# SNOBOL4 Information Bulletin

Madge T. Griswold and Ralph E. Griswold

Status of the SNOBOL4 Information Bulletin

The SNOBOL4 Information Bulletin has been published aperiodically since 1968. For the most part, is has contained information about new and planned implementations of SNOBOL4, notices of new documents related to SNOBOL4, and requests for information from readers. In recent years, the frequency with which the Bulletin has been published has dropped off, mostly due to lack of material.

With the appearance of implementations of SNOBOL4 for personal computers and a number of articles on SNOBOL4 in computer magazines, there has been an increase in interest in SNOBOL4. The subscription list for this bulletin now contains the names of 850 persons, many of which have been added recently. For those of you who are receiving the Bulletin for the first time, welcome!

Since there is renewed interest in SNOBOL4, we are adding two new features with this issue: ''From Our Mail'' and ''Programming Corner''. From Our Mail features information of general interest collected from our mail from users and implementors. Programming techniques in SNOBOL4 are featured in Programming Corner. This material generally is on the sophisticated side, and it is not intended as a tutorial for persons just learning SNOBOL4 — there are a number of books for that purpose.

We encourage our readers to comment on the Bulletin, especially on new features, and to contribute material for future issues. Implementations of SNOBOL4

There is not much new in the way of SNOBOL4 implementations. Current flyers for two personal computer implementations are attached to this Bulletin. A complete list of known implementations is available, free of charge, from:

> SNOBOL4 Project
> The University of Arizona
> Tucson, Arizona   85721

Ask for S4D57.  Upcoming SNOBOL4 Conference

The second International Conference on the Applications of SNOBOL and SPITBOL (ICEBOL 86) will be held in Madison, South Dakota on October 9-11, 1986. In addition to papers, there will be programming clinics and the opportunity to meet with both users and implementors of SNOBOL4.

Subscribers to this Bulletin should have received information about the conference already. If you did not receive this information, you can obtain it by writing to:

> Professor Lynn Ryan
> 111 Beadle Hall
> Dakota State College
> Madison, South Dakota   57042

From Our Mail

*Is there an implementation of SNOBOL4 for the Macintosh?*

No. Several persons have talked about doing such an implementation, but nothing has appeared yet. The same is true of other computers that use the MC 68000 processor.

*Is there an implementation of SNOBOL4 for the Data General MV 8000?*

No, here also. Lots of folks would like one.

*Where do I get SNOBOL4 for my IBM mainframe computer?*

The University of Arizona distributes SNOBOL4 for computers with IBM 370 architectures. For ordering information, ask for S4D57, mentioned previously.

*Can I get source for the original Macro (SIL) implementation of SNOBOL4 on a diskette?*

Yes, it is available on a 2S/DD 5-1/4" diskette in MS-DOS format.  See S4D57 for ordering information.

*Can I get information about SNOBOL4 via electronic mail?*

Yes. The electronic mail addresses for SNOBOL4 are:

```
        snobol4@arizona.edu    (CSNET)
```

*Can I get source for the Macro implementation of SNOBOL4 electronically?*

It is too large to send as mail, but if you have access to ARPA Internet, you can get it via an anonymous FTP to `arizona.edu`.  To see what's available, get the file `snobol4/README`.  Programming Corner

For the first programming corner, we have contributions from two of the implementors of SNOBOL4 for personal computers: Viktors Berstis and Mark Emmer.

**The N-Queens Problem**

One of the programs contained in Berstis' Minnesota SNOBOL4 distribution provides solutions to the *n*-queens problem: the ways in which *n* queens can be placed on an *n*-by-*n* chess board in such a way that none can attack another.  This is a famous problem for which many techniques for solution have been used.  Berstis' program uses backtracking in pattern matching. The format of his program has been changed slightly for listing here:

```
    * N queens problem, a string oriented version to demonstrate the power
    *  of pattern matching.

    * A numerically oriented version will run faster than this.

        OUTPUT = "Please supply the number of queens"
        N = INPUT    :F(END)

        NM1 = N - 1
        NP1 = N + 1
        NSZ = N * NP1

        &STLIMIT = 10 ** 9
        &ANCHOR = 1

        DEFINE("SOLVE(B)I")

    * This pattern tests if the first queen attacks any of the others:
        TEST = BREAK("Q") "Q" (ARBNO(LEN(N) "-") LEN(N) "Q"
    +           | ARBNO(LEN(NP1) "-") LEN(NP1) "Q"
    +           | ARBNO(LEN(NM1) "-") LEN(NM1) "Q")
        P = LEN(NM1) . X LEN(1)
        L = "Q" DUPL("-",NM1) " "

        SOLVE()  :(END)

    SOLVE    EQ(SIZE(B),NSZ)  :S(PRINT)
```

```
* Add another row with a queen:
    B = L B
LOOP I = LT(I,N) I + 1    :F(RETURN)
    B TEST   :S(NEXT)
    SOLVE(B)
* Try queen in next square:
NEXT B P = "-" X  :(LOOP)
PRINT   SOLUTION = SOLUTION + 1
    OUTPUT = "Solution number " SOLUTION " is:"
PRTLOOP B LEN(NP1) . OUTPUT =    :S(PRTLOOP)F(RETURN)
    END
```

Berstis suggests that readers try to solve the problem with substantially less code (except for cosmetics) or more efficiently.

## Sorting

Mark Emmer, who implemented Catspaw SNOBOL4+, contributes the following interesting suggestions on sorting.

The REPLACE function can be used to alter the collating sequence of SORT and LGT. Suppose you want each lowercase letter to collate immediately before its uppercase sibling. A ''map from'' string specifies the desired order. It should include all letters that can appear in the data:

```
MF = "aAbBcCdD .. zZ"
```

Create a ''map to'' string consisting of the first N characters of &ALPHABET, where N is the length of the ''map from'' string:

```
&ALPHABET LEN(SIZE(MF)) . MT
```

Finally, to compare strings X and Y under the new collation, use:

```
LGT(REPLACE(X,MF,MT), REPLACE(Y,MF,MT))
```

If you're building a table, with some string X as the subscript, create each entry using the ''replaced'' value of the letters in X:

```
T[REPLACE(X,MF,MT)] = ..
```

Then sorting T produces an array with entries sorted by the new collating sequence:

```
A = SORT(T)
```

When accessing the elements of A, the first column would have to be ''decollated'' by running the REPLACE function in reverse, for example:

```
OUTPUT = REPLACE(A[1,I],MT,MF) "--" A[2,I]
```

Bug Fix for SNOBOL4+

Catspaw reports that Daryl Gibb of the University of Tampere, Finland, has found a bug in SNOBOL4+. The comparison and sort functions (LLT, LGT, SORT, etc.) are comparing characters with ASCII values above 128 as signed quantities. This means that the extended IBM characters (values 128 to 255) are collating ahead of the regular ASCII character set.

The problem can be corrected by running a SNOBOL4 patching program against the SNOBOL4.EXE file. This patch applies to Version 1.7 (Prentice-Hall) of SNOBOL4+ only. Users with earlier versions should contact Catspaw for the patch addresses.

```
*      Program PATCH, for V1.7 SNOBOL4+ only.
*      Note that "v", "r", and "w" are lowercase letters.
*
       OUTPUT("OUT", 1, "RWB")  :F(ERROR)
       SEEK(1, 32906.0, 0)
       OUT = "v"
       SEEK(1, 35035.0, 0)
       OUT = "r"
       SEEK(1, 35052.0, 0)
       OUT = "w"
       SEEK(1, 35086.0, 0)
       OUT = "r"
       SEEK(1, 35069.0, 0)
       OUT = "w"
       SEEK(1, 40717.0, 0)
       OUT = "r"
END
```

After entering the patch program, place a working copy of SNOBOL4+ on your default drive and enter the following DOS command line:

```
C>SNOBOL4 PATCH /1:SNOBOL4.EXE
```

This loads and executes the PATCH program, which in turn corrects several instructions in the SNOBOL4.EXE file. Use this new version of the SNOBOL4.EXE file for your daily work.

A quick confidence check of the patch can be made by entering the following short program. It should print OK, and the dump output should place the variable U (with an umlaut) after the variable UM.

```
       &DUMP = 1
       &ALPHABET TAB(154) LEN(1) . UM
       OUTPUT = LGT(UM, "U") "OK"
       $UM = 123
END
```

New Books

There are several new books related to SNOBOL4. Mark Emmer provides the following information:

Three books on SNOBOL4 programming for the humanities have been published recently in Great Britain. None of the books assumes any computer experience on the part of the reader. All contain student exercises and solutions. They are:

1.  Susan Hockey. *SNOBOL Programming for the Humanities*. Oxford: Oxford University Press, 1985. 178 pages. ISBN 0-19-824676-5.

    The author's title is Manager, Computing in the Arts, Oxford University Computing Service. This book is an introduction to computer programming for non-scientific applications. It was developed from a programming course given by the author at Oxford University.

2.  Christopher Butler. *Computers in Linguistics*. Oxford: Basil Blackwell, Inc. 1985. 266 pages. ISBN 0-631-14267-3.

    The author is Lecturer in Linguistics at the University of Nottingham. The first third of the book is devoted to an overview of the packaged programs available for computing in the arts. The remainder of the book teaches the SNOBOL4 language, concluding with two large programs dealing with poetry analysis and lexical density.

3.  A. Colin Day. *Text Processing*. Cambridge: Cambridge University Press, 1984. 141 pages. ISBN 0-521-28683-2.

> The author is with the Computer Center, University College, London.  The book begins with a discussion of computer internals as they relate to string processing. Fortran, Pascal, and SNOBOL4 are compared for simple problems, and an introduction to SNOBOL4 is presented.

> The first two books are available from Catspaw, Inc. for $15.00 each, plus $2.00 shipping (domestic).

There is another book that may be of interest to some of our readers:

> G. Sampath. *An Implementation to Text Processing*. 1985. 258 pages.

While this book is not related directly to SNOBOL4, it contains a variety of information about the representation and manipulation of textual data.  Copies are available from:

> River Valley Publishing
> P.O. Box 99752
> Jeffersontown, Kentucky   40299

The price is $20 plus $2 for UPS delivery (continental United States).  Enquire about deliveries outside the United States before ordering.

SNOBOL4 (StriNg Oriented and symBOlic Language) is a language for text processing, pattern matching, and much more, first designed and implemented at Bell Telephone Laboratories, Inc. (BTL) in the 1970's. SNOBOL4, while known primarily as a string language excels at any task involving symbolic manipulations. It provides run time typing, garbage collection, user data types, on the fly compilation. The language has a simple and highly orthogonal syntax, making it easy to learn. A later implementation, developed by Robert B. K. Dewar and called "SPITBOL", added several major enhanc snobol4.com at WI. All about SNOBOL4 and SPITBOL, compilers and interpreter for a text-processing, pattern-matching programming language. Â Stats & Details Whois IP Whois Expand all blocks. SNOBOL4 and SPITBOL Information. All about SNOBOL4 and SPITBOL, compilers and interpreter for a text-processing, pattern-matching programming language. Keywords: programming language, Snocone, snobol4, snobol, SPITBOL. Nov 28, 2019. Created