCH 2:2.2, 2:5.3, 2:5.4 2:5.5
SWITCH DECLARATIONS 2:5.3
SWITCH-DECLARATION 2:5.0
SWITCH-DESIGNATOR 2:5.3, 2:6.6
SWITCH-IDENTIFIER 2:5.3, 2:6.2, 2:6.6
SWITCH-LIST 2:5.3
SYNTACTIC UNIT 2:1.0, 2:1.3
SYNTACTIC VARIABLE 2:1.0, 2:1.1
SYNTAX 2:1.0
SYSIN 3:5.0
SYSOUT 3:5.0
TERMINATED 3:3.0
TERMINATED STATE 2:6.7, 3:1.0
TEXT 2:2.2, 2:4.1, 2:5.1 2:5.5, 3:4.0, 3:5.0
TEXT ASSIGNMENT 3:4.0
TEXT EDITING AND DE-EDITING 3:4.0
TEXT EXPRESSIONS AND TEXT VALUES 2:6.8
TEXT GENERATION 3:4.0
TEXT HANDLING FACILITIES 3:4.0
TEXT HANDLING 3:5.0
TEXT OBJECT 2:6.8, 3:4.0, 3:5.0
TEXT QUOTE (A) 2:2.1, 2:4.2
TEXT REFERENCE 2:6.4

TEXT VALUES,COMPARISON 2:6.4
TEXTUAL LINK 1:2.0
TEXTUALLY ENCLOSING BLOCK 2:3.0
TEXT- 3:4.0
TEXT-CONSTANT 2:4.2, 2:4.2, 2:6.8
TEXT-EXPRESSION 2:6.0, 2:6.8
TEXT-FUNCTION-DESIGNATOR 2:6.8, 2:7.2
TEXT-REFERENCE-ASSIGNMENT 3:4.0
TEXT-VALUE 2:6.0, 2:6.8, 3:4.0
TEXT-VALUE-ASSIGNMENT 2:7.2, 3:4.0
TEXT-VARIABLE 2:6.8, 3:4.0
THE CONTROLLED VARIABLE 2:7.2
THE HIGHEST MARK 1:1.0
THE SYNTAX AND SEMANTICS OF SIMULA 2: .0
THE SYSTEM CLASS BASICIO 3:5.0
THE SYSTEM CLASS SIMSET 3:2.0
THE SYSTEM CLASS SIMULATION 3:3.0
THE USE OF BLANKS 2:2.4
THEN 2:2.2, 2:6.0, 2:7.2
THIS 2:2.2, 2:6.7, 2:7.1
THREE DIMENSIONAL ARRAY 2:5.2
THREE DOTS 2:1.7
TIME 3:3.0, 3:3.0
TIME AXIS 3:3.0
TIMING CLAUSE 3:3.0
TO 2:2.2
TRACE OF A MODEL 1:4.0
TRUE 2:2.2, 2:4.2, 2:6.4
TWO DIMENSIONAL ARRAY 2:5.2
TWO WAY LIST 3:2.0
TYING A DDNAME TO A FILE 3:5.0
TYPE 2:4.0, 2:4.1, 2:5.2
2:5.4, 2:5.5, 2:5.5
TYPE ARRAY 2:5.5
TYPE CONVERSION (ARITHMETIC) 2:5.1
TYPE DECLARATIONS 2:5.1
TYPE OF A QUANTITY 2:5.0
TYPE OF ARITHMETIC EXPRESSIONS 2:6.3
TYPES 2:4.1, 2:7.2
TYPES AND CONSTANTS 2:4.0
TYPE-DECLARATION 2:5.0, 2:5.1
UNDERSCORE (_) 2:2.1, 2:3.0
UNTIL 2:2.2, 2:7.2
UPPER SUBSCRIPT BOUND 2:5.2
UPPER-BOUND 2:5.2

VALUE 1:3.0, 2:2.2, 2:4.0 2:5.4, 2:5.5
VALUE COMPARATORS 2:6.4
VALUE OF CONTROLLED VARIABLE ON EXIT 2:7.2
VALUE-ASSIGNMENT 2:7.2
VALUE-ELEMENT 2:7.2
VALUE-EXPRESSION 2:7.2, 2:7.2
VALUE-TYPE 2:4.1, 2:5.1
VARIABLE 2:4.0, 2:6.0, 2:6.1 2:6.7, 2:7.2
VARIABLES 2:6.1
VERTICAL STROKE 2:1.5
VIRTUAL 2:2.2, 2:5.5
VIRTUAL ATTRIBUTES OF FILES 3:5.0
VIRTUAL-PART 2:5.5
VIRTUAL-QUANTITIES 2:5.5
VIRTUAL,ATTRIBUTE REDEFINITION 2:5.5
VIRTUAL,INNER ACCESSIBILITY 2:5.5
VIRTUAL,MATCHING ATTRIBUTE 2:5.5
VIRTUAL,VALIDITY OF A MATCH 2:5.5
WAIT 3:3.0
WHEN 2:2.2, 2:7.2
WHILE 2:2.2, 2:7.2
WHILE STATEMENTS 2:7.2
WHILE-STATEMENT 2:7.2