THIS FILE CONTAINS INFORMATION ON THE MERITSS SNOBOL SUBSYSTEM. THE SNOBOL INTERPRETER AVAILABLE ON MERITSS IS ADAPTED FROM THE ONE WRITTEN AT THE UNIVERSITY OF CALIFORNIA AT BERKELEY (CAL SNOBOL). THIS IMPLEMENTATION IS MORE OR LESS A SUBSET OF STANDARD SNOBOL AS DEFINED IN THE BOOK:


THIS BOOK IS AN EXCELLENT REFERENCE MANUAL DESCRIBING THE MACRO IMPLEMENTATION OF SNOBOL4 FOR THE IBM 360 COMPUTER. CAL SNOBOL IS NOT A MACRO IMPLEMENTATION, BUT INSTEAD A VERSION TAILORED FOR EFFICIENCY ON THE CDC 6000 SERIES MACHINES. THE MERITSS VERSION HAS BEEN EXTENDED FOR THE TIMESHRING ENVIRONMENT, ALLOWING LINE NUMBERS AND ENABLING INTERACTIVE INPUT/OUTPUT.

ALSO CONTAINED IN THIS FILE IS A DESCRIPTION OF THE DIFFERENCES IN CAL SNOBOL OVER STANDARD SNOBOL: CHARACTER SETS, ARITHMETIC, DATATYPES, PRIMITIVE FUNCTIONS, INPUT/OUTPUT FUNCTIONS AND KEYWORDS.

FURTHER SECTIONS DESCRIBE KNOWN BUGS, SNOBOL SIMULATED FUNCTIONS, RUNNING UNDER THE BATCH SUBSYSTEM, AND TIMESHRING CONVENTIONS FOR THE SNOBOL SUBSYSTEM. AN INDEX IS INCLUDED.

AN EXAMPLE SNOBOL PROGRAM EXISTS ON LIB FILE SN0PIG WHICH TRANSLATES ENGLISH TEXT TO PIG LATIN, A TYPICAL EXAMPLE OF A SNOBOL APPLICATION.

TO LIST THIS PROGRAM DO:

LIB SN0PIG
LNH

TO RUN THE PROGRAM DO:

LIB SN0PIG
RNH

A GOOD BOOK FOR BEGINNING SNOBOLERS IS:


CHARACTER SETS:

TWO CHARACTER SETS ARE AVAILABLE WHEN USING CAL SNOBOL ON MERITSS. BECAUSE THE REMOTE TERMINAL KEYBOARD IS SIMILAR TO THE CHARACTER SET DESCRIBED IN STANDARD SNOBOL, THIS SET IS DEFAULT. THERE ARE SOME DIFFERENCES, THE ONLY ONE AFFECTING CAL SNOBOL BEING THE CHARACTER FOR ALTERNATION IN PATTERNS: THIS IS AN UP ARROW: ↑.

THE OTHER CHARACTER SET CORRESPONDS TO THE CDC 6000 LINE PRINTER CHARACTER EQUIVALENTS ON REMOTE TERMINALS. THIS SET IS OBTAINED BY SPECIFYING AN 'X' ON THE RUN OR RNH COMMAND IN THE SNOBOL SUBSYSTEM, OR ON THE SNOBOL CONTROL CARD IN THE BATCH SUBSYSTEM.
CHARACTER SET TABLE:

<table>
<thead>
<tr>
<th>STANDARD</th>
<th>DEFAULT</th>
<th>MERITSS</th>
<th>'RNH,X' (BASED ON CDC 6000 LINE PRINTER EQUIVALENTS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-Z</td>
<td>A-Z</td>
<td>A-Z</td>
<td></td>
</tr>
<tr>
<td>0-9</td>
<td>0-9</td>
<td>0-9</td>
<td></td>
</tr>
<tr>
<td>PERIOD</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>COMMA</td>
<td>,</td>
<td>,</td>
<td></td>
</tr>
<tr>
<td>COLON</td>
<td>:</td>
<td>:</td>
<td></td>
</tr>
<tr>
<td>SEMICOLON</td>
<td>;</td>
<td>;</td>
<td></td>
</tr>
<tr>
<td>PLUS</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>MINUS</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>TIMES</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>DIVIDE</td>
<td>/</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>QUOTE</td>
<td>&quot;</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>SINGLE QUOTE</td>
<td>'</td>
<td>'</td>
<td></td>
</tr>
<tr>
<td>OPEN PAREN</td>
<td>(</td>
<td>(</td>
<td></td>
</tr>
<tr>
<td>CLOSE PAREN</td>
<td>)</td>
<td>)</td>
<td></td>
</tr>
<tr>
<td>LESS THAN</td>
<td>[</td>
<td>ØR &lt; ARRAY</td>
<td></td>
</tr>
<tr>
<td>GREATER THAN</td>
<td>] ØR &gt; BRACKETS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQUAL SIGN</td>
<td>=</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>VERTICAL BAR</td>
<td>‖</td>
<td>‖</td>
<td>USED FOR ALTERNATION IN PATTERNS</td>
</tr>
<tr>
<td>DOLLAR SIGN</td>
<td>$</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>AMPERSAND</td>
<td>&amp;</td>
<td>LINE FEED</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>AT SIGN</td>
<td>[</td>
<td>#</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>ØGICAL NOT</td>
<td>J</td>
<td>NONE</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>QUESTION MARK</td>
<td>?</td>
<td>&amp;</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>EXCLAMATION</td>
<td>NONE</td>
<td>&quot;</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>PERCENT</td>
<td>NONE</td>
<td>?</td>
<td>SEE NOTE BELOW</td>
</tr>
<tr>
<td>NUMBER SIGN</td>
<td>#</td>
<td>NONE</td>
<td>SEE NOTE BELOW</td>
</tr>
</tbody>
</table>

NOTE: THESE LAST 7 CHARACTERS ARE INVALID OPERATORS IN CAL SNØBØL.

ARITHMETIC:

1. THE MAXIMUM MAGNITUDE FOR AN INTEGER IS 10 DIGITS.
2. REAL NUMBERS EXIST IN THE RANGE 10**-294 TO 10**322 WITH 14 SIGNIFICANT DIGITS.
3. EXPONENTIATION OF INTEGERS DOES NOT WORK.
4. MIXED MODE ARITHMETIC IS ONLY ALLOWED BY USING THE FUNCTION 'CONVERT' EXPLICITLY. SEE BELOW.

DATATYPES:

STRINGS, SIMPLE VARIABLES, ARRAYS, PATTERNS, PROGRAMMER-DEFINED, AND CODE ARE THE DATATYPES IN CAL SNØBØL. TABLES ARE NOT IMPLEMENTED.

THE PRIMITIVE PATTERN STRUCTURE: SUCCEED, IS NOT PREDEFINED.
OPERATORS:

---

Of the unary operators: cursor position, interrogation, and negation are not implemented. Deferred evaluation (unevaluated expressions using the unary * ) may only be used with variables, not full expressions or function calls.

Of the binary operators, exponentiation of integers does not work.

All other operators are valid.

PRIMITIVE FUNCTIONS:

-------------

The functions arg, clear, collect, copy, dup, dupl, eval, field, integer, load, local, opsyn, remdr, replace, stoptr, table, trace, unload, and value are not implemented.

The following functions are unique to cal snobol (#1), have different calls (#2), or different definitions (#3) over standard snobol.

apply(f,x1,x2,...,xn) - apply assumes that extra arguments are thrown away; those missing are assumed to be null (#2).

array(string) - array takes only one argument -- an array prototype. All elements are initialized to the null string (#2).

clock() - clock returns a string which is the time of day in the form 'HH:MM:SS'. Any arguments are ignored (#1).

convert(x) - convert is only used for numeric datatype conversion. If x is real, a string is returned; if x is integer or a string of digits, possibly containing a period and/or leading sign, the corresponding real number is returned. Convert must be used in mixed mode arithmetic and to output real numbers (#3).

freeze(string, recnam) - freeze writes out a copy of the entire field length, which includes cal snobol and the user's program, in the format of an absolute overlay on the file named by the argument string, using recnam, if it is present (optional) as the ident name of the binary, and using 'ice' if it is not. It then terminates execution. This file can be saved and later executed by loading the file in the batch subsystem. Execution is resumed in the program at the point where freeze was called. Example: if the statement freeze('concord') is in a program, a file concord is created and snobol will end execution. The system command:

    save concord  

will save this copy of the file. To run the program where it left off, do:

    old concord  
x, concord

Freeze is useful if a large part of the program initializes strings and patterns before computation of results (#1).
IF(E1*E2*...*EN) - IF ALLOWS AN ARBITRARY NUMBER OF EXPRESSIONS TO GROUPED FOR EVALUATION IN ONE STATEMENT (#1).

ITEM(A,S1,S2,...,SN) - A IS AN ARRAY. IF THE OTHER ARGUMENTS ARE ALL INTEGERS AND A IS N-DIMENSIONAL, THEN ITEM DOES AN NRETURN OF THE NAME OF THE ELEMENT OF A WHOSE SUBSCRIPTS ARE THE SI. THE SI CAN ALSO BE STRINGS OF INTEGERS SEPARATED BY COMMAS. ITEM SCANS THE SI UNTIL ENOUGH INTEGERS TO SATISFY THE DIMENSIONALITY OF A HAVE BEEN FOUND - THE EXTRA ARE IGNORED; NULL STRINGS ARE ASSUMED FOR MISSING SUBSCRIPTS. ITEM FRETURNS IF ANY SUBSCRIPTS ARE OUT OF BOUNDS (#3).

NEXTVAR(X) - IN CAL SNØBØL, IT IS POSSIBLE TO SEQUENTIALLY ACCESS "FAMILIES" OF VARIABLES. EXAMPLES OF FAMILIES INCLUDE ALL ELEMENTS OF AN ARRAY, FIELDS IN A PROGRAMMER-DEFINED DATA STRUCTURE, AND MOST IMPORTANT, ALL SIMPLE VARIABLES IN EXISTENCE DURING A PROGRAM RUN: BOTH NATURAL AND CREATED.

NEXTVAR RETURNS THE NAME OF THE "NEXT" VARIABLE IN THE SAME FAMILY AS THE VARIABLE NAMED BY X. X MUST BE OF TYPE NAME OR STRING. NEXTVAR WILL TREAT A FAMILY OF VARIABLES CYCICALLY. FOR ARRAYS, ELEMENTS ARE SELECTED BY THE ORDERED SET OF INTEGER SUBSCRIPTS (#1).

PROTOTYPE(X) - EXISTS IN CAL SNØBØL AS A GENERALIZATION OF THE FUNCTIONS ARG, FIELD, AND LOCAL IN STANDARD SNØBØL. HENCE PROTOTYPE IS DESIGNED TO RETURN THE "PROTOTYPE" DEFINITION FOR VARIABLES OF NEARLY ALL DATATYPES (#1).


2. IF X IS A PROGRAMMER-DEFINED DATA STRUCTURE, PROTOTYPE RETURNS THE <DATA PROTOTYPE> THAT WAS USED IN THE DATA FUNCTION CALL WHEN X WAS DEFINED.

3. IF X IS A PATTERN, PROTOTYPE RETURNS ONE OF A FIXED SET OF PATTERN "PROTOTYPE". THESE STRINGS ARE IN THE FORM OF PROTOTYPES FOR PROGRAMMER-DEFINED DATA STRUCTURES. TO FURTHER THE ANALOGY BETWEEN PATTERNS AND PROGRAMMER-DEFINED DATA STRUCTURES, A SET OF PSEUDO-FIELD FUNCTIONS, CORRESPONDING TO THE FIELD NAMES IN THE PATTERN PROTOTYPES, HAVE BEEN IMPLEMENTED. THESE PROCEDURES ALLOW A PATTERN TO BE DECOMPOSED INTO THE OBJECTS USED TO BUILD THE PATTERN. IF X IS ARB, REM, BAL, FAIL, ABORT, OR FENCE, PROTOTYPE RETURNS THE STRING: ARB(), REM(),... NO PSEUDO-FIELD FUNCTIONS EXIST FOR THESE PATTERNS.

IF X IS A LEN, POS, RP0S, TAB, RTAB, ANY, NOTANY, SPAN, BREAK, OR ARBNØ PATTERN, PROTOTYPE RETURNS THE STRING: Y(PARAM)
WHERE Y IS EITHER LEN, POS, ETC. THE PSEUDOFIELD FUNCTION
PARAM MAY BE USED ON ANY OF THESE PATTERNS: E.G.
PARAM(LEN('3')) RETURNS THE INTEGER 3.
PARAM(ANY('CBAC')) RETURNS THE STRING ABC.
PARAM(ARBN0('CBAC')) RETURNS THE STRING CBAC.

IF X IS A CONCATENATED OR ALTERNATED PATTERN, PROTOTYPE
RETURNS THE STRING: CAT(FIRST,REST) OR ALT(FIRST,REST).
THE PSEUDOFIELD FUNCTIONS FIRST AND REST ARE DEFINED ON
THE CLASS OF CONCATENATED AND ALTERNATED PATTERNS: E.G.
FIRST('A' 'B' 'C') RETURNS THE STRING A.
REST('A' 'B' 'C') RETURNS THE PATTERN WHICH IS THE
VALUE OF THE EXPRESSION ('B' 'C').
FIRST(ARB BAL REM) RETURNS THE PATTERN ARB.

IF X IS A "NAMING" PATTERN, E.G. $P$ R OR R . P, PROTOTYPE
RETURNS THE STRING DOL(LEFT,RIGHT) OR PRD(LEFT,RIGHT): E.G.
LEFT($P$ R) RETURNS THE PATTERN P.
RIGHT($P$ R) RETURNS THE NAME OF THE VARIABLE R.

IF X IS A DEFERRED EVALUATION PATTERN, I.E. *V*, PROTOTYPE
RETURNS THE STRING STAR(RIGHT). RIGHT(X) RETURNS THE NAME
OF THE VARIABLE V.

IF X CAN ALSO BE A NAME OF A VARIABLE. SEE THE EXPLANATION OF
FAMILIES OF VARIABLES IN THE DEFINITION OF THE FUNCTION
NEXTVAR ABOVE.

IF X IS A SIMPLE VARIABLE, PROTOTYPE RETURNS THE STRING
INDIRECT(RIGHT). RIGHT(X) RETURNS THE STRING CORRESPONDING
TO THE SIMPLE VARIABLE NAMED BY X: E.G.
RIGHT(.VAR) RETURNS THE STRING VAR.

IF X IS AN ARRAY ELEMENT OR A PROGRAMMER-DEFINED DATA
STRUCTURE FIELD, PROTOTYPE RETURNS THE STRING
ITEM(FAMILY,SELECT0R) OR APPLY(SELECT0R,FAMILY). FAMILY(X)
RETURNS THE ARRAY OR DATA STRUCTURE OBJECT WHICH CONTAINS THE
VARIABLE NAMED BY X; SELECTOR RETURNS A STRING WHICH CAN BE
USED TO SELECT THIS VARIABLE IN FAMILY(X).
ITEM(FAMILY(.AI,J1),SELECTOR(.AI,J1)) IS EQUIVALENT TO
AI+J1.
APPLY(SELECTOR(.S0N(NODE)),FAMILY(.S0N(NODE))) IS EQUIVALENT
TO S0N(NODE).

REMARK(STRING) - REMARK PLACES STRING IN THE DAYFILE. REMARK RETURNS
THE NULL STRING (#1).

TYPE(X) - TYPE IS EQUIVALENT TO DATATYPE(X) EXCEPT FOR THE CASE WHEN X
IS A PROGRAMMER-DEFINED DATATYPE (DATA STRUCTURE). TYPE THEN
RETURNS THE STRING "DATA", WHEREAS DATATYPE RETURNS THE DATA
STRUCTURE NAME STRING WHICH WAS DEFINED IN THE PROTOTYPE TO
THE FUNCTION DATA (#1).
INPUT/OUTPUT FUNCTIONS:

THE FUNCTIONS BACKSPACE AND ENDFILE ARE NOT AVAILABLE.

DETACH(VNAME) - REMOVES THE I/O ASSOCIATION THAT THE STRING VNAME (A VARIABLE) MIGHT HAVE.

ENDGROUP(FNAME,I) - WRITES AN END OF RECORD MARK ON THE FILE REPRESENTED BY THE STRING FNAME WITH THE LEVEL SPECIFIED BY INTEGER, WHICH CAN BE DETECTED LATER BY THE FUNCTION EORLEVEL. INTEGER MUST BE IN THE RANGE 0 TO 14 (WHICH CORRESPONDS TO AN END OF RECORD) OR 15 (CORRESPONDING TO AN END OF FILE).

E0I(FNAME) - SUCCEEDS AND RETURNS THE NULL STRING IF THE FILE REPRESENTED BY THE STRING FNAME IS POSITIONED OVER AN END OF INFORMATION. E0I FAILS OTHERWISE.

EORLEVEL(FNAME) - RETURNS THE RECORD LEVEL (0 OR 15) OF THE END OF RECORD MARK WRITTEN ON THE FILE REPRESENTED BY THE STRING FNAME IF THE FILE IS POSITIONED OVER AN END OF RECORD. IF POSITIONED OVER AN END OF INFORMATION, -1 IS RETURNED (E0I WOULD SUCCEED IN THIS CASE). OTHERWISE EORLEVEL FAILS. SEE "KNOWN BUGS" CONCERNING KRØNØS END OF FILE LEVELS.

INPUT(VNAME, FNAME, LENGTH) - INPUT ASSOCIATES THE VARIABLE REPRESENTED BY THE STRING VNAME TO THE FILE REPRESENTED BY THE STRING FNAME. LENGTH IS AN INTEGER SPECIFYING THE NUMBER OF CHARACTERS TO BE READ FROM EACH RECORD ON THE FILE. IF VNAME ALREADY HAS AN I/O ASSOCIATION, IT IS DETACHED. INPUT RETURNS THE NULL STRING AS ITS VALUE.

OUTPUT(VNAME, FNAME, PREFIX) - OUTPUT ASSOCIATES THE VARIABLE REPRESENTED BY THE STRING VNAME TO THE FILE REPRESENTED BY THE FILE FNAME. PREFIX IS A STRING OF LENGTH 0 OR 1 WHICH IS CONCATENATED TO THE BEGINNING OF EACH RECORD WRITTEN TO THE FILE. OUTPUT DETACHES ANY PREVIOUS I/O ASSOCIATIONS THAT VNAME HAS AND RETURNS THE NULL STRING. OUTPUT ASSOCIATED VARIABLES ARE NOT FORMATTED INTO "LINES".

REWIND(FNAME) - REWIND REPOSITIONS A FILE REPRESENTED BY THE STRING FNAME TO ITS BEGINNING OF INFORMATION. REWIND WRITES AN END OF RECORD BEFORE REWINDING. REWIND RETURNS THE NULL STRING AS ITS VALUE.

THE INPUT ASSOCIATION INPUT('INPUT','INPUT',80) AND THE OUTPUT ASSOCIATION OUTPUT('OUTPUT','OUTPUT','') ARE PREDEFINED, ENABLING THE SNØBØL USER TO IMMEDIATELY USE THE VARIABLES INPUT AND OUTPUT TO REFER TO THE TERMINAL. IT IS HELPFUL TO USE THE TRIM FUNCTION TO TRIM TRAILING BLANKS UPON INPUT.

THE VARIABLE PUNCH DOES NOT HAVE A PREDEFINED ASSOCIATION IN CAL SNØBØL. OF COURSE MERITSS DOES NOT HAVE A CARD PUNCH.
KEYWORDS:

Here are implemented as primitive functions in CAL SNOBOL. The only ones available are:

**ALPHABET()** returns as its value the 63 characters of the CDC6000 series character set in their standard collating sequence (in numerical order by display code). Two of these give trouble on MERITSS as they are ASCII escape characters.

**ANCHOR(X)** turns on the anchored mode of pattern matching if X is any non-null expression; turns off anchored mode otherwise. Initially SNOBOL pattern matching is not in anchored mode. Example: ANCHOR(1) turns it on; ANCHOR(0) turns it off.

**FNCLEVEL()** returns an integer corresponding to the number of functions called but have not yet returned.

**MAXLENGTH(I)** returns the value corresponding to the maximum allowable length string in SNOBOL if I = 0; for I a positive integer, this maximum is reset to the value I. Initially MAXLENGTH(0) has the value 131070.

**STCOUNT(I)** returns the running total of the number of statements which have been executed so far if I = 0; for I a positive integer this value is reset to I.

**STLIMIT(I)** returns the value corresponding to the maximum number of statements allowed to be executed before execution is halted if I = 0; for I a positive integer this value is reset to I. Initially STLIMIT(0) has the value 1000000.

There are no protected keywords in CAL SNOBOL (e.g. &ERRTYPE).
KNOWN BUGS:

1. Garbage collection occurs in two phases: list storage compaction and heap block compaction. List storage compaction works properly. However, heap block compaction still contains numerous bugs and is turned off. (See No. 5, below.) When string space fills up, garbage collection is invoked. If there is not enough space to garbage collect, Snobol will error terminate with the message 'the maximum field length has been exceeded'. An increase in field length usually fixes this. When running a program, extra memory (up to a total of 52200b) can be allotted by using the mi or ma parameter on the run command under the Snobol subsystem, or by running under the batch subsystem, using the f parameter on the Snobol card.

2. Exponentiation of integers does not work. (String arithmetic is not yet implemented.)

3. Deferred evaluation (Unary * operator for unevaluated expressions) applied to anything other than a variable is not detected as an inappropriate Cal Snobol expression. To perform deferred evaluation of a function, for example, f(x), where f does not return a variable, use an nreturn in f.

4. Level 15 end of records, equivalent to an end of file, are not sensed properly by the Snobol function eorlevel. (Endgroup will write an end of file if its record level argument is 15). This information is not available under the Kronos operating system, and appears to eorlevel as two consecutive level 0 end of records.

5. Documentation from Berkeley accompanying the present version of Cal Snobol states that heap block compaction is a major source of bugs ("and may always be"). Heap block compaction is therefore turned off. It may be enabled by specifying h=1 on the Snobol control card under the batch subsystem, and making a call to the primitive function HBC (there are no arguments to HBC) at the point in the program where HBC is desired. A message - "heap block compaction called." - is printed in the dayfile every time HBC is called.

However, use of heap block compaction is not advised, except for experimentation, and results cannot be guaranteed.
SNØBØL SIMULATED FUNCTIONS:

-----------------------------

BECAUSE SEVERAL STANDARD SNØBØL FUNCTIONS ARE NOT PREDEFINED IN CAL SNØBØL, AN ATTEMPT HAS BEEN MADE AT THE UNIVERSITY OF MINNESOTA TO WRITE SOME OF THESE FUNCTIONS IN SNØBØL IN AN EFFICIENT MANNER AND TO PLACE THEM ON A FILE FOR EASY ACCESS TO SNØBØL USERS. OTHER USEFUL NON-STANDARD FUNCTIONS ARE MADE AVAILABLE ALSO. THIS FILE MAY BE GOTTEN BEFORE A SNØBØL PROGRAM IS RUN AND THEN READ IN AND CODED AT EXECUTION TIME IN THE SNØBØL PROGRAM USING THE CODE FUNCTION.

AT THE PRESENT TIME, ON THE LIB FILE SNØLIB, THE FUNCTIONS

GET     ANCHR     DUPL     EVAL
INTEGER REPLACE TABLE REMDR
TRACE   STØPTR    DUMP    CLEAR
REVERSE LFN        SØRTI   SØRTØ
STATUS   RANDØM    MODIFY  ASCII

ARE AVAILABLE. IN ADDITION, THE PRIMITIVE PATTERN STRUCTURE: SUCCEED IS GIVEN A VALUE. TO ACCESS THESE FUNCTIONS ENTER THE FOLLOWING LINES OF CODE AT THE BEGINNING OF YOUR PROGRAM:

00100 INPUT('SNØ..1','SNØLIB',1000)
00110 REWIND('SNØLIB')
00120 SNØLIB = CODE(TRIM(SNØ..1))<SNØLIB>
00130 OUTPUT = 'SNØLIB NOT FOUND' :(END)
00140 SNØLIB.C0DED GET()
00150 * CONTINUE WITH YOUR PROGRAM BELOW

THE VARIABLES 'SNØ..1', 'SNØLIB', 'SNØLIB', AND THE LABEL 'SNØLIB.C0DED' SHOULD BE USED EXACTLY AS GIVEN HERE FOR BEST RESULTS. ALSO, THE FUNCTION GET IS NEEDED AS THE LIBRARY MANAGER.

YOUR PROGRAM MAY NOW OPERATE UNDER THE ASSUMPTION THAT THE FUNCTIONS REQUESTED THROUGH THE GET FUNCTION ARE AVAILABLE AS PART OF CAL SNØBØL.

SEE THE DESCRIPTION OF THE GET FUNCTION, BELOW, FOR ITS PURPOSE AND OPERATION, AND FOR THE DESCRIPTION OF THE STRUCTURE OF SNØLIB.

WHEN RUNNING YOUR PROGRAM DO A :

GET,SNØLIB/UN-LIBRARY

BEFORE YOU FIRST ENTER 'RUN' OR 'RNH'. THE ABOVE STATEMENT:

00110 REWIND('SNØLIB')

ENSURES POSITIONING OF SNØLIB AT THE BEGINNING FOR EVERY RUN OF YOUR PROGRAM. SNØLIB MUST BE ON A LOCAL FILE NAMED 'SNØLIB'.

THE SNØLIB FUNCTIONS ARE DESCRIBED BELOW. THE IMMEDIATELY FOLLOWING LIST IS IN THE SAME ORDER AS THE FUNCTIONS IN THE LIBRARY. FOR FUNCTIONS IDENTICAL IN OPERATION TO STANDARD SNØBØL THE READER IS DIRECTED TO THE BOOK:

'THE SNØBØL PROGRAMMING LANGUAGE' BY GRISWOLD ET AL.

PAGE 9
"STANDARD" FUNCTIONS:

GET(STRING) [CONTAINS ANCHR()]
DUPL(STR) I
EVAL(STR)
INTEGER(STR)
REPLACE(STR1, STR2, STR3)
TABLE(VAR, CHAR)
REMDR(X, Y)
TRACE(VAR, FILE, TAG)
STOPTR(VAR)
DUMP(MESG, FILE, TERM)
CLEAR()

[ THE PATTERN 'SUCCEED' ]
REVERSE(STR)

"EXTENDED" FUNCTIONS:

LFN(TESTFN)
SORTI(STRING, DELIMITER, PROTECTOR)
SORTD(DELETE)
STATUS(INFO, FILE)
RANDOM(RANGE, COUNT, DELIM, SEED)
MODIFY(MESS, FIX)

GET(STRING) - THE GET FUNCTION, ALTHOUGH NOT A STANDARD SNØBØL
FUNCTION, IS THE BASIS OF SNØLIB, AND IS USED TO DEFINE AND
CODE SUBSEQUENT FUNCTIONS. (GET IS ANALOGOUS TO 'LOAD' IN
STANDARD SNØBØL.)

STRING IS A LIST OF FUNCTION NAMES SEPARATED BY COMMAS OR
NULL. IF NULL, ALL "STANDARD" FUNCTIONS IN SNØLIB ARE DEFINED
AND CODED.

ALTERNATELY, IF THE ARGUMENT GIVEN IS AN ASTERISK ('*'), ALL
FUNCTIONS INCLUDING THE ONES CALLED "EXTENDED" FUNCTIONS ARE
READ AND CODED.

STRING MAY BE A LIST OF AVAILABLE FUNCTION NAMES SEPARATED BY
COMMAS ('*') AND THEN GET WILL ONLY DEFINE AND CODE THE
FUNCTIONS REQUESTED, RETURNING AS ITS VALUE THE LIST OF ANY
FUNCTION NAMES THAT ARE NOT IN SNØLIB, COMPARING THE USERS
LIST AGAINST AN INTERNAL LIST. IN THIS WAY GET DOES NOT READ
THE REST OF SNØLIB. THIS Is A METHOD OF CONSERVING CORE SPACE
IN LIEU OF CODING ALL FUNCTIONS WHEN ALL ARE NOT NEEDED.

SNØLIB IS ORGANIZED WITH THE NAME OF A FUNCTION ON ONE LINE
FOLLOWED BY THE ENTIRE FUNCTION ON THE NEXT LINE, WITH THE
MOST FREQUENTLY USED FUNCTIONS AT THE BEGINNING, AND THE
LARGEST AND LEAST USED FUNCTIONS AT THE END. THIS IS SO THAT
THE NORMAL USER NEED NOT READ IN ALL OF SNØLIB TO SATISFY HIS
REQUESTS, WHILE ALL AVAILABLE FUNCTIONS MAY BE OBTAINED
THROUGH GET JUST AS EASILY.
IN THE EXAMPLE PROGRAM ABOVE, THE LINE
00140 SNOLIB.CODED GET()
PROVIDES THE RE-ENTRY LABEL FOR THE DEFINITION OF THE GET
FUNCTION, AND THIS EXACT LABEL MUST BE PRESENT, ALONG WITH A
CALL TO GET ON THE SAME LINE AS THE RE-ENTRY LABEL OR
SUBSEQUENT TO IT. AS DISCUSSED ABOVE, A CALL TO GET MAY BE
NULL, OR MAY CONTAIN A STRING OF FUNCTIONS TO GET, AS:
00140 SNOLIB.CODED GET('DUFL,LFN,DUMP')

GET MAY BE CALLED MORE THAN ONCE, IN FACT SOME OF THE
FUNCTIONS THEMSELVES MAKE CALLS TO GET TO COMPLETE THEIR
DEFINITIONS (SEE SNOLIB NOTES). GET RETURNS AS ITS VALUE ONLY
THE NAMES OF FUNCTIONS SPECIFICALLY DESIRED WHICH IT CANNOT
FIND IN SNOLIB. GET WILL NOT CODE THE SAME FUNCTION TWICE.

ANCHR() - GET DEFINES 'ANCHR()' (WHICH HAS NO ARGUMENTS) FOR ITS OWN
NEEDS, AND ANCHR MAY BE USED FREELY BY THE SNOLIB USER. ANCHR
RETURNS THE NULL STRING IF THE ANCHOR IS OFF, AND RETURNS '1'
IF THE ANCHOR IS ON. (SEE 'KEYWORDS' FOR 'ANCHOR()').

DUFL(STR,N) - RETURNS STR CONCATENATED WITH ITSELF N TIMES; RETURNS
THE NULL STRING IF N<0; RETURNS IF N<0 (SEE GRISWOLD ET AL.).

REVERSE(STR) - RETURNS AS ITS VALUE THE ARGUMENT WITH ITS CHARACTERS
REVERSED.

EVAL(STR) - RETURNS THE VALUE OF STR (ANY EXPRESSION) BY EXECUTING THE
CODE FUNCTION UPON THE STRING. EVAL FRETURNS IF THE
EVALUATION OF THE EXPRESSION FAILS. UNEVALUATED EXPRESSIONS
(USE OF THE UNARY OPERATOR '*' BEFORE AN EXPRESSION) CANNOT BE
HANDEDLED BY EVAL, UNLIKE THE EVAL IN GRISWOLD. NOR DOES EVAL
CHECK SYNTAX.

INTEGER(STR) - THIS PREDICATE TEST SUCCEEDS IF AND ONLY IF THE VALUE
OF STR IS A SIGNED OR UNSIGNED INTEGER. (THE NULL STRING IS
EQUIVALENT TO ZERO AND SUCCEEDS. SEE GRISWOLD ET AL.).

REPLACE(STR1,STR2,STR3) - PERFORMS A ONE FOR ONE CHARACTER
REPLACEMENT. (SEE GRISWOLD ET AL. FOR A FULL DISCUSSION.)

TABLE(VAR,CHAR) - CREATES A TABLE NAMED VAR. TABLE ELEMENTS ARE
SUBSEQUENTLY REFERENCED AS FUNCTION CALLS WITH PARENTHESES '(',
')'. BECAUSE TABLES ARE SIMULATED BY USING INDIRECT
REFERENCE, A PREFIX CHARACTER IS SUPPLIED TO PROTECT TABLE
ENTRIES. CHAR ALLOWS THE USER TO SPECIFY THIS CHARACTER; THE
DEFAULT IS AN EXCLAMATION POINT ('!'). EXAMPLE:
TABLE('TEXT') - CREATES A TABLE WHOSE ENTRIES MIGHT BE
TEXT('WORD'), TEXT('AN'), ETC. (AVOID CREATING TABLES OF
THE SAME NAME AS OTHER FUNCTIONS AND NOTE THAT THE 'CONVERT'
FUNCTION IN CAL SNOBOL DOES NOT OPERATE ON ARRAYS OR TABLES.)

REMDR(X,Y) - RETURNS THE INTEGER REMAINDER OF THE DIVISION OF X BY Y.
(SEE GRISWOLD ET AL.)
SUCCEED - NOT AN ACTUAL FUNCTION, BUT A PRIMITIVE PATTERN. SUCCEED IN A PATTERN MATCH ALWAYS ENSURES SUCCESS AT THE POINT IT IS ENCOUNTERED. (SEE GRISWOLD ET AL.)

STATUS(INFO, FILE) - DISPLAYS THE VALUES OF TIME, CLOCK, DATE, FNCLEVEL, STCOUNT, STLIMIT, AND MAXLENGTH UPON THE FILE. DEFAULT 'OUTPUT', WITH THE OPTIONAL INFO (ANY STRING OF CHARACTERS AS A 'LABEL'), AND RETURNS THE NULL STRING. (FNCLEVEL IS ADJUSTED (FNCLEVEL() - 1) FOR THE LEVEL OF STATUS ITSELF.)

TRACE(VAR, FILE, TAG)
ST0PTR(VAR) - TRACE IS A TRACE BY OUTPUT ASSOCIATION OF NATURAL VARIABLES ONLY; ST0PTR MAY BE CALLED TO DETACH THE VARIABLE FOR WHICH TRACE HAS BEEN CALLED.

VAR - THE VARIABLE TO BE TRACED (OR ST0PTRACED); MUST BE A NATURAL VARIABLE EXPRESSED AS A STRING.
FILE - THE FILE TO RECEIVED THE TRACE OUTPUT, DEFAULT 'OUTPUT'.

ST0PTR IS DEFINED WHENEVER TRACE IS. UNFORTUNATELY, TRACE CANNOT PROVIDE MORE INFORMATION ABOUT THE STATE OF A VARIABLE, SUCH AS TRACE IN STANDARD SN0B0L.

LFN(TESTFN) - RETURNS THE SAME LOGICAL FILE NAME AS THE ARGUMENT TESTFN IF IT IS A LEGAL KRONOS OPERATING SYSTEM FILE NAME. OTHERWISE, LFN RETURNS.

SORTI(STRING, DELIMITER, PROTECTOR)
SORT0(DELETE) - SORTI AND SORT0 ARE USED TOGETHER TO DO IN-CORE LEXICAL SORTING OF STRINGS. ONLY SORTI NEEDS TO BE OBTAINED VIA GET, AS SORTI AND SORT0 ARE DEFINED TOGETHER. THE PARAMETERS DELIMITER, PROTECTOR, AND DELETE ARE OPTIONAL ONES USED TO ALLOW FLEXIBILITY. SORTI AND SORT0 ARE FIRST DISCUSSED FOR THE CASE IN WHICH THESE PARAMETERS ARE LEFT NULL.

IN GENERAL, STRINGS ARE INSERTED ONE AT A TIME VIA SORTI INTO A MASTER LIST; WHEN NO MORE STRINGS ARE TO BE ADDED, SORT0 MAY BE USED TO RETRIEVE STRINGS ONE AT A TIME IN LEXICAL ORDER. WHEN ALL STRINGS HAVE BEEN RETRIEVED, SORT0 FAILS. INSERTION AND RADIX METHODS ARE USED.

THIS SORT HAS BEEN FOUND TO BE EFFICIENT IN TIME AND STRING SPACE USED.
THE PARAMETERS WHICH ALLOW CHANGES IN THE USE OF SORTI AND SORTO SPECIFY SPECIAL CHARACTERS, SELECT DELETION OF MEMBER STRINGS RETURNED THROUGH SORTO, AND REINITIALIZE THE SORT FUNCTION PAIR FOR ANOTHER SORT JOB IN THE SAME PROGRAM RUN.

STRING - THE LEADING CHARACTER OF EACH STRING MUST BE ALPHANUMERIC, AND FOLLOWING CHARACTERS MAY BE ANY CHARACTER IN THE CDC CHARACTER SET OTHER THAN THE DELIMITER CHARACTER. THE STRINGS MAY BE OF ANY LENGTH.

DELIMITER - USED DURING THE SORT TO DELIMIT SUBSTRINGS; SHOULD NOT OCCUR AS ANY CHARACTER IN THE STRINGS TO BE SORTED. IF NOT SPECIFIED, A RIGHT PARENTHESIS (') IS USED. DELIMITER MUST BE A SINGLE CHARACTER.

PROTECTOR - USED IN AN INDIRECT REFERENCE DURING THE SORT TO MAINTAIN UP TO 36 DISTINCT SUB-LISTS. IF NOT SPECIFIED A BACKSLASH (\) IS USED.

DELETE - ANY NON-NULL ARGUMENT IN THE FIRST CALL TO SORTO WILL SET THE CONDITION FOR DELETING STRINGS RETURNED DURING THE SORT.

SORTI AND SORTO MAY BE USED AGAIN DURING A RUN ONCE THE LIST HAS BEEN EXHAUSTED (SORTO FRETurns) BY ADDING TO THE LIST (IF IT WAS NOT DELETED) OR STARTING ON A NEW LIST, ALLOWING RESPECIFICATION OF PARAMETERS OTHER THAN STRING UPON THE INITIAL CALL TO A FUNCTION IN THIS PART OF THE JOB.

THE SORT IgNORES STRINGS WHICH ARE EXACT DUPLICATES OF STRINGS ALREADY IN THE SET, STRINGS WITH NON-ALPHANUMERIC FIRST CHARACTERS, AND THE NULL STRING.

RANDOM(RANGE,COUNT,DELIM,SEED) - RETURNS A STRING OF ONE OR MORE RANDOM INTEGERS UNIFORMLY DISTRIBUTED ON THE INTERVAL ('0:9') OR THE RANGE SPECIFIED BY THE CALL. EACH NUMBER IS FOLLOWED BY A SPACE OR THE DELIMITER GIVEN AS A LITERAL IN THE CALL.

RANGE - IS GIVEN SIMILAR TO THE FORM OF AN ARRAY RANGE - THAT IS, A SIMPLE INTEGER WILL SET THE EXCLUSIVE UPPER LIMIT OF THE VALUE OF THE NUMBER(S) RETURNED, AND AN ARGUMENT OF THE FORM 'J:K' IS USED TO SET THE LOWER BOUND OTHER THAN ZERO (BOUNDS MAY BE NEGATIVE OR POSITIVE OR BOTH) AND AN INCLUSIVE UPPER LIMIT.

THIS FORM MUST BE GIVEN AS A STRING (IN QUOTES OR A NAME OF A LITERAL) AND THE LOWER BOUND (J) SEPARATED FROM THE UPPER BOUND (K) BY A COLON - '1'. DEFAULT VALUE FOR RANGE IS '0:9'.

COUNT - IF GREATER THAN ONE, RANDOM WILL RETURN A STRING OF RANDOM NUMBERS (NOT AN ARRAY) DELIMITED BY A BLANK (' ') OR BY AN OPTIONAL CHARACTER SPECIFIED BY DELIM. IF THE COUNT IS LESS THAN OR EQUAL TO ZERO, THE COUNT IS SET TO ONE.

DELIM - GIVEN AS A LITERAL, IS USED AS THE DELIMITER. A SPECIAL CHARACTER - 'N' - IS A CUE TO USE THE NULL STRING AS THE DELIMITER. IF COUNT IS ONE, DELIM IS SET TO NULL.
SEED - RANDOM USES THE TIME FUNCTION TO SEED ITSELF: IF AN INTEGER IS PRESENT AS THE FOURTH ARGUMENT, IT IS USED TO SEED THE RANDOM NUMBER ALGORITHM. IF ANYTHING ELSE NON-NULL IS PRESENT, RANDOM RE-SEEDS ITSELF FROM THE TIME FUNCTION. A SEQUENCE OF THE ALGORITHM CONTINUES UNTIL A NEW SEED VALUE OR RESEED PARAMETER IS GIVEN.

FOR EXAMPLE: RANDOM('1:52', 10) RETURNS A STRING OF TEN NUMBERS DELIMITED BY BLANKS IN THE RANGE 1 TO 52.

DUMP(MESS, FILE, TERM) - DUMP USES NEXTVAR TO CYCLE THROUGH THE NATURAL VARIABLES IN THE SNOBOL SYSTEM AND LIST THE VALUE OF STRINGS AND NUMBERS OR THE DATATYPE OF A VARIABLE.

MESS - ANY STRING OF CHARACTERS USED AS A LABEL TO IDENTIFY THE DUMP.

FILE - A LOGICAL FILE NAME (DEFAULT 'OUTPUT' TO RECEIVE THE DUMP.

TERM - IF THE THIRD PARAMETER IS NULL, TRANSFER IS BACK TO THE CALLING STATEMENT; IF NON-NULL, TRANSFER IS TO THE END STATEMENT.

DUMP REFERENCES ALL NATURAL VARIABLES. WHEN IT REFERENCES A VARIABLE WITH AN INPUT ASSOCIATION, A NEW INPUT ASSIGNMENT IS MADE TO THE VARIABLE. THEREFORE IT IS IMPORTANT THAT INPUT VARIABLES BE DETACHED TO AVOID UNEXPECTED RESULTS. REDEFINING THEM UPON RETURN FROM DUMP MAY LOSE THE REMAINDER OF THE FILE I/O BUFFER, HOWEVER. THE VARIABLE 'INPUT' IS DETACHED UPON ENTRY TO DUMP, AND REDEFINED AS ('INPUT', 'INPUT', 80) IF TRANSFERRING BACK TO THE CALLING STATEMENT.

DUMP DOES NOT DUMP THE VALUES OF THE VARIABLES IN SNOLIB ITSELF, AS THEY ARE EXCESSIVELY LONG FOR THE TELETYPewriter.

CLEAR() - CLEAR SEARCHES FOR ALL NATURAL VARIABLES AND SETS THEM TO THE NULL STRING. THE PRIMITIVE PATTERNS ABORT, ARB, BAL, FAIL, FENCE, REM, SUCCEED ARE INCLUDED AS LOCAL VARIABLES IN THE DEFINITION OF CLEAR TO PROTECT THEM. CLEAR HAS NO ARGUMENTS. VARIABLES WITH AN OUTPUT ASSOCIATION SHOULD BE DETACHED BEFORE CALLING CLEAR. ('OUTPUT' IS DETACHED AND REDEFINED AS OUTPUT('OUTPUT', 'OUTPUT', '').)

CLEAR WILL NOT AFFECT THE VARIABLES IN SNOLIB ITSELF, DUE TO A PROTECTION ARRANGEMENT.

IF THERE ARE OTHER VARIABLES TO PROTECT, CLEAR SHOULD BE REDEFINED, DECLARING ALL VARIABLES TO BE RESTORED UPON RETURN AS LOCAL VARIABLES.

EXAMPLE: DEFINE('CLEAR() REM, BAL, MYVAR, NOJUNK, GOODSTUFF')

MODIFY(MESS, FIX) - MODIFY RETURNS AS ITS VALUE THE MODIFICATION (A LA XEDIT'S MODIFY DIRECTIVES) OF THE STRING MESS BY THE STRING FIX. THE LOGIC OF MODIFY, INCLUDING THE ACTION OF SPECIAL CHARACTERS, IS IDENTICAL IN BEHAVIOR TO THE LOGIC OF THE MODIFY
DIRECTIVES IN XEDIT, BUT MODIFY CONTAINS NO INPUT/OUTPUT AND THE USER WOULD ARRANGE HIS OWN SCHEME FOR INPUTTING THE FIX STRING ALIGNED PROPERLY WITH THE MESS. (MERELY PRINT OUT THE MESS SHIFTED TWO BLANKS TO THE RIGHT TO LINE UP WITH A SUBSEQUENT INPUT REQUEST.)

THE READER IS REFERRED TO XEDIT/UN-LIBRARY, FOR A FULL DISCUSSION OF THE MODIFICATION DIRECTIVES. (THE UCC LIBRARIAN STOCKS COPIES IN THE UCC REFERENCE CENTER, AND THEY ARE ALSO AVAILABLE IN ROOM 140 EXPERIMENTAL ENGINEERING.) BRIEFLY, THE DIRECTIVES ARE:

<table>
<thead>
<tr>
<th>DIRECTIVE</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) <em>STRING</em></td>
<td>CAUSES THE STRING OF CHARACTERS BETWEEN THE * AND THE NEXT # TO BE INSERTED BEFORE THE CHARACTER POINTED TO BY THE * AN &amp; OR ✯ WITHIN THE STRING IS TREATED AS A REGULAR CHARACTER.</td>
</tr>
<tr>
<td>2.) #</td>
<td>(WHEN NOT THE FIRST # AFTER AN ✯) CAUSES THE CHARACTER ABOVE IT TO BE DELETED.</td>
</tr>
<tr>
<td>3.) (SPACE)</td>
<td>A SPACE BELOW A CHARACTER LEAVES IT UNCHANGED.</td>
</tr>
<tr>
<td>4.) &amp;</td>
<td>REPLACES THE CHARACTER ABOVE IT WITH A SPACE.</td>
</tr>
<tr>
<td>5.) ANY OTHER CHARACTER REPLACES THE CHARACTER ABOVE IT.</td>
<td></td>
</tr>
</tbody>
</table>

IF NO # APPEARS AFTER AN ✯ ONE IS ASSUMED AFTER THE LAST NON-BLANK CHARACTER OF THE DIRECTIVE LINE.

ASCII(OCTAL·NUM) - RETURNS AS ITS VALUE THE ASCII CHARACTER WHOSE CHARACTER CODE IS SPECIFIED AS THE ARGUMENT. THE ARGUMENT MAY BE A TWO OR FOUR DIGIT OCTAL NUMBER (A DISPLAY CODE) OR NULL. THE FIRST CALL TO ASCII INITIALIZES AS A GLOBAL VARIABLE AN ARRAY CALLED ASCII, WHOSE DIMENSIONS ARE: '0:7,0:7', AND LOADS THIS ARRAY WITH THE ALPHABET FOR OCTAL BASE INDEXING.

IF A CALL TO ASCII HAS A NULL ARGUMENT, A SET OF VARIABLE NAMES ARE SET UP MAKING AVAILABLE SOME OF THE ASCII CHARACTERS AND CONTROL CHARACTERS. THESE VARIABLES ARE GLOBAL AND ARE FOR THE CONVENIENCE OF THE USER. THEY ARE: BELL IS THE BELL (OR BLEEP); LF IS A LINE FEED; CR IS A CARRIAGE RETURN; EXCL IS AN '!' EKARR IS A BACK ARROW; CONTROL CHARACTERS ARE GIVEN THESE NAMES: CTLJ, CTLX, CTLZ, CTLN, CTLØ, CTLU; AND EJECT IS GIVEN AS ITS VALUE SIX CONSECUTIVE LINE FEEDS.

ASCII FAILS IF ITS ARGUMENT IS NOT A TWO OR FOUR OCTAL DIGIT NUMBER OR NULL.
SNØLIB NOTES:

1. THE FOLLOWING FUNCTIONS MAKE REFERENCE TO OTHER FUNCTIONS IN SNØLIB:
   - GET - NEEDS ANCHR
   - RANDOM - NEEDS INTEGER
   - DUMP - NEEDS LFN
   - STATUS - NEEDS LFN
   - MODIFY - NEEDS DUPL, ANCHR
   - TRACE - NEEDS LFN
   AND MAKE CALLS TO GET TO COMPLETE THE DEFINITION OF THE NEEDED FUNCTIONS. IF A FUNCTION IS ALREADY DEFINED AND CODED, GET DOES NOT DO SO AGAIN.

2. CLEARO IS THE ONLY FUNCTION WITH NO ARGUMENTS. THE FOLLOWING FUNCTIONS WILL OPERATE ON THEIR DEFAULTS UPON A NULL CALL:
   - GET() - DEFINES THE STANDARD FUNCTIONS.
   - EVAL() - THE NULL STRING CAN BE CODED.
   - INTEGER() - THE NULL STRING IS EQUIVALENT TO ZERO.
   - RANDOM() - EQUIVALENT TO RANDOM(0:9), 1, 'N'.
   - DUMP() - EQUIVALENT TO DUMP('OUTPUT').
   - CLEAR() - HAS NO ARGUMENTS.
   - SORT0() - RETURNS THE NEXT MEMBER OF A SORTED SET.
   - ASCII() - SETS UP A SET OF ASCII CHARACTERS.

3. INTERNAL TO THE FUNCTIONS, CERTAIN CONVENTIONS ARE OBSERVED.
   A) ALL FUNCTION NAMES ARE ENTRY POINTS. DUPLICATE LABELS ARE NOT TAKEN CARE OF PROPERLY BY CAL SNØBOL (AS DESCRIBED IN GRISWOLD ET AL.) AND ARE AN ERROR.
   B) ALL GLOBAL VARIABLES AND OTHER LABELS USED IN SNØLIB BEGIN WITH THREE ALPHABETIC CHARACTERS, THEN A DOUBLE PERIOD, THEN AN INTEGER (EX: 'DUM..2'), TO AVOID CONFLICT WITH USER VARIABLES. IN ADDITION, INDIRECT REFERENCING (USING ASCII CHARACTERS) OF THE FORM $(VAR '1') AND THE LIKE IS USED IN SOME OF THE FUNCTIONS (EVAL, TRACE, GET, SORTI, SÖRT0).

4. IF YOUR PROGRAM REACHES MAXIMUM FIELD LENGTH (WHICH IS SET TO 33000 UNDER THE RUN OR RNH COMMAND IN TIMESHARING), SEE THE SECTIONS BELOW ON TIMESHARING CONVENTIONS AND ON RUNNING SNØBOL UNDER THE BATCH SUBSYSTEM FOR HELP ON EXTENDING YOUR FIELD LENGTH. SEE ALSO 'KNOWN BUGS'.

5. FAILURE (FRETURN) DUE TO A VARIABLE NAME WHICH IS NOT A LEGAL FILE NAME CAN RESULT FROM THE CALL TO ANY FUNCTION THAT USES A FILE (TRACE, DUMP, STATUS).

6. TO LOOK AT A FILE THAT A FUNCTION CREATES, REWIND THE FILE AND COPY IT TO OUTPUT. THUS:
   REWIND, FILENAM
   LNH, F=FILENAM

PAGE 16
RUNNING SNOBOL UNDER THE BATCH SUBSYSTEM:

SNOBOL PROGRAMS MAY BE RUN UNDER THE BATCH SUBSYSTEM. THIS IS USEFUL IF A FIELD LENGTH > 33000 IS NEEDED FOR A PROGRAM. THE COMMANDS BELOW ACCESS SNOBOL IN THE BATCH SUBSYSTEM:

BATCH,40000.  (FOR A FIELD LENGTH OF 40000)
X, SNOBOL, PARAMETERS

FOR EXAMPLE:  BATCH,40000.
X, SNOBOL, I=SN0FLAK, L=LIST, N, P=40000.

LEGAL PARAMETERS ARE DESCRIBED BELOW:
I - SPECIFIES THE FILE ON WHICH THE SOURCE PROGRAM RESIDES; DEFAULT IS I=INPUT.
L - SPECIFIES ONTO WHAT FILE THE LISTING OF THE SOURCE PROGRAM IS TO GO; DEFAULT IS L=OUTPUT. L=0 SUPPRESSES THE LISTING.
O - SAME AS THE L PARAMETER.
N - INDICATES THAT THE SOURCE PROGRAM HAS LINE NUMBERS.
B - SELECTS THE BUFFER SIZE TO BE USED FOR ALL I/O FILES DURING EXECUTION. DEFAULT IS B=401 (OCTAL). MINIMUM IS B=101 (OCTAL).
F - SETS THE UPPER LIMIT ON FIELD LENGTH EXPANSION THAT SNOBOL MAY GROW TO DURING EXECUTION. DEFAULT IS F=33000 (OCTAL). THIS PARAMETER MUST BE USED TO GET A FIELD LENGTH IN EXCESS OF 33000.
X - SPECIFIES THE ALTERNATE CHARACTER SET BASED ON CDC 6000 LINE PRINTER CHARACTER EQUIVALENTS. THE TWO CHARACTER SETS AVAILABLE ARE GIVEN IN A TABLE ON PAGE 2 OF SN0INFO. SEE ALSO THE DISCUSSION OF CHARACTER SETS IN THE 'TIMESHARING CONVENTIONS' SECTION, BELOW.
H - SETTING H=1 'ALLOWS HEAP BLOCK COMPACTI0N. DEFAULT IS H=0. SEE 'KNOWN BUGS'.

WHEN EXECUTED, THE SNOBOL PROGRAM WILL BE LISTED (UNLESS L=0 IS SPECIFIED), AND LISTING DIRECTIVES INSERTED INTO THE SOURCE BECOME USEFUL. OF THE LISTING DIRECTIVES AVAILABLE IN CAL SNOBOL (-UNLIST, -LIST, -EJECT, -SPACE) ONLY -UNLIST AND -LIST AFFECT THE REMOTE TERMINAL PRINTER.

-UNLIST TURNS THE LISTING OFF AT THE POINT IT APPEARS.
-LIST TURNS ON THE LISTING AFTER THE LINE ON WHICH IT APPEARS.

INPUT LINES ARE LIMITED TO 72 COLUMNS UNDER BATCH SNOBOL ( N PARAMETER NOT SELECTED). SNOBOL IGNORES COLUMNS 73-80, IF PRESENT, MOVING THEM TEN SPACES OVER TO THE RIGHT ON THE LISTING, AND COMPLETELY MISSES ANY FURTHER CHARACTERS PAST COLUMN 80.
TIMESHARING CONVENTIONS:
---------------

SNOBOL IS AVAILABLE AS A SUBSYSTEM ON MERITSS BY TYPING:

SNOBOL,NNNNN

WHERE 'NNNNN' IS ONE OR MORE ADDITIONAL COMMANDS. IF 'NNNNN' IS NULL AND NO PRIMARY FILE IS CURRENTLY SPECIFIED, THE SYSTEM RESPONDS:

OLD, NEW, OR LIB FILE:

THE USER MUST THEN SPECIFY WHICH OF THE THREE FILE TYPES HE WISHES TO USE. THE ADDITIONAL COMMANDS CAN BE ANY LOGICAL SEQUENCE OF COMMANDS THAT WOULD NORMALLY FOLLOW THE ABOVE COMMAND. EXAMPLES:

SNOBOL,OLD,A - BRINGS UP THE SNOBOL SUBSYSTEM AND THE PERMANENT FILE A.
SNOBOL,RUN,I=B - BRINGS UP THE SNOBOL SUBSYSTEM AND EXECUTES THE SECONDARY LOCAL FILE B.
SNOBOL,STATUS - BRINGS UP THE SNOBOL SUBSYSTEM AND EXECUTES THE STATUS COMMAND.

THE IBM CHARACTER SET FOR SNOBOL4 IS THE DEFAULT SET ON MERITSS; THE ONLY DIFFERENCE IS THE CHARACTER FOR ALTERNATION IN PATTERNS; THIS IS AN UP ARROW : . THE OTHER CHARACTER SET CORRESPONDS TO THE CDC 6000 LINE PRINTER CHARACTER EQUIVALENTS ON REMOTE TERMINALS, AND IS OBTAINED BY SPECIFYING AN 'X' ON THE RUN OR RNH COMMAND IN THE SNOBOL SUBSYSTEM, OR ON THE SNOBOL CONTROL CARD IN THE BATCH SUBSYSTEM. SEE THE DISCUSSION PRECEEDING THE CHARACTER SET TABLE.

IN TIMESHARING SNOBOL, OR IF THE 'N' PARAMETER IS SPECIFIED ON THE BATCH CONTROL CARD, EXACTLY ONE SPACE MUST FOLLOW THE LINE NUMBER IF THE STATEMENT HAS A LABEL (THIS SPACE IS SUPPLIED AUTOMATICALLY IN 'AUTO' MODE). AT LEAST TWO SPACES MUST FOLLOW THE LINE NUMBER IF THE SNOBOL STATEMENT DOES NOT CONTAIN A LABEL.

WHEN EXECUTED, THE LINE NUMBERS AND ONE LEADING CHARACTER, (USUALLY A BLANK, AS FROM "AUTO" MODE) ARE REMOVED, AND THE REST OF THE LINE IS CONSIDERED THE SNOBOL STATEMENT. THE MAXIMUM LINE LENGTH FROM A TIMESHARING TERMINAL IS 150 CHARACTERS, WHILE THE SNOBOL INTERPRETER DOES NOT LIMIT INPUT LENGTH.
THE USER MAY START EXECUTION OF A SNØBØL PROGRAM BY ENTERING THE 'RUN' OR 'RNH' COMMAND. LEGAL PARAMETERS FOR THE RUN COMMAND UNDER THE SNØBØL SUBSYSTEM ARE:

I=FILE - WHERE FILE IS A SECONDARY FILE TO BE EXECUTED. THE PRIMARY FILE NEED NOT BE SPECIFIED IF IT IS THE ONE TO BE EXECUTED.
MA=NNNNN - OR MI=NNNNN, TO REQUEST ADDITIONAL FIELD LENGTH. MA - USE NNNNN AS THE FIELD LENGTH FOR THE JOB (NNNNN IS IN THE DEFAULT BASE; OCTAL, UNLESS POST RADIX IS GIVEN); MI - USE NNNNN AS AN INCREMENT TO THE SYSTEM CALCULATED FL; OTHERWISE IT IS THE SAME AS MA.
X - SPECIFIES THE ALTERNATE CHARACTER SET, BASED ON CDC 6000 LINE PRINTER EQUIVALENCES. THE TWO CHARACTER SETS AVAILABLE ARE GIVEN IN A TABLE ON PAGE TWO OF SNØINFØ.

EXPLANATIONS OF COMPILE TIME ERRORS INCLUDE THE LINE NUMBER OF THE OFFENDING STATEMENT, AND A BRIEF DIAGNOSTIC OF ONE OR MORE ERRORS ON THAT LINE.

EXPLANATIONS OF EXECUTION TIME ERRORS APPEAR AS FOLLOWS:

ERROR TERMINATION
IN STATEMENT N
AT LEVEL M

< EXECUTION ERROR MESSAGE >

STATEMENT N IS NOT THE LINE NUMBER, BUT THE SNØBØL STATEMENT AT WHICH THE ERROR WAS DETECTED (COUNT STATEMENTS BEGINNING WITH THE FIRST STATEMENT, NOT INCLUDING COMMENTS AND CONTINUATIONS.) LEVEL M, OF COURSE, IS THE FUNCTION LEVEL AT WHICH THE ERROR OCCURRED. THE MAIN PROGRAM IS LEVEL 0. THE EXECUTION ERROR MESSAGE DESCRIBES THE TYPE OF FATAL ERROR.

SNØINFØ VERSION 2.1 74/05/03. JOHN T. EASTØN

LOCATED ON THE MERITSS PROGRAM LIBRARY. OFFSET COPIES ARE AVAILABLE IN ROOM 140 EXPERIMENTAL ENGINEERING, OR FROM THE UCC LIBRARIAN.

TO LIST FROM THE MERITSS LIBRARY, DO:
LIB,SNØINFØ
LIST

PAGE 19
INDEX TO SNOLINFO

7 - ALPHABET()
5 - ALT(FIRST,REST)
7 - ANCHOR(X)
11 - ANCHOR() [SNOLIB]
3 - APPLY(f,X1,X2,...,Xn)
2 - ARITHMETIC
3 - ARRAY(STRING)
15 - ASCII(\OCTAL\NUM) [SNOLIB]
7 - ASCII CHARACTERS
17 - BATCH CONTROL CARD PARAMETERS
17 - BATCH LINE LENGTH LIMIT
3 - BINARY OPERATORS
8 - BUGS, KNOWN
5 - CAT(FIRST,REST)
18 - CHARACTER SETS ON MERITSS
2 - CHARACTER SETS ON MERITSS, TABLE OF
14 - CLEAR() [SNOLIB]
3 - CLOCK()
3 - CONVERT(X)
2 - DATATYPES
3 - DEFERRED EVALUATION
8 - DEFERRED EVALUATION
6 - DETACH(VNAME)
5 - DOL(LEFT,RIGHT)
14 - DUMP(MESS,FILE,TERM) [SNOLIB]
11 - DPLIC(STR,N) [SNOLIB]
6 - ENDGROUP(FNAME,1)
6 - ENDLEVEL(FNAME)
6 - EOL(FNAME)
19 - ERROR MESSAGES
11 - EVAL(STR) [SNOLIB]
9 - EXAMPLE PROGRAM TO ACCESS SNOLIB
1 - EXAMPLE SNOBOL PROGRAM
8 - EXPONENTIATION OF INTEGERS
2 - EXPONENTIATION OF INTEGERS
5 - FAMILY(X)
17 - FIELD LENGTH
5 - FIRST(PATTERN)
7 - FNCLEVEL()
3 - FREEZE(STRING)
16 - FUNCTIONS WHICH MAKE A CALL TO GET
6 - FUNCTIONS, INPUT/OUTPUT
3 - FUNCTIONS, PRIMITIVE
9 - FUNCTIONS, SIMULATED [SNOLIB]
8 - GARBAGE COLLECTOR
10 - GET(STRING) [SNOLIB]
8 - HEAP BLOCK COMPACTOR BUGS
4 - IF(E1,E1,...,EN)
6 - INPUT/OUTPUT FUNCTIONS
6 - INPUT(VNAME,FNAME,LENGTH)
6 - INPUT AND OUTPUT ASSOCIATIONS
2 - INTEGERS
11 - INTEGER(STR) [SNOLIB]
INDEX TO SN0INFO
74/05/02. 18.19.45.

2 - INTRODUCTION
4 - ITEM(A,S1,S2,...,SN)
7 - KEYWORDS
5 - LEFT(NAMING,PATTERN)
12 - LFN(TESTFN) [SN0LIB]
17 - LISTING DIRECTIVES
10 - LIST OF FUNCTIONS IN SN0LIB
7 - MAXLENGTH(I)
2 - MIXED MODE ARITHMETIC
14 - MODIFY(MESS,FIX) [SN0LIB]
15 - MODIFY DIRECTIVES [SN0LIB]
4 - NEXTVAR(X)
3 - OPERATORS
6 - OUTPUT(VNAME,FNAME,PREFIX)
5 - PARAM(PRIM,FNC)
5 - PRD(LEFT,RIGHT)
3 - PRIMITIVE FUNCTIONS
3 - PRIMITIVE FUNCTIONS NOT IMPLEMENTED
4 - PROTOTYPE(X)
4 - PSEUDOFIELD FUNCTIONS
13 - RANDOM(RANGE,COUNT,DELIM,SEED) [SN0LIB]
2 - REAL NUMBERS
5 - REMARK STRING)
11 - REMDR(X,Y) [SN0LIB]
11 - REPLACE(STR1,STR2,STR3) [SN0LIB]
5 - REST(PATTERN)
11 - REVERSE(STR) [SN0LIB]
6 - REWIND(FNAME)
5 - RIGHT(NAMING,PATTERN)
17 - RUNNING SN0B0L UNDER THE BATCH SUBSYSTEM
19 - RUN (RNH) COMMAND PARAMETERS
8 - SENSING END OF FILE
17 - SN0B0L CONTROL CARD PARAMETERS
9 - SN0B0L SIMULATED FUNCTIONS
18 - SN0B0L SUBSYSTEM
10 - SN0LIB FUNCTIONS AND THEIR PARAMETERS
16 - SN0LIB NOTES
12 - SORT(STRING,DELIMITER,PROTECT) [SN0LIB]
12 - SORT(DELETE) [SN0LIB]
5 - STAR(RIGHT)
12 - STATUS(INFO,FILE) [SN0LIB]
7 - STCOUNT(I)
7 - STLIMIT(I)
12 - STOPTR(VAR) [SN0LIB]
2 - SUCCEED
12 - SUCCEED [SN0LIB]
11 - TABLE(VAR,CHAR) [SN0LIB]
18 - TIMESHARING CONVENTIONS
12 - TRACE(VAR,TAG,FILE) [SN0LIB]
5 - TYPE(X)
3 - UNARY OPERATORS
8 - UNEVALUATED EXPRESSIONS
3 - UNEVALUATED EXPRESSIONS