

SHARE PROGRAM LIBRARY AGENCY

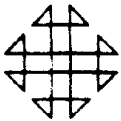


PROGRAM NUMBER

066004

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257



CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM
(for IBM S/360, 1130 and 1800)

IBM Corporation
Program Information Department (PID)
40 Saw Mill River Road
Hawthorne, New York 10532, U.S.A.
Attention: Program Control Desk

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

①	Program Order Number (to be filled in by PID)	360D-06.6.004
②	System Type (machine)	S / 360
③	Search Key	CHARACTER FILTER PL1
④	Programming Language	360 OS ALF
⑤	Author's Name and Address	Harold P. Sieglaff 3610 W Northview Phoenix Arizona 85021
⑥	Direct Inquiries to Name and Address (if different than Author)	
⑦	Title of Program	CHARACTER FILTER PL1
⑧	Submitter's User Group Affiliation Code and Installation Code	N
⑨	Submitter's Own Program Identification and Suffix (optional)	PSAS
⑩	Primary Subject Code	0.6.6
⑪	Secondary Subject Codes	
⑫	Operating or Monitor System Required	
⑬	New or Revision Code (if revision, show prior Program Order Number in item 1)	N
⑭	Year Completed	68
⑮	Date of Submittal	12 20 68
⑯	Documentation (number of original pages submitted)	8
⑰	Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.	

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements
- g. Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT	The subroutine provides a means to skip or seek specified
	characters while scanning a string of characters for a PL1 program.
	The subroutine is written in OS Assembly Language F and was
	tested using OS PL1 Language F Level and OS Version 13 on a S/360
	model 50.
	The routine can be used to filter in/out alphabetic,
	numeric, alphanumeric, or other characters while scanning a
	character string.
(Please attach additional pages if necessary) Total pages attached _____	

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

(18) Signature of Submitter and Date Harold P. Sieglaff 20 DEC 68

(19) Signature of Installation Addressee _____

T4SF

DECK KEY

Deck Source deck of SETSKP, sequence 00 thru 1080 in cc 73-80;
132 cards.

Table of Contents Page

Card deck key 4

Write-up of the subroutine SKIP / SEEK 1-4 5

DISCLAIMER
Triangle Universities Computation Center (TUCC)
serves solely as the distribution agent for contributed
programs and does not test or maintain them. They
are distributed "as is" without warranty. They
are distributed by the author. Neither TUCC nor SHARE, INC.,
makes any warranty, expressed or implied, as to the
documentation, function, or performance of the con-
tributed programs.

NAME SETSKP

ENTRY

POINTS: SETSKP supplies a string of characters which are to be ignored (skipped) while scanning a string supplied by the CALL SKIP statement

SKIP supplies a string of characters which is to be scanned while skipping the characters supplied by the CALL SETSKP statement

SETSEK supplies a string of characters which are to be searched for while scanning a string supplied by the CALL SEEK statement

SEEK supplies a string of characters which is to be scanned in a search for the characters supplied by the CALL SETSEK statement

DESCRIPTION: SETSKP provides a means to skip or seek specified characters while scanning a string of characters

AUTHOR: Harold P. Sieglaff

LANGUAGE: BAL

MEMORY REQUIREMENTS: 916 bytes (2 tables of 256 bytes each are used)

AVAILABILITY: SETSKP is available as a source deck from the author.

CALLING SEQUENCES:

CALL SETSKP (IGNORE, NBYTES)

CALL SKIP (STRING, NSTART, NCHARS, NUMPOS)

CALL SETSEK (STOPIF, NBYTES)

CALL SEEK (STRING, KBEGIN, NCHARS, NUMPOS)

PARAMETER DESCRIPTIONS:

IGNORE = a string of characters which are to be ignored (skipped) when scanning a string supplied by the CALL SKIP statement

NBYTES = no. of bytes (characters) to be used from string IGNORE

STRING = a string of characters which is to be scanned while skipping the NBYTES characters of the string IGNORE

NSTART = no. of the position of STRING where the scan is to start

NCHARS = no. of bytes (characters) in STRING which are to be scanned

upon return to the calling program:

NUMPOS = no. of the first position in STRING not equal to any character in string IGNORE

If NUMPOS = 0 then all of the characters of STRING are equal to some character in IGNORE.

NUMPOS = 1 for NSTART
2 for NSTART + 1
3 for NSTART + 2, etc.

CALL SKIP may be executed any number of times after CALL SETSKP has been executed.

CALL SEEK may be executed any number of times after CALL SETSEK has been executed.

STOPIF = a string of characters which are to be searched for while scanning a string supplied by the CALL SEEK statement.

NBYTES = no. of bytes (characters) to be used from string STOPIF

STRING = a string of characters which is to be scanned in a search for the NBYTES characters of string STOPIF

KBEGIN = no. of the position of STRING where the scan is to start

NCHARS = no. of bytes (characters) in STRING which are to be searched upon return to the calling program:

NUMPOS = no. of the first position in STRING equal to any of the characters in string STOPIF.

If NUMPOS = 0 then no character in STRING is equal to any of the characters in string STOPIF.

NUMPOS = 1 for KBEGIN
2 for KBEGIN + 1
3 for KBEGIN + 2, etc.

RESTRICTIONS:

NCHARS must be less than or equal to 256

NBYTES must be less than or equal to 255

ERROR

none

EXAMPLESOF ITS USE:

```
CALL SETSKIP (IGNORE, NBYTES)
CALL SKIP (STRING, NSTART, NCHARS, NUMPOS)
CALL SKIP (STRING, NSTART, NCHARS, NUMPOS)
CALL SETSEK (STOPIF, NBYTES)
CALL SEEK (ALPHNUM, KBEGIN, KSYMS, NUMCHR)
CALL SEEK (ALPHNUM, KBEGIN, KSYMS, NUMCHR)
```

NOTE: STRING, IGNORE, STOPIF are fixed length character strings

```
SKPSEEK: PROCEDURE OPTIONS (MAIN);
  DCL STRING CHAR(80);
  DCL IGNORE CHAR(80);
  DCL STOPIF CHAR(80);
  DCL LDUMMY CHAR(70);
  /* READ NO. OF STARTING POSITIONS FOR SKIP & SEEK */
  GET EDIT ( NISTSKP, NISTSEK, LDUMMY) (2 F(5), A(70));
  LSIZSKP = 81 - NISTSKP;
  LSIZSEK = 81 - NISTSEK;
  /* READ THE 3 STRINGS ; PREPARE TO SKIP & SEEK 40 CHARACTERS */
  GET EDIT (STRING, IGNORE, STOPIF) (3 A(80));
  N2SKIP = 40;
  N2SEK = 40;
  /* INITIALIZE THE CHARACTERS TO BE SKIPPED */
  CALL SETSKIP (IGNORE, N2SKIP);
  /* SCAN STRING SKIPPING CHARACTERS IN IGNORE */
  CALL SKIP (STRING, NISTSKP, LSIZSKP, NSKIP);
  PUT PAGE DATA (NSKIP);
  /* INITIALIZE THE CHARACTERS TO BE SEARCHED FOR */
  CALL SETSEK (STOPIF, N2SEK);
  /* SCAN STRING SEARCHING FOR THE CHARACTERS IN STOPIF */
  CALL SEEK (STRING, NISTSEK, LSIZSEK, NSEK);
  PUT SKIP DATA (NSEK);
  END SKPSEEK;
```

PLI F Programmer's Guide, File No. S360-29, Form C28-6594-3, pp. 112, 135.

(see the index under Default attributes, Dope Vectors)

(under Default attributes in the text see REAL FIXED BINARY)

(under Dope Vectors in the text see String Dope Vector (SDV))