ANL-PHY-1972G

Argonne Physics Division
Informal Report PHY-1972G
July 1972

MASTER

THE SPEAKEASY HELP DOCUMENTS

by

J. K. Fink

Argonne National Laboratory, Argonne, Illinois 60439

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

Argonne Physics Division
Informal Report PHY-1972G
July 1972

THE SPEAKEASY HELP DOCUMENTS*

by

J. K. Fink

Argonne National Laboratory, Argonne, Illinois 60439

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

A limited number of copies are available for general distribution. Anyone who desires a copy should contact S. Cohen of the Argonne Physics Division; the PHY Informal Reports are not handled through the ANL Technical Publications Department.

*Work performed under the auspices of the U. S. Atomic Energy Commission.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	SPEAKEASY WORDS DEFINED IN THE HELP DOCUMENTS	3
III.	ONE-LINE DEFINITIONS OF THE SPEAKEASY WORDS	5
IV.	THE SPEAKEASY HELP DOCUMENTS	11
	APPENDIX	47

ABSTRACT

The HELP documents provide SPEAKEASY users with concise definitions of most of the words available in the current processors. In this report, the documents are given in a variety of formats to enable one to find specific information quickly. The bulk of this report consists of computer read out of the HELP library via SPEAKEASY.

I. INTRODUCTION

The user of an interactive language such as TSO SPEAKEASY needs a means of quickly determining the operations available to him and of easily learning to use particular features. To fill this need, a special library named HELP has been created and attached to all SPEAKEASY processors. The HELP library consists of concise documents that describe each word of the SPEAKEASY language. The interactive user may obtain each document by a simple command from his terminal.

The growth capability of SPEAKEASY has created difficulties in formally documenting all new features as rapidly as they are added. Since the HELP documents are available to all SPEAKEASY processors and thus readily accessible to all users, they have become the primary method of documentation of new features.

This report is a preliminary attempt to supply the information available to these documents to the users of SPEAKEASY in an easily read form. The HELP documents have been obtained via SPEAKEASY programs in three formats. The SPEAKEASY program that generates each of the three sections precedes it.

Section II consists of an index to the words described in the HELP documents. Each entry in the HELP library is given a reference name consisting of eight characters. The reference name is usually identical to the name used in the SPEAKEASY processor. If the SPEAKEASY word being described is more than eight characters in length, the reference name consists of the first eight characters.

Short one-line definitions of the words in the HELP library are given in Section III. These one-line definitions are the first line of each HELP document. Each line contains the word being defined with arguments of its SPEAKEASY call followed by a brief definition.

The last section consists of the HELP documents themselves. Each document begins with the word being described and gives a concise definition of the word, including all possible SPEAKEASY calls. The word being defined and its arguments are capitalized throughout the document so that they may be easily located. Most documents are structured in levels; each paragraph gives greater detail than its predecessor so an interactive user can halt the printing of a document after a paragraph if by then he has sufficient information.* While the documents are brief and do not go into great detail in describing a word, they give sufficient information to enable a user who is unfamiliar with a feature to employ that feature after reading the appropriate HELP document. For the user interested in further information, the HELP documents refer to longer more detailed documents if such documentation exists and is readily available.

---

*This feature has not, as of the date of this report, been implemented.

II. SPEAKEASY WORDS DEFINED IN THE HELP DOCUMENTS

The words listed on the next pages are those defined in the HELP documents of SPEAKEASY-3E.* The SPEAKEASY program by which they were obtained is the following.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAM TAB
* NEWPAGE
* MARGINS(1,77)
* WORDS=LIBINDEX(HELP)
* NU=NOELS(WORDS)
* PRINT ('THE NUMBER OF HELP WORDS IS 'NU)
* SPACE(4)
* TABULATE(WORDS)
* END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT...EXECUTE TAB
```

---

*The versions (i.e. SPEAKEASY-3C and 3D) operational at the time of this printing do not employ periods in the definition of logical operators.

THE NUMBER OF HELP WORDS IS NU = 241

WORDS	WORDS	WORDS	WORDS	WORDS	WORDS	WORDS	WORDS
ABS	CONJUGAT	ENDAUTOP	INTEGERS	MAT	OMITCLAS	SETGAUSS	TOTALINT
ACCURACY	CONSTRAI	ENDLOOP	INTEGRAL	MATRIX	ONERROR	SETINFIN	TOTINT
ACOS	CONTINUE	EQ	INTERP	MAX	OR	SETJACOB	TRACE
ACOT	COPY	ERF	INTERPOL	MAXOF	ORDERED	SETLAGUE	TRANSP
ADDGRAPH	COS	ERFC	INTPART	MAXOFCOL	ORDERER	SETLEGEN	TRANSPOS
AFAM	COSH	ERRORS	INTS	MAXOFROW	PLOTSYMB	SETLIB	UMAT
AMAT	COT	EXECUTE	INT2	MELD	PRINT	SETNULL	UNITMAT
AND	CREATE	EXP	INT4	MFAM	PRINTCLA	SETPLOT	UPPERTRI
ARRAY	CREATEME	FIN	INVERSE	MIN	PROD	SIGN	USE
ARRAY2D	CUMPROD	FOR	KEEP	MINOF	PRODCOLS	SIGNIFIC	USEMEMBE
ASIN	CUMSUM	FRAC	KEPT	MINOFCOL	PRODROWS	SIMEQ	VARIABLE
ASYMMAT	DATA	FRACPART	LABEL	MINOFROW	PROGRAM	SIN	VEC
ATAN	DEC	FREE	LE	MOVE	PUNCH	SINH	VECTOR
AUTOCORE	DEGRID	GAMMA	LENGTH	MYDOCS	QUIT	SIZE	VERSIONS
AUTOPRIN	DELETE	GE	LIBINDEX	MYHELP	RANDOM	SMAT	VFAM
AUTOTAB	DERIV	GEIGEN	LIBNAMES	MYLINKS	RANKED	SOLINDE2	VLABEL
AVERAGE	DERIVATI	GOTO	LINKLIB	NAMES	RANKER	SPACE	VOCABULA
A1D	DET	GRAPH	LINKULES	NE	READ	SPHBES	VSCALE
A2D	DIAGELS	GRID	LIST	NEUMANN	REAL	SPHBESN	VSIZE
BESSEL	DIAGMAT	GT	LISTHEAD	NEWGRAPH	REALPART	SQRT	WHERE
BESSELK	DIFFEQ	HELP	LISTMEMB	NEWPAGE	REAL4	SUM	WHEREVER
CGAMMA	DMAT	HENCEFOR	LOADDATA	NOCOLS	REAL8	SUMCOLS	WHOLE
CLEAR	DOCUMENT	HIERARCH	LOC	NOECHO	RECLASS	SUMROWS	WRITE
CLEARDAT	DOMAIN	HLABEL	LOCMAX	NOELS	RETURN	SUMSQ	ZEROS
COLARRAY	DUMP	HSCALE	LOCMIN	NOLIST	ROOTS	SUMSQCOL	
COLMAT	ECHO	HSIZE	LOCS	NOROOTS	ROWARRAY	SUMSQROW	
COLMAX	EDIT	IF	LOG	NOROWS	ROWMAT	SYMBOLS	
COLMIN	EIGENSYS	IMAG	LOGGAMMA	NOT	ROWMAX	SYMMAT	
COLWIDTH	EIGENVAL	IMAGPART	LOWERTRI	NOZEROS	ROWMIN	TABULATE	
COMPILE	EIGENVEC	INSERT	LT	NUMBERS	RUN	TAN	
CONJ	END	INTEG	MARGINS	OBJECT	SELECT	TIME	

III. ONE-LINE DEFINITIONS OF THE SPEAKEASY WORDS

On the following pages are one-line definitions of the words in the HELP documents. The first word of each sentence is the word being defined. The word is written with arguments used in a typical SPEAKEASY call. The SPEAKEASY program which lists the first line of each HELP document and thus generates these one-line definitions is below.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAM LISTHELP
*   MARGINS(1,120)
*   NEWPAGE
*   II=0;JJ=0
*   SPACE(6)
*   FOR I=1,NU
*   LIST MEMBER OBJECT(WORDS(I)) OF LIBRARY HELP TO 1
*   II=II+1
*   IF (II .EQ. 5) GO TO S
*   GO TO J
* S:SPACE(1)
*   II=0
* J:JJ=JJ+1
*   IF(JJ .EQ. 50) GO TO K
*   GO TO E
* K:SPACE(6)
*   JJ=0
* E:ENDLOOP I
*   END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT...EXECUTE LISTHELP
```

ABS(X) returns the absolute value of X.
ACCURACY(VAL) specifies the smallest number not to be taken as zero.
ACOS(X) defines the arc cosine of X.
ACOT(X) defines the arc cotangent of X.
ADDGRAPH(I:J) plots I versus J on the previous graph.

AFAM(X) defines a member of the array family.
AMAT is a synonym for ASYMMAT.
.AND. is the logical operator "and".
ARRAY(N:) defines a 1-dimensional N-component array.
ARRAY2D(N,M:) defines a 2-dimensional N-by-M array.

ASIN(X) defines the arc sine of X.
ASYMMAT(N:I,J,...,K) defines an N-by-N antisymmetric matrix.
ATAN(X) defines the arc tangent of X.
AUTOCORE specifies that the data storage space is in LCS.
AUTOPRINT(X,Y,...,Z) prints X,Y,...,Z each time they are redefined.

AUTOTAB specifies uniform tabulations.
AVERAGE(X) returns the average value of the elements of X.
A1D(N:) defines a 1-dimensional array.
A2D is a synonym for ARRAY2D.
BESSEL(NU,X) calculates cylindrical Bessel fn. of the first kind.

BESSELK(NU,X) returns the modified Bessel function K of X.
CGAMMA(X) returns the gamma function of X (X is complex).
CLEAR erases the entire active data storage.
CLEARDATA frees all previously defined numerical data.
COLARRAY(N:) defines a 2-dim. N-component column array.

COLMAT(N:) defines an N-by-1 matrix which is a column.
COLMAX(X) specifies the column with the maximum element of X.
COLMIN(X) specifies the column with the minimum element of X.
COLWIDTH(N) specifies the minimum column width to be used in printing.
%COMPILE returns the processor to the MANUAL mode from EDIT.

CONJ is a synonym for CONJUGATE.
CONJUGATE(X) returns the complex conjugate of X.
CONSTRAIN(A,B,...,C:X) constrains A,B,...,C according to condition X.
CONTINUE causes no operation to be performed.
%COPY N,M,K(I) copies statements N through M to K,K+I,K+2I,...

COS(X) returns the cosine of X.
COSH(X) defines the hyperbolic cosine of X.
COT(X) returns the cotangent of X.
CREATE MEMBER X ON LIBRARY Y creates a new member from cards.
CREATEMEMBER(XX,YY) creates member XX in library YY.

CUMPROD(X) defines the cumulative product of the elements of X.
CUMSUM(X) defines the cumulative sum of the elements of X.
DATA NAME marks the begining of a data file.
DEC(ZA1,ZA2,...,ZAN) converts A1,A2,...,AN from hex to decimal.
DEGRID(X0,XMAX,STEP) gives a grid for DIFFEQ.

%DELETE N,M deletes the statements N through M.
DERIV is a synonym for DERIVATIVE.
DERIVATIVE(F:X) finds the derivative of F with respect to X.
DET(X) defines the determinant of the matrix X.
DIAGELS(X) selects the diagonal elements of X.

DIAGMAT(N:I,J,...,K) defines an N-by-N diagonal matrix.
DIFFEQ(X,IN,T1,V1,T2,V2,...,TN,VN:XX,NOIT,METH) solves diff. eq.
DMAT is a synonym for DIAGMAT.
DOCUMENT is a library of larger documents.
DOMAIN specifies whether real or complex numbers are being used.

DUMP creates an easily read complete printout of all defined objects.
ECHO prints out the input along with results.
EDIT is a mode available for editing a program.
EIGENSYSTEM(SYMMAT) returns all the eigenvalues of SYMMAT.
EIGENVALS(X) gives the eigenvalues of the matrix X.

EIGENVECS(X) gives the eigenvectors of the matrix X.
END terminates a SPEAKEASY program or datafile.
ENDAUTOPRINT(X,Y,...,Z) turns off AUTOPRINT for objects X,Y,...,Z.
ENDLOOP N marks the end of the FOR loop N.
.EQ. is the relational operator "equal to".

ERF(X) calculates the error function of X.
ERFC(X) calculates the complementary error function of X.
ERRORS are ambiguities making it impossible to carry out a statement.
EXECUTE NAME executes the stored program, NAME.
EXP(X) defines the exponential function.

FIN is a synonym for ENDLOOP.
FOR N=START,STOP,INC designates a loop in a program.
FRAC is a synonym for FRACPART.
FRACPART(X) returns the fractional part of X.
FREE(N1,N2,...,NN) frees the definitions of the objects N1,N2,...,NN.

GAMMA(X) defines the gamma function of X.
.GE. is the relational operator "greater than or equal to".
GEIGEN(M,METRIC) finds the eigenvalues of the matrix M.
GOTO X causes execution to be transferred to the statement labeled X.
GRAPH(I:J) plots the members of I versus the object J.

GRID(I,J) defines a 1-dim. array of 101 points from I to J.
.GT. is the relational operator "greater than".
HELP contains a brief description of the SPEAKEASY words.
HENCEFORTH X IS Y redefines the word X to mean Y.
HIERARCHY is the order in which mathematical operations are done.

HLABEL='ANY MESSAGE' labels the horizontal scale of a graph.
HSCALE=(LEFT,RIGHT) specifies the horizontal limits of a graph.
HSIZE=X specifies the horizontal size of the graph.
IF (X) Y is a conditional statement for scalar operations.
IMAG(X) returns the imaginary part of X.

IMAGPART(X) returns the imaginary part of X.
%INSERT N(I) inserts the statements that follow at N, N+I,N+2I,...
INTEG is a synonym for INTEGRAL.
INTEGERS(I,J,K) defines an array with the integers I,I+K,I+2K,...,J.
INTEGRAL(F:X) defines the integral of F with a variable upper limit.

INTERP is a synonym for INTERPOL.
INTERPOL(Y,F,X) defines the function F at the points Y.
INTPART(X) returns the integer part of X.
INTS is a synonym for INTEGERS.
INT2(X) is a function that returns an integer2 number.

INT4(X) is a function that returns an integer4 number.
INVERSE(X) defines the inverse of the matrix X.
KEEP XX AS YY ON ZZ saves the SPEAKEASY object XX.
KEPT(YY ON ZZ) retrieves the data stored in data set YY on ZZ.
LABEL refers to a statement label.

.LE. is the relational operator "less than or equal to".
LENGTH(X) defines a scalar that is the number of elements in X.
LIBINDEX(Y) defines the names of the members of Y.
LIBNAMES returns the names of the SPEAKEASY libraries.
LINKLIB='XX' adds the LINKULE library XX to those available.

LINKULES is a library of operations that are FORTRAN subroutines.
LIST MEMBER X OF LIBRARY Y FROM I TO J lists lines I through J OF X.
LISTHEAD(N,XX,YY) lists N lines of member XX of library YY.
LISTMEMBER(XX,YY) lists member XX of library YY.
LOADDATA(A,NAME) loads A with N values from the data file NAME.

LOC is a synonym for LOCS.
LOCMAX(X) specifies the location of the maximum element of X.
LOCMIN(X) specifies the location of the minimum element of X.
LOCS(X) gives the indices of the nonzero (true) elements of X.
LOG(X) returns the natural logarithm of X.

LOGGAMMA(X) defines the natural logarithm of the gamma function of X.
LOWERTRI(X) returns the lower triangular part of X.
.LT. is the relational operator "less than".
MARGINS enables the user to contol the margins of the output.
MAT is a synonym for MATRIX.

MATRIX(N,M:) defines an N-by-M matrix.
MAX(X) specifies the value of the maximum element in X.
MAXOF(A,B,...,C) returns the largest element of A,B,...,C.
MAXOFCOL(X) returns the largest element in each column of X.
MAXOFROW(X) returns the largest element in each row of X.

MELD(I,J,...,K) gives an odometer ordering of elements of I,J,...,K.
MFAM(X) defines a member of the matrix/vector family.
MIN(X) specifies the value of the minimum element in X.
MINOF(A,B,...,C) returns the smallest element in A,B,...,C.
MINOFCOL(X) returns the smallest element in each column of X.

MINOFROW(X) returns the smallest element in each row of X.
%MOVE N,M,K(I) moves statements N through M to K,K+I,K+2I,...
MYDOCS is the library name for private documents.
MYHELP is the library name for private HELP documents.
MYLINKS is the library name for private linkules.

NAMES prints the names of all currently defined SPEAKEASY objects.
.NE. is the relational operator "not equal to".
NEUMANN(NU,X) returns cylindrical Bessel fns. of the second kind.
NEWGRAPH causes the next graph to be drawn on a new area of paper.
NEWPAGE causes the printer to start a new page.

NOCOLS(X) defines a scalar which is the number of columns in X.
NOECHO suppresses the echoing of input data.
NOELS(X) defines a scalar equal to the number of elements in X.
NOLIST indicates that the program is not to be printed with the output.
NOROOTS(F) gives the number of roots of the function F.

NOROWS(X) defines a scalar equal to the number of rows in X.
.NOT. is the logical operator "not".
NOZEROS(F) gives the number of zeros of the function F.
NUMBERS refers to numerical data.
OBJECT(I,J,...,K) generates an inplace name.

OMITCLASS suppresses the printing of the class.
ONERROR(X,Y) specifies the action to be taken after an error.
.OR. is the logical operator "or".
ORDERED(X) gives the elements of X in increasing order.
ORDERER(X) gives the indices of the ordered elements of X.

PLOTSYMB(N,M) specifies the symbols used and frequency of points.
PRINT(A,B,C,...,Z) specifies that A,B,C,...,Z are to be printed.
PRINTCLASS prints the class of structured objects.
PROD(X) defines the product of the elements of X.
PRODCOLS(X) multiplies the elements in each column of X.

PRODROWS(X) multiplies the elements in each row of X.
PROGRAM NAME gives a name to a program.
PUNCH(F:X) punches the object X on cards in the format F.
QUIT terminates execution of a SPEAKEASY program.
RANDOM(X) generates random numbers.

RANKED(X) gives the elements of X in increasing order.
RANKER(X) gives the indices of the ordered elements of X.
READ(F:X) reads data from cards punched in format F and puts
REAL is a synonym for REALPART.
REALPART(X) returns the real part of X.

REAL4(X) is a function that returns a real4 number.
REAL8(X) is a function that returns a real8 number.
RECLASS(A:B,C,...,Z) alters the structure of B,C,...,Z to
RETURN returns execution to the program calling the stored program.
ROOTS(F:X) finds the roots of the function F.

ROWARRAY(N:) defines a 2-dim. N-component array that is a row.
ROWMAT(N:) defines a 1-by-N matrix which is a row.
ROWMAX(X) specifies the row containing the maximum element of X.
ROWMIN(X) specifies the row containing the minimum element of X.
%RUN is equivalent to %COMPILE followed by the command EXECUTE.

SELECT(A,B,...,C:I) truncates or expands A,B,...,C using the index I.
SETGAUSS(N,X,W,XLO,XHI) returns Gauss-Legendre coords. and weights.
SETINFINITY(VAL) specifies an upper limit to the numbers printed.
SETJACOBI(N,X,W,A,B,TX,TW) defines Gauss-Jacobi coords. and weights.
SETLAGUERRE(N,X,W,A) returns Gauss-Laguerre coords. and weights.

SETLEGENDRE(N,X,W) returns Gauss-Legendre coord. and weights.
SETLIB(XX,YY) changes the name of library XX to YY.
SETNULL(VAL) specifies a lower limit to numbers printed.
SETPLOT(X,Y,Z) specifies BOX, NOBOX; SCALES, NOSCALES; LINES, POINTS.
SIGN(X) specifies whether X is positive or negative.

SIGNIFICANCE(N) gives the number of significant figures to be printed.
SIMEQ(A,B) solves a set of simultaneous linear equations.
SIN(X) returns the sine of X.
SINH(X) defines the hyperbolic sine of X.
SIZE=N,X specifies space for data and must be the first card.

SMAT is a synonym for SYMMAT.
SOLIMDE2(X:A,B,C:Y1,Y1P:YP) solves Y"+AY'+BY=C on grid X.
SPACE(N) skips N lines.
SPHBES(L,X) returns the spherical Bessel fn. of the first kind.
SPHBESN(L,X) returns the spherical Bessel fn. of the second kind.

SQRT(X) defines the square root of X.
SUM(X) sums the elements of X.
SUMCOLS(X) sums the elements in each column of X.
SUMROWS(X) sums the elements in each row of X.
SUMSQ(X) sums the squares of the elements of X.

SUMSQCOLS(X) sums the squares of the elements in each column.
SUMSQROWS(X) sums the squares of the elements in each row.
SYMBOLS designate mathematical operations or special cards.
SYMMAT(N:I,J,...,K) defines a symmetric N-by-N matrix.
TABULATE(A,B,...,C) prints 1-dim. objects A,B,...,C in tabular form.

TAN(X) returns the tangent of X.
TIME gives the time from which one starts.
TOTALINT(F:X) defines the definite integral of F over the array X.
TOTINT is a synonym for TOTALINT.
TRACE(X) gives the trace of the matrix X.

TRANSP is a synonym for TRANSPOSE.
TRANSPOSE(X) defines the transpose of X.
UMAT is a synonym for UNITMAT.
UNITMAT(N) defines an N-by-N unit matrix.
UPPERTRI(X) returns the upper triangular part of X.

USE MEMBER X OF LIBRARY Y causes the deck X to be used as input.
USEMEMBER(XX,YY) causes input of the member XX of library YY.
VARIABLE(I,J) defines a 1-dim. array of 101 points from I to J.
VEC is a synonym for VECTOR.
VECTOR(N:) defines an N-component vector.

VERSIONS refers to the various versions of the SPEAKEASY processor.
VFAM(X) defines a member of the matrix/vector family.
VLABEL='ANY MESSAGE' labels the vertical scale.
VOCABULARY generates a list of all currently defined words.
VSCALE=(BOTTOM,TOP) specifies the vertical limits of a graph.

VSIZE=Y specifies the vertical size of the graph.
WHERE is a synonym for WHEREVER.
WHEREVER(X) Y is a conditional statement for array operations.
WHOLE(X) returns the integer part of X.
WRITE(F:X) prints the defined object X in the format F.

ZEROS(F:X) finds the zeros of the function F.

IV. THE SPEAKEASY HELP DOCUMENTS

The HELP documents* of the SPEAKEASY-3E follow. They give a brief definition of each word in the HELP library. This listing of the HELP documents has been obtained by the following program.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PROGRAM HELPDOC
*   NEWPAGE
*   I1=ARRAY(:0,7,11,18,25,33,41,49,53,58,65,71,78,83,91,98,106,113,122)
*   I2=ARRAY(:130,139,149,157,164,172,180,186,190,197,203,211,216,226)
*   DOCEND=ARRAY(:I1,I2,229,235,241)
*   FOR II=1,35
*   EN=DOCEND(II+1)
*   ST=DOCEND(II)+1
*   FOR KK=ST,EN
*   HELP OBJECT(WORDS(KK))
*   SPACE(2)
*   ENDLOOP KK
*   NEWPAGE
*   ENDLOOP II
*   END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT... EXECUTE HELPDOC
```

---

*In the HELP documents the notation 4E60 is the computer form of the scientific notation 4×10^{60}.

ABS(X) returns the absolute value of X.
ABS(X) defines an object with the same structure as X but with elements of the result equal to the absolute values of the corresponding elements of X.

ACCURACY(VAL) specifies the smallest number not to be taken as zero. Whenever the equality of two numbers is checked during the execution of a program, any number less than a certain number is regarded as zero. The value of this number may be set to be any number VAL by the statement ACCURACY(VAL).
If not specified, the value of VAL is 10E-8.

ACOS(X) defines the arc cosine of X.
ACOS(X) defines an object with the same structure as X with elements that are the arc cosine of the corresponding elements of X.
All the elements Xi of X must be real and satisfy the condition |Xi|<1 or the condition |Xi|=1.

ACOT(X) defines the arc cotangent of X.
ACOT(X) defines an object with the same structure as X with elements that are the arc cotangent of the corresponding elements of X.
All elements Xi of X must be real.

ADDGRAPH(I:J) plots I versus J on the previous graph.
ADDGRAPH has the same meaning as graph except a new area of paper is not used. Design statements except the scale statements and the label statements are reexamined prior to adding the graph. Thus the plotting format may be changed for each additional graph.
I is plotted along the vertical axis and J is along the horizontal. All objects should be 1-dimensional and real and have the same number of elements. A 2-dimensional object in I is treated as several 1-dimensional objects each consisting of a row of the original 2-dimensional object. Thus a 2-dimensional object in I must have as many columns as J has elements.
If no previous graph has been drawn, ADDGRAPH acts as GRAPH.
ADDGRAPH is available only in version GRAPHEZ.

AFAM(X) defines a member of the array family.
The result has the same structure as X but is an array. If X is a vector or a 1-dimensional array, the result is a 1-dimensional array. If X is a matrix or a 2-dimensional array, the result is a 2-dimensional array.

AMAT is a synonym for ASYMMAT.
AMAT(N:I,J,...,K) defines an N-by-N antisymmetric matrix. The element list is used to fill the lower triangular part by rows. The portion above the diagonal is then filled by making the matrix antisymmetric. If any object in the list defining the elements is structured, the elements of that structured object are used.

.AND. is the logical operator "and".
The periods must appear on both sides of the operator.

ARRAY(N:) defines a 1-dimensional N-component array.
If no further arguments are given, all the components are set equal to zero.
An alternative form is A1D.
ARRAY(:I,J,...,K) defines a 1-dim. array and specifies the elements.
The elements are preset by the element list I,J,...,K.
If any argument of the defining element list is structured, the elements of the structured object are used. If a complex element is encountered, then a complex 1-dimensional array is defined.
ARRAY(N:I,J,...,K) defines a 1-dim. N-component array.
The elements are preset by the element list I,J,...,K.
If any argument in the element list is structured, the elements of the structured object are used. If a complex element is encountered, then a complex 1-dimensional array is defined. If all N-components are not specified, the unspecified components are set equal to zero.
ARRAY(N,M:) defines a 2-dimensional N-by-M array.
If no further arguments are given, all elements are set equal to zero.
An alternative form is ARRAY2D.
ARRAY(N,M:I,J,...,K) defines a 2-dim. N-by-M array with preset els.
The elements are preset by the element list I,J,...,K. If any argument in the element list is structured, then the elements of the structured object are used. If a complex element is encountered, then a two-dimensional N-by-M complex array is defined. All the elements not preset by the element list are set equal to zero.

ARRAY2D(N,M:) defines a 2-dimensional N-by-M array.
If no further arguments are given, all the elements are set equal to zero.
An alternative form is ARRAY(N,M:).
A shortened form is A2D.
ARRAY2D(N,M:I,J,...,K) defines a 2-dim. N-by-M array.
The elements are preset by the element list I,J,...,K.
If any object of the defining element list is structured, the elements of the structured object are used. If a complex element is encountered, then a 2-dimensional N-by-M complex array is defined. All elements not preset by the element list are set equal to zero.

ASIN(X) defines the arc sine of X.
ASIN(X) defines an object with the same structure as X with elements that are the arc sine of the corresponding elements of X.
All the elements Xi of X must be real and satisfy the condition $|Xi|<1$ or the condition $|Xi|=1$.

ASYMMAT(N:I,J,...,K) defines an N-by-N antisymmetric matrix. The element list is used to fill the lower triangular part by rows. The portion above the diagonal is then filled by making the matrix antisymmetric. If any object in the list defining the elements is structured, the elements of that structured object are used.
A shortened form is AMAT.

ATAN(X) defines the arc tangent of X.
ATAN(X) defines an object with the same structure as X with elements that are the arc tangent of the corresponding elements of X.
All elements Xi of X must be real.

AUTOCORE specifies that the data storage space is in LCS. AUTOCORE means that the largest amount of LCS available is to be used for the job.
If AUTOCORE is to be used, it should be the first card in the SPEAKEASY deck.

AUTOPRINT(X,Y,...,Z) prints X,Y,...,Z each time they are redefined. Each object of the name list X,Y,...,Z is printed whenever it appears on the left of an equation.
AUTOPRINT prints all objects as they are defined or altered. AUTOPRINT is turned off by the statement ENDAUTOPRINT.

AUTOTAB specifies uniform tabulations.
The value of N for the column width is automatically set at 10 greater than the number of significant figures so that the print is uniform for all objects composed of real numbers.

AVERAGE(X) returns the average value of the elements of X. X is a structured object.

A1D(N:) defines a 1-dimensional array.
If no further arguments are given, all the components are set equal to zero.
A1D is a synonym for ARRAY(N:).
A1D(:I,J,...,K) defines a 1-dim. array and specifies the elements. The components are preset by the element list I,J,...,K. If any argument in the element list is structured, then the elements of the structured object are used. If a complex element is encountered, then a complex 1-dimensional array is defined.
A1D(N:I,J,...,K) defines a 1-dim. N-component array. The components are preset by the element list I,J,...,K. If any argument of the element list is structured, then the elements of the structured object are used. If a complex element is encountered, then a complex 1-diminsional array is defined. If all N-components are not specified, the unspecified components are set equal to zero.

A2D is a synonym for ARRAY2D.
A2D(N,M:) defines a 2-dimensional N-by-M array.
If no further arguments are given, the elements are set equal to zero.
A2D(N,M:I,J,...,K) defines a 2-dim. N-by-M array with preset els.
The elements are preset by the element list I,J,...,K.
If any object of the defining element list is structured, the elements of the structured object are used. If a complex element is encountered, then a 2-dimensional N-by-M complex array is defined. All elements not preset by the element list are set equal to zero.

BESSEL(NU,X) calculates cylindrical Bessel fn. of the first kind.
NU and X must be real and nonnegative. There exists a restriction in the SPEAKEASY linkule that (X+NU)<400.

BESSELK(NU,X) returns the modified Bessel function K of X.
The resultant is the modified Bessel function K sub NU of X.
NU and X must be real and nonnegative. There exists a restriction in the SPEAKEASY linkule that (X+NU)<400.

CGAMMA(X) returns the gamma function of X (X is complex).
The result has the same structure as X with elements equal to the gamma function of the corresponding elements of X.
If X is real, the gamma function of X may be obtained from GAMMA(X).

CLEAR erases the entire active data storage.
After the execution of the statement CLEAR there are no user-defined objects, programs, or henceforth definitions.

CLEARDATA frees all previously defined numerical data.

COLARRAY(N:) defines a 2-dim. N-component column array.
If no further arguments are given, all N-components are set equal to zero.
COLARRAY(:I,J,...,K) defines a 2-dim. column array with preset els.
The components are preset by the element list I,J,...,K . If any argument of the element list is structured, them the elements of that structured object are used. If a complex element is encountered, then a comlex 2-dimensional column array is defined.
COLARRAY(N:I,J,...,K) defines a 2-dim. N-component column array.
The components are preset by the element list I,J,...,K . If any argument of the element list is structured, then the elements of that structured object are used. If a complex element is encountered, them a complex 2-dimensional column array is defined. If all N-components are not preset by the element list the unspecified components are set equal to zero.

COLMAT(N:) defines an N-by-1 matrix which is a column. If no further arguments are given, all N elements are set equal to zero.

COLMAT(:I,J,...,K) defines a column matrix by specifying the els. The element list I,J,...,K is used to preset the elements of the column matrix. If one of the arguments of the element list is structured, the elements of that structured object are used. If one of the elements is complex, the column matrix is complex.

COLMAT(N:I,J,...,K) defines an N-by-1 column matrix with preset els. The elements are specified by the element list I,J,...,K. If one of the arguments of the list is structured, the elements of the structured object are used. The column matrix is complex if one of the preset elements is complex. If all N elements are not preset by the element list, the unspecified elements are set equal to zero.

COLMAX(X) specifies the column with the maximum element of X.

COLMIN(X) specifies the column with the minimum element of X.

COLWIDTH(N) specifies the minimum column width to be used in printing. This statement prevents the column width of the print routine from being less than N characters. If N is 10 larger than the number of significant figures desired, the print will be uniform for all objects composed of real numbers.

%COMPILE returns the processor to the MANUAL mode from EDIT. %COMPILE is used after a program has been edited to return from the EDIT mode to the MANUAL mode.

CONJ is a synonym for CONJUGATE.

CONJ(X) returns the complex conjugate of X. CONJ(X) defines an object with the same structure as X but with elements equal to the complex conjugate of the corresponding elements of X.

CONJUGATE(X) returns the complex conjugate of X. CONJUGATE(X) defines an object with the same structure as X but with elements equal to the complex conjugate of the corresponding elements of X.

A shortened form is CONJ.

CONSTRAIN(A,B,...,C:X) constrains A,B,...,C according to condition X. A,B,...,C are 1-dimensional objects of the same length. X is a logical expression of the same length as the objects to the left of the colon. The objects to the left of the colon are truncated so that they contain only those elements satisfying the logical expression X.

All 1-dimensional objects must have the same length.

For an example refer to the SPEAKEASY-3 manual.

CONTINUE causes no operation to be performed.
The CONTINUE statement is a nonoperational statement to which a label may be attached. CONTINUE is restricted to the program mode. It is ignored in the manual mode.
CONTINUE is a restricted word.

%COPY N,M,K(I) copies statements N through M to K,K+I,K+2I,...
%COPY N,M,K copies statements N through M to K,K+1,K+2,...
%COPY N copies statement N to the last position.
%COPY N,M copies statements N through M to the last positions.
%COPY is a function available in the EDIT mode.

COS(X) returns the cosine of X.
COS(X) defines an object with the same structure as X but the elements of the result are the cosine of the corresponding elements of X.
Each element Xi of X must satisfy the condition |Xi|<1.E15.

COSH(X) defines the hyperbolic cosine of X.
COSH(X) defines an object with the same structure as X but with elements equal to the hyperbolic cosine of the corresponding elements of X.
The elements Xi of X must be real and must satisfy the condition |Xi|<170.

COT(X) returns the cotangent of X.
COT(X) defines an object with the same structure as X but the elements of the result are the cotangent of the corresponding elements of X.
Each element Xi of X must satisfy the condition |Xi|<1.E15.

CREATE MEMBER X ON LIBRARY Y creates a new member from cards.
All cards following the card CREATE MEMBER X on LIBRARY Y until the card ENDCREATE is encountered are put into the member with the name X in library Y.
If the library reference is omitted, it will refer to the library MYPROC.

CREATEMEMBER(XX,YY) creates member XX in library YY.
XX is the name of the member that is created.
YY is the name of the library into which XX is put.
All the cards following the CREATEMEMBER card are put into the member named XX in library YY until the card ENDCREATE is encountered.
If the library reference is omitted, the member will be put in the library MYPROCS.
An alternative form is CREATE MEMBER XX on LIBRARY YY.

CUMPROD(X) defines the cumulative product of the elements of X.
CUMPROD(X) defines an object with the same structure as X with elements that are the cumulative product of all previous elements of X. Two-dimensional objects are treated row by row.

CUMSUM(X) defines the cumulative sum of the elements of X. CUMSUM(X) defines an object with the same structure as X with elements that are the cumulative sum of all previous elements of X. Two-dimensional objects are treated row by row.

DATA NAME marks the begining of a data file. NAME is the name given to the data file. The data file is terminated by the single word END. All cards between the card DATA NAME and the card END contain data that may be punched in any format. It is read into a specified array by a LOADDATA statement.
DATA is a restricted word.

DEC(ZA1,ZA2,...,ZAN) converts A1,A2,...,AN from hex to decimal. The hexidecimal numbers A1,A2,...,AN must be prefixed by a letter to prevent their conversion to numbers by SPEAKEASY. They may be 7 hexidecimal figures in length. The first character must be a letter so they are 8-character words.
The result is an array with elements that are the decimal equivalents of the hex arguments.

DEGRID(X0,XMAX,STEP) gives a grid for DIFFEQ.
XMAX is a 1-dimensional array of the form X1,X2,...XN.
STEP is a 1-dimensional array giving the increments to be added to each Xi; STEP=S1,S2,...,SN.
The result is a 1-dimensional array. The first 48 elements of the result are specially constructed for DIFFEQ. X(49)=X0. The remaining elements are formed by adding S1 to X0 until X1 is reached; S2 to X1 until X3 is reached; etc. The elements beginning with the 49th are
X0,X0+S1,X0+2S1,...,X1,X1+S2,X1+2S2,...,X3,...,XN.
This is not equivalent to GRID.

%DELETE N,M deletes the statements N through M.
%DELETE N deletes the statement numbered N.
%DELETE is one of the functions available in the EDIT mode.

DERIV is a synonym for DERIVATIVE.
DERIV(F:X) finds the derivative of the function F with respect to X. X is a 1-dimensional array and so is the result. Each element of the result is dF/dXi, where Xi is the corresponding element of X.

DERIVATIVE(F:X) finds the derivative of F with respect to X. X is a 1-dimensional array and so is the result. Each element of the result is dF/dXi, where Xi is the corresponding element of X.
A shortened form is DERIV.

DET(X) defines the determinant of the matrix X.

DIAGELS(X) selects the diagonal elements of X.
The diagonal elements are given in the order in which they occur in the square matrix X. The result is a 1-dimensional array.

DIAGMAT(N:I,J,...,K) defines an N-by-N diagonal matrix.
All elements are zero except for those along the diagonal, which are defined by the element list I,J,...,K. If any object in the element list is structured, the elements of that structured object are used.
A shortened form is DMAT.

DIFFEQ(X,IN,T1,V1,T2,V2,...,TN,VN:XX,NOIT,METH) solves diff. eq.
DIFFEQ solves N coupled differential equations, each of order less than or equal to 4. N must be less than or equal to 10.
The result is an array whose rows contain the solution for each dependent variable evaluated at each point of the grid XX.
The Kth row of the result contains the solution for the Kth dependent variable evaluated at each point of XX.
X is a grid formed by DEGRID.
IN is a 2-dimensional array containing the initial conditions of the dependent variables and their derivatives.
The first column of IN lists the subscript of each variable; the second column lists the order of the derivative; the third column lists the initial value. The ordering of the rows is arbitrary.
TK is a 2-dimensional array referring to the Kth equation. TK gives the ordering of the terms.
The first column of TK gives the subscript of the variable in a term; the second column lists the order of the derivative of the variable.
VK is a 2-dimensional array refering to the Kth equation. The rows of VK consist of the coeficients of the variables in the Kth equation.
The coefficients are evaluated at each point of the grid formed by DEGRID. The first row of VK contains the coefficient of the second term of TK; the second row of VK corresponds to the coefficient for the varible of the third row of TK; etc.
XX is a grid returned by DIFFEQ.
XX is formed by deleting the first 48 elements of X.
NOIT is an optional argument containing the number of corrective iterations. The default value is 2.
METH is an optional argument giving the degree of the method for each equation. METH is a 1-dimensional array. The default value is the order of the equation +4.
For a more detailed explanation and examples, refer to the linkule document.

DMAT is a synonym for DIAGMAT.
DMAT(N:I,J,...,K) defines an N-by-N diagonal matrix.
All elements are zero except for those along the diagonal, that are defined by the element list I,J,...,K. If any object in the element list is structured, the elements of that structured object are used.

DOCUMENT is a library of larger documents.
The document library contains the documents of the system other than the HELP documents.

DOMAIN specifies whether real or complex numbers are being used. If no domain is specified by the user, it is assumed to be the real domain. The domain may be altered at will by the user by the statements DOMAIN(REAL) or DOMAIN(COMPLEX).

DOMAIN(REAL) specifies that real numbers are to be used. Any calculation leading to an imaginary or complex result is treated as an error.

DOMAIN(COMPLEX) specifies that complex numbers are allowed.

DUMP creates an easily read complete printout of all defined objects. DUMP enables the user to examine all the information at a given point in a calculation. After the printout, the calculation continues.

DUMP is not available in TSO.

ECHO prints out the input along with results.
The first line of output is the user's input. Immediately following are the results or the response to the input.

ECHO is default for batch jobs; NOECHO is default for the interactive operation.

EDIT is a mode available for editing a program.
The EDIT mode is automatically entered when one of the words PROGRAM, PROCEDURE, or DATA is encountered. In the EDIT mode statements are assigned successive integer values starting with 1. The program may be altered by using control functions that are available in the EDIT mode. These are selected by a % in the first field of the input card. The functions are:

%LIST
%COPY
%DELETE
%INSERT
%MOVE
%COMPILE
%RUN
%

For a description of most of these functions, type HELP FUNCTION where FUNCTION is the name of the function desired without the %.

EDIT XX returns the processor to the edit mode with the program XX. XX is the name of a previously defined program that is to be edited.

The EDIT mode is available on all processors but it is normally not used except for interactive operations.

EIGENSYSTEM(SYMMAT) returns all the eigenvalues of SYMMAT. The eigenvalues are returned as a vector. If the SPEAKEASY call is Y=EIGENSYSTEM(SYMMAT), then Y is a vector containg the eigenvalues of SYMMAT.

SYMMAT must be a real symmetric matrix.

EIGENSYTEM(SYMMAT:+M) returns the M largest eigenvalues of SYMMAT. The M eigenvalues are in descending order as the M-components of a vector.

SYMMAT must be a real symmetric matrix.

EIGENSYSTEM(SYMMAT:-M) returns the M lowest eigenvalues of SYMMAT. The M lowest eigenvalues are returned in ascending order as the M-components of a vector.

SYMMAT must be a real symmetric matrix.

EIGENSYSTEM(X,EVECS) returns the eigenvalues and eigenvectors of X. If the SPEAKEASY call is Y=EIGENSYSTEM(X,EVECS), then Y is a vector containg the eigenvalues of X. The corresponding eigenvectors are stored row-wise in the N-by-N matrix, EVECS.

X must be a real symmetric matrix.

EIGENSYSTEM(X,V:+M) returns the M largest eigenvalues and eigenvectors. The M largest eigenvalues are returned in descending order as a vector. Their corresponding eigenvectors are returned as the rows of the M-by-N matrix V.

X must be a real symmetric matrix.

EIGENSYSTEM(X,V:-M) returns the M smallest eigenvalues and eigenvectors. The M smallest eigenvalues of X are returned as a vector. V is returned as an M-by-N matrix containing M eigenvectors corresponding to the M smallest eigenvalues. The eigenvectors are stored row-wise in V.

X must be a real symmetric matrix.

EIGENVALS(X) gives the eigenvalues of the matrix X. The eigenvalues are listed in order of descending values. X must be a real symmetric matrix.

EIGENVECS(X) gives the eigenvectors of the matrix X. EIGENVECS(X) defines a matrix whose rows are the eigenvectors of X. The resulting matrix is unitary.

END terminates a SPEAKEASY program or datafile. During execution in the program mode, the single word END corresponds to a return. In the manual mode of SPEAKEASY, the word END is ignored.

ENDAUTOPRINT(X,Y,...,Z) turns off AUTOPRINT for objects X,Y,...,Z.

ENDLOOP N marks the end of the FOR loop N. N is the name of the FOR loop that occurs in the corresponding FOR statement.

ENDLOOP is a restricted word.

.EQ. is the relational operator "equal to". The periods must appear on both sides of the operator.

ERF(X) calculates the error function of X.
The result has the same structure as X with elements equal to the error function of the corresponding elements of X.
The elements of X must be real.

ERFC(X) calculates the complementary error function of X.
The result has the same structure as X with elements that correspond to the complementary error functions of the corresponding elements of X.
The elements of X must be real.

ERRORS are ambiguities making it impossible to carry out a statement. There are two types of errors in SPEAKEASY. They are syntax errors and execution errors. Syntax errors are those common to all computer languages and are due to misuse of symbols, such as parenthesis imbalance. Such errors are detected before execution of the SPEAKEASY statement. Execution errors relate to an inconsistency in the use of defined objects and are unique to SPEAKEASY. Errors encountered during compilation of a SPEAKEASY program do not inhibit later execution.

EXECUTE NAME executes the stored program, NAME.
The stored program is executed by this command and continues until a RETURN statement or END statement is encountered. These statements terminate the execution of the stored program and the statement following the one calling for the execution of the stored program is then executed.
EXECUTE NAME:LABEL executes the program NAME beginning at LABEL.
EXECUTE NAME:LABEL causes the stored program with this name to begin execution at the designated internal label.
EXECUTE statements may occur in the manual mode or in any stored program. In the manual mode, the Execute statement should occur alone; not as part of a multistatement card.
EXECUTE is a restricted word.

EXP(X) defines the exponential function.
EXP(X) defines an object with the same structure as X but with elements equal to the exponential of the corresponding elements of X.
Each element Xi of X must satisfy the conditions Real(Xi)<170 and |Imag(Xi)|<5E6.

FIN is a synonym for ENDLOOP.
FIN N marks the end of the FOR loop N.
N is the name of the loop appearing in the corresponding FOR statement.
FIN is a restricted word.

FOR N=START,STOP,INC designates a loop in a program.
N is the name of a scalar that may appear in any context in the loop without altering its value. START, STOP, and INC are scalar expressions not involving N.
START is the initial value of N.
STOP is the final value of N.
INC is the increment to be added to N every time the loop is repeated. If INC is not specified, its value is assumed to be 1.
A FOR loop is terminated by an ENDLOOP statement or by a FIN statement.
Up to 10 nested FOR loops are allowed. A FOR loop started within a FOR loop must be terminated within that loop.
Since FOR loops are neither efficient nor compact in SPEAKEASY, they should be avoided. Logical transfers into and out of FOR loops should be used with extreme caution.
FOR is a restricted word.

FRAC is a synonym for FRACPART.
FRAC(X) returns the fractional part of X.
FRAC(X) defines an object with the same structure as X but the elements of the result are the fractional part of the corresponding elements of X.

FRACPART(X) returns the fractional part of X.
FRACPART(X) defines an object with the same structure as X but the elements of the result are the fractional part of the corresponding elements of X.
A synonym is FRAC.

FREE(N1,N2,...,NN) frees the definitions of the objects N1,N2,...,NN.
N1,N2,...,NN are names of defined objects.

GAMMA(X) defines the gamma function of X.
GAMMA(X) defines an object with the same structure as X but with elements equal to the gamma function of the corresponding elements of X.
The elements Xi of X must be real and must satisfy the condition 10E-50<Xi<56.

.GE. is the relational operator "greater than or equal to".
The periods must be on both sides of the operator.

GEIGEN(M,METRIC) finds the eigenvalues of the matrix M.
M is a real symmetric matrix in a non-orthogonal basis.
METRIC is the positive definite matrix of the inner products of the basis states. The eigenvalues of M are returned as the components of a vector.
GEIGEN(M,METRIC,EVECS) finds the eigenvalues and eigenvectors of M.
M is a real symmetric matrix in a nonorthogonal basis.
METRIC is the positive definite matrix of the inner products of the basis states. EVECS is a matrix containing the eigenvectors of M stored row-wise. The eigenvalues of M are returned as the components of a vector.

GOTO X causes execution to be transferred to the statement labeled X. The GOTO statement is used to alter the sequence of execution of statements. Logical branches are created by combining an IF statement with a GOTO statement.

GOTO is a restricted word.

An equivalent form is GO TO.

GRAPH(I:J) plots the members of I versus the object J. I is in the vertical direction and J is along the horizontal.

All objects should be 1-dimensional and real and have the same number of elements. A 2-dimensional object in I is treated as several 1-dimensional objects, each consisting of a row of the original 2-dimensional object. Thus a 2-dimensional object in I must have as many columns as J has elements.

Each time GRAPH is encountered, a graph is drawn on a new area of paper. All of the design statements should accompany GRAPH.

The graphical output from GRAPH in the program mode is plotted by CALCOMP.

GRAPH is available only in version GRAPHEZ.

GRID(I,J) defines a 1-dim. array of 101 points from I to J. The increment is chosen by the computer so that the points are equally spaced. Both points I and J are included.

A synonym is VARIABLES(I,J).

GRID(I,J,K) defines a 1-dim. array I,J+K,I+2K...until J is reached. The last element is equal to or less than J depending on the increment K.

GRID(I,J,K) is not equal to VARIABLE(I,J,K) unless K is such that I+NK=J.

.GT. is the relational operator "greater than". The periods must be on both sides of the operator.

HELP contains a brief description of the SPEAKEASY words. An alphabetical listing of the words in the HELP library may be obtained by the statements

```
CATALOG=LIBINDEX(HELP)
TABULATE(CATALOG)
```

To obtain the description of a single SPEAKEASY word XX, the input statement is

```
HELP XX
```

All the words and a description of each may be obtained by the single statement

```
USE MEMBER LISTHELP
```

A 1-line description of a SPEAKEASY word may be obtained by the statement

```
LIST MEMBER XX OF LIBRARY HELP TO 1.
```

HENCEFORTH X IS Y redefines the word X to mean Y. After this statement, anytime the word X is encountered it will be treated as if it were the word Y.

The word 'IS' in the expression is arbitrary. Any word may be used.

Restricted words may not be given synonyms by this method.

HIERARCHY is the order in which mathematical operations are done. The HIERARCHY in SPEAKEASY is:

** raising to a power,
/,* division and multiplication,
+,- addition and subtraction.

Operations of the same hierarchy are done left to right.

HLABEL='ANY MESSAGE' labels the horizontal scale of a graph. The desired label may be anything. Its beginning and end are marked by apostrophes in the specifying statement.

HLABEL is available only in version GRAPHEZ.

HSCALE=(LEFT,RIGHT) specifies the horizontal limits of a graph. LEFT is the left limit and RIGHT is the right limit of the scale. Values are labeled at inch marks. The default limits for the horizontal scale are 0 for the left and 10 for the right.

HSCALE is used for graphical output.

HSIZE=X specifies the horizontal size of the graph. X is in inches. The default is 10 inches.

HSIZE is used for graphical output.

IF (X) Y is a conditional statement for scalar operations. X is a scalar expression. If the numerical value of the expression X is nonzero, then the associated statement Y is carried out; otherwise the statement Y is ignored.

IF is a restricted word.

IMAG(X) returns the imaginary part of X. IMAGINARY(X) defines an object with the same structure as X but with elements that are the imaginary part of the corresponding elements of X.

A synonym is IMAGPART.

IMAGPART(X) returns the imaginary part of X. IMAGPART(X) defines an object with the same structure as X but with elements that are the imaginary part of the corresponding elements of X.

A synonym is IMAG.

%INSERT N(I) inserts the statements that follow at N, N+I,N+2I,...
%INSERT N(I) inserts the statements that follow the words
%INSERT N(I) into the positions N,N+I,N+2I,...
%INSERT N inserts the statements that follow at N, N+1,N+2,...
%INSERT N inserts the statements that follow the insert command
into locations labeled by N,N+1,N+2,...
% stops the insert.
%INSERT is one of the functions available in the EDIT mode.

INTEG is a synonym for INTEGRAL.
INTEG(F:X) defines the integral of F with a variable upper limit.
X is a 1-dimesional array (X1,X2,X3,...,XN). The result is
a 1-dimensional array the same size as X. Each element of the
result is the integral from X1 to Xi where Xi is the
corresponding element of X. The integration is done by the
trapezoidal rule.

INTEGERS(I,J,K) defines an array with the integers I,I+K,I+2K,...,J.
The array ends when the value J is reached or passed.
INTEGERS(I,K) defines an array with the integers I,I+1,I+2,...,K.
If K is less than I, INTEGERS(I,K) defines the integer array
I,I-1,I-2,...,K.
A shortened form is INTS.

INTEGRAL(F:X) defines the integral of F with a variable upper limit.
X is a 1-dimensional array (X1,X2,X3,...,XN). The result is
a 1-dimensional array the same size as X. Each element of the
result is the integral from X1 to Xi where Xi is the corresponding
element of X. The integration is done by the trapezoidal rule.
A shortened form is INTEG.

INTERP is a synonym for INTERPOL.
INTERP(Y,F,X) defines the function F at the points Y.
INTERP(Y,F,X) is an interpolation method which is cubic
except in the first and last intervals where it is quadratic.
The result is an array with the values of the function
F evaluated at the points Y. X and Y are arrays.
If points in Y are outside the range, the interpolation is
based on the last 3 points in the range. F(X) must be given.

INTERPOL(Y,F,X) defines the function F at the points Y.
INTERPOL(Y,F,X) is an interpolation method which is cubic
except in the first and last intervals where it is quadratic.
The result is an array with the values of the function
F evaluated at the points Y. X and Y are arrays.
If points in Y are outside the range, the interpolation is
based on the last 3 points in the range. F(X) must be given.
A synonym is INTERP.

INTPART(X) returns the integer part of X.
INTPART(X) defines an object with the same structure as X but the
elements of the result are the integer part of the corresponding
elements of X.
A synonym is WHOLE.

INTS is a synonym for INTEGERS.
INTS(I,J,K) defines an array with the integers I,I+K,I+2K,...,J. The array ends when the value J is reached or passed.
INTS(I,K) defines an array with the integers I,I+1,I+2,...,K. If K is less than I, INTS(I,K) defines the integer array I,I-1,I-2,...,K.

INT2(X) is a function that returns an integer2 number. X may be real8, real4, or integer4. The result is INTEGER2.

INT4(X) is a function that returns an integer4 number. X may be real4, real8, or integer2. The result is integer4.

INVERSE(X) defines the inverse of the matrix X. The result is a matrix.
An alternative statement to obtain the inverse of the matrix X is 1/X.

KEEP XX AS YY ON ZZ saves the SPEAKEASY object XX. The command means that object XX is to be copied onto data set ZZ and given the name YY. The object XX is recovered by a KEPT statement.
The data set ZZ is a private data set belonging to the user. It is created by including cards of the form

```
//ZZ DD DSN=MYNAME,DISP=(NEW,CATLG),
//      SPACE=(TRK,(5,5,2)),DCB=(RECFM=FB,BLKSIZE=400,LRECL=80)
```

immediately following the // EXEC SPEAKEZ card.
To use the data set, change DISP=(NEW,CATLG) to DISP=(OLD,KEEP). To delete the data set when it is no longer needed, use DISP=(OLD,DELETE).

KEPT(YY ON ZZ) retrieves the data stored in data set YY on ZZ. KEPT(YY ON ZZ) is a normal SPEAKEASY function, which may be imbedded in a statement.
YY is the name assigned to the data in the KEEP statement.
ZZ is the name of the data set on which the data are stored.
To use the data set ZZ, include a card of the form

```
//ZZ DD DSN=MYNAME,DISP=(OLD,KEEP),UNIT=2321
```

immediately following the // EXEC SPEAKEZ card.
To delete the data set when it is no longer needed, change DISP=(OLD,KEEP) to DISP=(OLD,DELETE).

LABEL refers to a statement label.
SPEAKEASY statements are labeled by a name of 8 symbols or less, beginning with an alphabetic character. This label is separated from the statement by a colon. The label may be in any column but preceeds the statement.

.LE. is the relational operator "less than or equal to". The periods must appear on both sides of the operator.

LENGTH(X) defines a scalar that is the number of elements in X. If X is undefined, the result is zero.
A synonym is NOELS.

LIBINDEX(Y) defines the names of the members of Y. LIBINDEX(Y) defines a literal 1-dimensional array with the names of the members of the library Y as components of the array.

LIBNAMES returns the names of the SPEAKEASY libraries. LIBNAMES defines an array whose elements are the names of the libraries attached to the SPEAKEASY processor.

LINKLIB='XX' adds the LINKULE library XX to those available. XX is an additional library of linkules that is searched if a designated member is not found in the other libraries.

LINKULES is a library of operations that are FORTRAN subroutines. LINULES are usually efficient routines for carrying out specific mathematical operations. However, any FORTRAN subroutine can be joined to SPEAKEASY by an appropriate interface and become a LINKULE.
To obtain a listing of the present LINKULES, use the statements
T=LIBINDEX(KEPT)
TABULATE T

LIST MEMBER X OF LIBRARY Y FROM I TO J lists lines I through J OF X. X is the name of a member of library Y to be listed. The words MEMBER, LIBRARY, FROM, and TO are keywords. If TO is omitted, the listing begins at line I and goes to the end of the member. If FROM is also omitted, the entire member is listed. LIBRARY Y specifies the library to be searched for the member. The default library is PROCLIB. If the member is not specified, the library index is listed.
LIST specifies that the program is to be listed with the output. If not specified, the default for batch jobs is usually LIST. In the interactive operation, NOLIST is default.
%LIST in the edit mode lists the edit file. The entire file is listed unless statement numbers follow the word LIST.
%LIST N in the edit mode lists line N of the edit file.
%LIST N,M in the edit mode lists lines N through M.

LISTHEAD(N,XX,YY) lists N lines of member XX of library YY. N specifies the maximum number of lines of XX to be listed. XX is the name of the member to be listed. YY is the name of the library to be searched for member XX. If YY is not specified, PROCLIB is default.

LISTMEMBER(XX,YY) lists member XX of library YY.
XX is the name of the member to be listed.
YY is the name of the library.
If YY is omitted, the default is PROCLIB.
The entire member is listed.
An alternative form is LIST MEMBER XX OF LIBRARY YY.

LOADDATA(A,NAME) loads A with N values from the data file NAME.
N is the number of elements of A or the number of numbers in NAME, whichever is smaller.
LOADDATA is a restricted word.

LOC is a synonym for LOCS.
LOC(X) gives the indices of the nonzero (true) elements of X.
X is a 1-dimenional object. LOC is a logical function and the argumert X is often an array with a relational operator. The result is usually used as a structured index to produce a new array.

LOCMAX(X) specifies the location of the maximum element of X.
X must be a 1-dimensional array.

LOCMIN(X) specifies the location of the minimum element of X.
X must be a 1-dimensional array.

LOCS(X) gives the indices of the nonzero (true) elements of X.
X is a 1-dimensional object. LOCS is a logical function and the argument is often an array with a relational operator. The result is usually used as a structured index to produce a new array.
LOC is a synonym for LOCS.

LOG(X) returns the natural logarithm of X.
LOG(X) defines an object with the same structure as X but with elements equal to the natural logarithm of the corresponding elements of X.
LOG(X) is not defined if any element in X is 0.

LOGGAMMA(X) defines the natural logarithm of the gamma function of X.
LOGGAMMA(X) defines an object with the same structure as X but with elements that are the natural logarithm of the gamma function of the corresponding elements of X.
The elements Xi of X must be real and satisfy the condition 0<Xi<4E60.

LOWERTRI(X) returns the lower triangular part of X.
The result is a 1-dimensional object of the same family as X with elements corresponding to the lower triangular elements and the diagonal elements of X.
X must be a 2-dimensional square object.

.LT. is the relational operator "less than".
The periods must be on both sides of the operator.

MARGINS enables the user to contol the margins of the output.
MARGINS(N,M) specifies the left and right limits of output.
MARGINS(N,M) means that the output is restricted to columns starting at N and ending with M.
MARGINS(M) is equivalent to MARGINS(1,M).
MARGINS(0,M) is equal to MARGINS(1,M) without control functions.
MARGINS(0,M) eliminates all carriage control characters that are used to control the vertical spacing and to position output on the top of a new page.

MAT is a synonym for MATRIX.
MAT(N,M:) defines an N-by-M matrix.
If no additional arguments are given, the matrix has all elements set to zero.
MAT(N,M:I,J,...,K) defines an N-by-M matrix and presets the elements. The elements are set row by row by use of the values I,J,...,K or the elements of I,J,...,K if they are structured objects. If a complex element is encountered, then a complex matrix is defined. If all the elements are not preset by the element list, the unspecified elements are set to zero.

MATRIX(N,M:) defines an N-by-M matrix.
If no additional arguments are present, the matrix has all elements set to zero.
A shortened form is MAT.
MATRIX(N,M:I,J,...,K) defines an N-by-M matrix with preset elements. The elements are set row by row by use of the values I,J,...,K or the elements of I,J,...,K if they are structured objects. If a complex element is encountered, then a complex matrix is defined. If all the elements are not specified by the element list, the unspecified elements are set to zero.

MAX(X) specifies the value of the maximum element in X.

MAXOF(A,B,...,C) returns the largest element of A,B,...,C. The result is a scalar that is the maximum element in any of the objects in the calling argument list. The arguments must be real objects or numbers and can be of any class.

MAXOFCOL(X) returns the largest element in each column of X. X is a 2-dimensional array or a matrix. The result is a 1-dimensional object of the same family as X with elements that are the largest element of each column of X.

MAXOFROW(X) returns the largest element in each row of X. X is a 2-dimensional array or a matrix. The result is a 1-dimensional object of the same family as X with elements that are the largest element of each row of X.

MELD(I,J,...,K) gives an odometer ordering of elements of I,J,...,K. The arguments I,J,...,K must be 1-dimensional objects. MELD alters the structure of I,J,...,K so that every element in each object of the argument list is associated with every element of every other object in the list.

After the statement MELD(I,J,...,K), any function of I,J,...,K can be written in a straightforward manner.

For a further explanation and examples, refer to the SPEAKEASY-3 manual or the linkule document.

MFAM(X) defines a member of the matrix/vector family. The result has the same structure as X. If X is a 1-dimensional array or a vector, the result is a vector. If X is a 2-dimensional array or a matrix, the result is a matrix.

MFAM and VFAM are synonyms.

MIN(X) specifies the value of the minimum element in X.

MINOF(A,B,...,C) returns the smallest element in A,B,...,C. The result is a scalar that is the minimum element in any of the objects in the calling argument list. The arguments must be real objects or numbers and can be of any class.

MINOFCOL(X) returns the smallest element in each column of X. X is a 2-dimensional array or a matrix. The result is a 1-dimensional object of the same family as X with elements that are the smallest element of each column of X.

MINOFROW(X) returns the smallest element in each row of X. X is a 2-dimensional array or a matrix. The result is a 1-dimensional object of the same family as X with elements that are the smallest element of each row of X.

%MOVE N,M,K(I) moves statements N through M to K,K+I,K+2I,...
%MOVE N,M,K moves statements N through M to K,K+1,K+2,...
%MOVE N moves statement N to the last position.
%MOVE N,M moves statements N through M to the last positions.
%MOVE is a function available in the EDIT mode.

MYDOCS is the library name for private documents. MYDOCS is the library name for private documents other than the private HELP documents that are attached to the SPEAKEASY processor.

MYHELP is the library name for private HELP documents. MYHELP is the library name for private HELP documents that are attached to the SPEAKEASY processor.

MYLINKS is the library name for private linkules.
MYLINKS is the library name for private linkules to be attached to the SPEAKEASY processor. It is of the same form as the system library but contains personal routines. If MYLINKS is an attached library, it will be searched for a given member prior to the search of the LINKULE library.

NAMES prints the names of all currently defined SPEAKEASY objects. The command NAMES is particularly useful in the interactive mode since it provides the user with a means of obtaining a listing of the names of variables that he has previously defined.

.NE. is the relational operator "not equal to".
The periods must appear on both sides of the operator.

NEUMANN(NU,X) returns cylindrical Bessel fns. of the second kind. NU and X are real and nonnegative. There is a SPEAKEASY restiction that (X+NU)<400.
For further details of accuracy and the method used for the calculation refer to the linkule document.

NEWGRAPH causes the next graph to be drawn on a new area of paper. NEWGRAPH may be used in conjunction with ADDGRAPH to completely avoid the use of the statement GRAPH.
NEWGRAPH is available only in version GRAPHEZ.

NEWPAGE causes the printer to start a new page.
The output following the control statement NEWPAGE is printed on the top of the next page.
NEWPAGE is a restricted word.

NOCOLS(X) defines a scalar which is the number of columns in X.

NOECHO suppresses the echoing of input data.
This command causes just the results to be printed out. This statement may occur at any place in the program.
NOECHO is default for the interactive operation; ECHO is default for batch jobs.

NOELS(X) defines a scalar equal to the number of elements in X. If X is undefined, the result is zero.
A synonym is LENGTH.

NOLIST indicates that the program is not to be printed with the output. If not specified, NOLIST is default for the interactive operation; LIST is usually default for batch jobs.

NOROOTS(F) gives the number of roots of the function F. The number of times the function F changes sign (goes through zero) is found.
A synonym is NOZEROS.

NOROWS(X) defines a scalar equal to the number of rows in X.

.NOT. is the logical operator "not".
The periods must be on both sides of the operator.

NOZEROS(F) gives the number of zeros of the function F. The number of times the function F changes sign (goes through zero) is found.
A synonym is NOROOTS.

NUMBERS refers to numerical data.
All numerical data are specified in decimal form. The lack of a decimal point implies that it belongs just to the right of the last digit.
NUMBERS can be real, imaginary, or complex.
A terminal I implies that the previous number was imaginary. Examples of imaginary and complex numbers are: 21.6I; 6.1+5.2I
Very large or very small numbers may be expressed by ending the number in E followed by the power of 10 associated with that number. For example, 1.052E+2 means 105.2. If a number is imaginary, the I comes before the E, eg., 6.2IE-3.

OBJECT(I,J,...,K) generates an inplace name.
Each of the arguments I,J,...,K can be either a literal quantity or a nonnegative number. For literals, the expression itself is used. For numbers, the integer part of the number is used as a literal. The various arguments are then joined together to compose the name.
The expression OBJECT(I,J,...,K) may occur anywhere in a SPEAKEASY statement. It is particularly usefull combined with the word HENCEFORTH.

OMITCLASS suppresses the printing of the class.
If OMITCLASS is not specified, the class of the resulting structured objects is printed out when they are printed.
The statement OMITCLASS suppresses the printing of the class.

ONERROR(X,Y) specifies the action to be taken after an error.
X may be DUMP or NODUMP.
The options DUMP or NODUMP specify whether all defined data is to be printed or not.
Y may be MANUAL or CONTINUE.
MANUAL means that computation is continued in the manual mode.
CONTINUE means that the error does not affect the path of execution.
The default options are DUMP,MANUAL.

.OR. is the logical operator "or".
The periods must be on both sides of the operator.

ORDERED(X) gives the elements of X in increasing order. ORDERED(X) defines an object with the same structure as X but whose elements are the elements of X arranged in order of increasing magnitude.
A synonym is RANKED.

ORDERER(X) gives the indices of the ordered elements of X. ORDERER(X) gives the indices of the elements of X arranged in order of increasing magnitude. X must be a 1-dimensional object.
A synonym is RANKER.

PLOTSYMB(N,M) specifies the symbols used and frequency of points.
N is an integer which determines the frequency with which each symbol is plotted. (1 means every point, 2 means every other point, 3 means every third, etc.) A negative value for N indicates that only the symbol should appear. A positive value means that a line should join successive points.
M is an integer 0 through 12 that designates one of 13 different symbols to be used in plotting data. The integers and their corresponding symbols are:
0 is a square.
1 is a circle.
2 is a triangle.
3 is a plus (+).
4 is an X.
5 is a diamond.
6 is an arrow pointing upward.
7 is an X with a bar on top.
8 is a Z.
9 is a Y.
10 is an X with a circle.
11 is an asterisk (*).
12 is an X with a bar on top and bottom.
PLOTSYMB is used for graphical output.

PRINT(A,B,C,...,Z) specifies that A,B,C,...,Z are to be printed. The user may control the form of output by using the control words MARGINS, SIGNIFICANCE, AUTOTAB, TABULATE, and COLWIDTH.
PRINT is a restricted word.

PRINTCLASS prints the class of structured objects. PRINTCLASS causes the class of structured objects to be printed when the structured objects are printed. PRINTCLASS is default. To turn it off, use OMITCLASS.

PROD(X) defines the product of the elements of X. The result is a scalar.

PRODCOLS(X) multiplies the elements in each column of X. PRODCOLS(X) defines a 2-dimensional object that is a row and is a member of the family of X. The element in each column is the product of the elements in the corresponding column of X.

PRODROWS(X) multiplies the elements in each row of X. PRODROWS(X) defines a 2-dimensional object that is a column and is a member of the family of X. The element in each row is the product of the elements in the corresponding row of X.

PROGRAM NAME gives a name to a program.
A program is a SPEAKEASY procedure. Naming a program enables the user to refer to that program by name to store it, to use it within another program, or to execute it.
A program is a defined object. If its name is identical to a previously defined object, its definition will replace that object.
PROGRAM is a restricted word.

PUNCH(F:X) punches the object X on cards in the format F. The PUNCH statement gives the user a means of transmitting information to other non-SPEAKEASY programs.
The format F is specified by a statement of the form F='(FORMAT)' where FORMAT is the standard 360-FORTRAN IV format excluding fixed-point form (integers). The F, D, and E formats are equally acceptable.
X is the array to be punched.

QUIT terminates execution of a SPEAKEASY program. When all input is processed in normal batch operation, execution automatically terminates. This word is therefore not normally used for such cases.

RANDOM(X) generates random numbers.
The result has the same class and structure as X but its elements are random numbers between 0.0 and 1.0.
RANDOM will produce the same numbers on successive calls unless the first use of it is preceded by the statement LOAD(RANDOM).

RANKED(X) gives the elements of X in increasing order. RANKED(X) defines an object with the same structure as X but whose elements are the elements of X in increasing order.
A synonym is ORDERED.

RANKER(X) gives the indices of the ordered elements of X. RANKER(X) gives the indices of the elements of X arranged in order of increasing magnitude. X must be a 1-dimensional object.
A synonym is ORDERER.

READ(F:X) reads data from cards punched in format F and puts them into X. The READ statement enables the user to read information that has been punched in some specialized format.
The format F is specified by a statement of the form F='(FORMAT)' where FORMAT is the standard 360-FORTRAN IV format excluding fixed-point form (integers).
X is the array (structured object) into which the data are put. It must exist and its size must be specified. The number of cards read is the number needed to fill the array.
READ is not available in all versions of SPEAKEASY.

REAL is a synonym for REALPART.
REAL(X) returns the real part of X.
REAL(X) defines an object with the same structure as X but the elements of the result are the real part of the corresponding elements of X.

REALPART(X) returns the real part of X.
REALPART(X) defines an object with the same structure as X but the elements of the result are the real part of the corresponding elements of X.
A synonym is REAL.

REAL4(X) is a function that returns a real4 number.
X may be real8, integer4, or integer2. The result is real4.

REAL8(X) is a function that returns a real8 number.
X may be real4, integer4, or integer2. The result is real8.

RECLASS(A:B,C,...,Z) alters the structure of B,C,...,Z to agree with A. B,C,...,Z are previously defined objects with the same number of elements as A.

RETURN returns execution to the program calling the stored program. The statement executed after the statement RETURN is the one following the EXECUTE statement that invoked the stored program. A RETURN statement is always implied before the END card of any program.
RETURN is restricted to the program mode. It is ignored if it is used in the manual mode.
RETURN is a restricted word.

ROOTS(F:X) finds the roots of the function F. The result is a 1-dimensional array. X is a 1-dimensional array of the same length as F. Roots defines a new object whose elements correspond to the zeros of the function F. The method of trapezoidal interpolation is used.
A synonym is ZEROS.

ROWARRAY(N:) defines a 2-dim. N-component array that is a row. If no further arguments are given, all N-components are set equal to zero.

ROWARRAY(:I,J,...,K) defines a 2-dim. row array with preset els. The components are preset by the element list I,J,...,K. If any argument of the element list is structured, then the elements of that structured object are used. If a complex element is encountered, then a complex 2-dimensional row array is defined.

ROWARRAY(N:I,J,...,K) defines a 2-dim. N-component row array. The components are preset by the element list I,J,...,K. If any argument of the element list is structured, then the elements of that structured object are used. If a complex element is encountered, then a complex 2-dimensional row array is defined. If all N-components are not preset by the element list, the unspecified components are set equal to zero.

ROWMAT(N:) defines a 1-by-N matrix which is a row. If no further arguments are given, all N elements are set equal to zero.

ROWMAT(:I,J,...,K) defines a row matrix and specifies the els. The element list I,J,...,K specifies the elements of the matrix. If an argument of the list is structured, the elements of that structured object are used. If one of the elements is complex, the matrix is complex.

ROWMAT(N:I,J,...,K) defines a 1-by-N row matrix and specifies the els. The elements are preset by the element list I,J,...,K. If an argument of the element list is structured, the elements of that structured object are used. If an element is complex, the matrix is complex. If all N elements are not specified by the element list, the unspecified elements are set equal to zero.

ROWMAX(X) specifies the row containing the maximum element of X.

ROWMIN(X) specifies the row containing the minimum element of X.

%RUN is equivalent to %COMPILE followed by the command EXECUTE. %RUN used in the EDIT mode returns the processor from the EDIT mode to the MANUAL mode, compiles the edited program, and then executes it.

SELECT(A,B,...,C:I) truncates or expands A,B,...,C using the index I. A,B,...,C are 1-dimensional objects. I is a structured index that acts as a control array. A,B,...,C are truncated or expanded so their final components are those indexed by the elements of the control array I.

For an example, refer to the SPEAKEASY-3 manual.

SETGAUSS(N,X,W,XLO,XHI) returns Gauss-Legendre coords. and weights.
SETGAUSS(N,X,W,XLO,XHI) defines the N-point Gauss-Legendre quadrature coordinates and weights with the coordinates in the interval XLO to XHI. The arguments XLO and XHI may be omitted. If XLO and XHI are omitted, the interval is -1 to +1.

N is the number of points required. N must satisfy the condition 0<N<51.

X is returned as a 1-dimensional array whose elements are the coordinates.

W is returned as a 1-dimensional array whose elements are the weights.

For further information about the method of calculation and the accuracy, refer to the LINKULE document.

SETINFINITY(VAL) specifies an upper limit to the numbers printed. Any number whose absolute value is greater than VAL is printed as INF for infinity. The default value is 1.E+30.

SETJACOBI(N,X,W,A,B,TX,TW) defines Gauss-Jacobi coords. and weights.
SETJACOBI(N,X,W,A,B,TX,TW) defines the Gauss-Jacobi quadrature coordinates and weights for the interval -1 to +1 and gives the user an idea of the precision with which the coordinates and weights are calculated. The coordinates are the roots of the Jacobi polynomial of degree N and order (A,B). A, B, TX, and TW are optional arguments and may be omitted. If A and B are omitted, the order of the polynomial is 0.

N is the number of points required. N must be an integer satisfying the condition 0<N<51.

X is returned as a 1-dimensional array whose elements are the roots of the Jacobi polynomial of degree N and order (A,B).

W is returned as a 1-dimensional array whose elements are the weights of the integration proceedure.

A and B are real numbers specifying the order of the polynomial. A must satisfy the condition A>-1. B must satisfy the condition B>1. The default value for A and B is 0.

TX is the theoretical sum of the coordinates.

TW is the theoretical sum of the weights.

For further information about the method of calculation and the accuracy, refer to the LINKULE document.

SETLAGUERRE(N,X,W,A) returns Gauss-Laguerre coords. and weights.
SETLAGUERRE(N,X,W,A) defines the Gauss-Laguerre quadrature integration points X and weights W appropriate for evaluating an integral from 0 to infinity. The integration points are the roots of the Laguerre polynomial of degree N and order A. A may be omitted. If A is omitted, the polynomial calculated is order 0.

N is the number of points required. N must be an integer and satisfy the condition 1<N<33.

X is returned as a 1-dimensional array whose elements are the roots of the Laguerre polynomial of order A and degree N.

W is returned as a 1-dimensional array with the weights of the integration proceedure.

A is the order of the polynomial. The default is 0. A must be a real number greater than -1. A value of A of the order 20 could cause overflows in the determination of the weights for large values of N.

SETLEGENDRE(N,X,W) returns Gauss-Legendre coord. and weights.
SETLEGENDRE(N,X,W) returns the coordinates X and weights W for the Nth order Gauss-Legendre quadrature formula.
N is the number of points required. N must satisfy one of the conditions 0<N<51, N=64, or N=96.
X is returned as a 1-dimensional array whose elements are the coordinates.
W is returned as a 1-dimensional array whose elements are the weights.
SETLEGENDRE is equivalent to SETGAUSS for 0<N<51. The only difference is that SETLEGENDRE includes N=64 and N=96 while SETGAUSS does not.

SETLIB(XX,YY) changes the name of library XX to YY.

SETNULL(VAL) specifies a lower limit to numbers printed. Any number whose absolute value is less than VAL is to be printed as 0. The default value is 1.E-30.

SETPLOT(X,Y,Z) specifies BOX, NOBOX; SCALES, NOSCALES; LINES, POINTS.
X is either BOX or NOBOX.
BOX means to draw a frame around the graph.
NOBOX specifies no frame.
Y is either SCALES or NOSCALES.
SCALES specifies that values at inch intervals are to be indicated. NOSCALES causes the indication of scales to be omitted.
Z is either LINES or POINTS.
LINES specifies that points are to be joined by lines leaving the points unmarked. POINTS specifies that the points are to be marked with crosses and not joined by lines.
The default options are BOX, SCALES, and LINES.
SETPLOT is available only in version GRAPHEZ.

SIGN(X) specifies whether X is positive or negative.
SIGN(X) defines an object with the same structure as X but with elements of the result equal to +1 for elements of X >0, equal to -1 for elements of X <0, and equal to 0 for elements of X equal to 0.
All the elements of X must be real.

SIGNIFICANCE(N) gives the number of significant figures to be printed. N is the number of significant figures desired. The default is 5 significant figures.

SIMEQ(A,B) solves a set of simultaneous linear equations. A is an N-by-N matrix. B is an N-by-N matrix of constants. The result, X has the same structure as B and is the matrix that satisfies the simultaneous linear equation A*X=B.

SIN(X) returns the sine of X.
SIN(X) defines an object with the same structure as X but the elements of the result are the sine of the corresponding elements of X.
Each element Xi of X must satisfy the condition
|Xi|<1.E15.

SINH(X) defines the hyperbolic sine of X.
SINH(X) defines an object with the same structure as X but with elements equal to the hyperbolic sine of the corresponding elements of X.
The elements Xi of X must be real and are restricted by the inequality
|Xi|<170.

SIZE=N,X specifies space for data and must be the first card.
SIZE=N,X specifies the amount of data space needed and its location.
N is the number of kilobytes of storage to be set aside for SPEAKEASY data. One kilobyte is approximately 120 user-defined numbers.
X may be MAIN or LCS. LCS means the data storage area may be placed in LCS. MAIN is default.
The card specifying the size is the first one in the SPEAKEASY deck.

SMAT is a synonym for SYMMAT.
SMAT(N:I,J,...,K) defines a symmetric N-by-N matrix.
The element list is used to fill the lower triangular part by rows. The portion above the diagonal is then filled by making the matrix symmetric. If any argument in the list defining the elements is structured, the elements of that structured object are used.

SOLINDE2(X:A,B,C:Y1,Y1P:YP) solves Y"+AY'+BY=C on grid X.
Y=SOLINDE2(X,A,B,C:Y1,Y1P:YP) is the typical usage of the function SOLINDE2. The colons may be replaced by commas. X, A, B, C, YP, and Y are all 1-dimensional arrays.
X is an equally-spaced grid of independent variables.
It may be an increasing or decreasing grid.
A,B, and C are coefficient functions evaluated on the grid X.
Y1 is the starting value of Y at X(1).
Y1P is the starting value of Y' at X(1).
Y is the solution on grid X and is returned as a functional value.
YP is the derivative of the solution on grid X. YP is an optional name and need not be the name of a defined object. If YP is omitted, the calculation proceeds unaltered but Y' is not saved.
Temporary restrictions are:
A and B must be named scalars with the value zero. Only homogeneous equations without first derivatives can be solved. An example is
Z=0; Y=SOLINDE2(X:Z,B,Z:Y1,Y1P:YP).
X must have an odd number of points. No internal checks are made.
The 4th order Runge-Kutta method is used.

CMV

SPACE(N) skips N lines.
SPACE is a restricted word.

SPHBES(L,X) returns the spherical Bessel fn. of the first kind. The result is j sub L of X (little j). L must be a nonnegative integer. Underflow will occur for excessively large L. X must be real and nonnegative. There is a SPEAKEASY restriction that (X+L)<400.
For information on the method of calculation, refer to the LINKULE document.

SPHBESN(L,X) returns the spherical Bessel fn. of the second kind. The result is n sub L of X (little n). L must be a nonnegative integer. Overflow will occur for excessively large L. X must be real and nonnegative. There is a SPEAKEASY restriction that (L+X)<400.
For information on the method of calculation, refer to the LINKULE document.

SQRT(X) defines the square root of X.
SQRT(X) defines an object with the same structure as X but with elements equal to the square root of the corresponding elements of X.
If the domain of the calculation is complex, complex roots can be obtained. Otherwise, the square root of a negative number leads to an error message.

SUM(X) sums the elements of X.
SUM(X) defines a scalar object equal to the sum of the elements of X.

SUMCOLS(X) sums the elements in each column of X.
SUMCOLS(X) defines a 2-dimensional object that is a row and is a member of the family of X. The element in each column is the sum of the elements in the corresponding column of X.

SUMROWS(X) sums the elements in each row of X.
SUMROWS(X) defines a 2-dimensional object that is a column and is a member of the family of X. The element in each row is the sum of all the elements in the corresponding row of X.

SUMSQ(X) sums the squares of the elements of X.
SUMSQ(X) defines a scalar that is the sum of the squares of all the elements of X.

SUMSQCOLS(X) sums the squares of the elements in each column.
SUMSQCOLS(X) defines a 2-dimensional object that is a row and is a member of the same family as X. The element in each column is the sum of the squares of elements in the corresponding columns of X.

SUMSQROWS(X) sums the squares of the elements in each row. SUMSQROWS(X) defines a 2-dimensional object that is a column and is a member of the family of X. The element in each row is the sum of the squares of the elements in the corresponding row of X.

SYMBOLS designate mathematical operations or special cards. The symbols and a brief description of their meanings are:
+ is the addition operator.
- is the subtraction operator.
* is the multiplication operator.
For matrices, * implies matrix multiplication.
For vectors, * implies take the inner (dot) product.
For arrays, * is element-by-element multiplication.
If A and B are 1-dim. arrays, C=A*B is Ci=Ai*Bi, where i implies the ith element.
/ is the division operator.
For matrices, A/B implies matrix multiplication of the matrix A times the inverse of matrix B.
For vectors, / is not defined.
For arrays, division is defined element-by-element.
** raises to a power.
** is not defined for matrices.
For vectors, ** means to take the outer (cross) product.
For arrays, the operation is done element-by-element.
= means replace the object on the left by the expression on the right.
: separates the statement label from the statement or separates arguments in some words.
; separates successive statements on a single card.
& designates a continuation of the previous card.
$ sets off a comment. The comment is written between two $.
There is an implied $ at the end of a card that has a single $ on it.

SYMMAT(N:I,J,...,K) defines a symmetric N-by-N matrix. The element list is used to fill the lower triangular part by rows. The portion above the diagonal is then filled by making the matrix symmetric. If any argument in the element list is structured, the elements of that structured object are used.
A shortened form is SMAT.

TABULATE(A,B,...,C) prints 1-dim. objects A,B,...,C in tabular form. Each column is labeled by the name of the corresponding object.
The user may control the tabulation by the specifications MARGINS, COLWIDTH, and SIGNIFICANCE. All tables may be made uniform by using AUTOTAB.
Only 1-dimensional objects may be tabulated.

TAN(X) returns the tangent of X.
TAN(X) defines an object with the same structure as X but the elements of the result are the tangents of the corresponding elements of X.
Each element Xi of X must satisfy the condition |Xi|<1.E15.

TIME gives the time from which one starts.
TIME gives the total time elapsed since the start of the computer run.
TIME(0) gives the elapsed time since the start of a computer run. This form can be used in SPEAKEASY statements.

TOTALINT(F:X) defines the definite integral of F over the array X. X is a 1-dimensional array, (X1,X2,X3,...,XN). The result is a scalar equal to the definite integral from X1 to XN. The integration is done by the trapezoidal rule.
A shortened form is TOTINT.

TOTINT is a synonym for TOTALINT.
TOTINT(F:X) defines the definite integral of F over the array X. X is a 1-dimensional array (X1,X2,X3,...,XN). The result is a scalar equal to the definite integral from X1 to XN. The integration is done by the trapezoidal rule.

TRACE(X) gives the trace of the matrix X.
The trace is the sum of the diagonal elements.

TRANSP is a synonym for TRANSPOSE.
TRANSP(X) defines the transpose of X.
TRANSP(X) defines an object that is of the same class as X and is the transpose of the object X.

TRANSPOSE(X) defines the transpose of X.
TRANSPOSE(X) defines an object that is of the same class as X and is the transpose of the object X.
A shortened form is TRANSP.

UMAT is a synonym for UNITMAT.
UMAT(N) defines an N-by-N unit matrix.
All elements are zero except the elements along the diagonal, which have the value 1.

UNITMAT(N) defines an N-by-N unit matrix.
All elements are zero except the elements along the diagonal, which have the value 1.
A shortened form is UMAT.

UPPERTRI(X) returns the upper triangular part of X.
X must be a square 2-dimensional object. The result is a 1-dimensional object that is of the same family as X and whose elements are the upper triangular elements including those along the diagonal of X.

USE MEMBER X OF LIBRARY Y causes the deck X to be used as input. If the library reference is omitted, PROCLIB is assumed. If the library MYPROCS is attached, it will be searched for the member X before the library PROCLIB is searched.
This provides the user with a simple means of supplying commonly used constants or SPEAKEASY programs to the processor.
USE is a restricted word.

USEMEMBER(XX,YY) causes input of the member XX of library YY.
XX is the name of the member desired.
YY is the name of the library to be searched for member XX.
If the library reference is omitted, PROCLIB is assumed.
If the library MYPROCS is attached, it will be searched for member XX before PROCLIB is searched.
This provides the user with a means of supplying commonly used constants or SPEAKEASY programs to the processor.
An alternative form is USE MEMBER XX OF LIBRARY YY.

VARIABLE(I,J) defines a 1-dim. array of 101 points from I to J. The increment is chosen by the computer so that the points are equally spaced. Both points I and J are included.
A synonym is GRID(I,J).
VARIABLE(I,J,K) defines the 1-dim. array I,I+K,I+2K,...,J. The last element in the array is J. If the increment K is not such that I+NK=J, then a new increment K' is chosen by the computer so that the last element is J and the elements are equally spaced.
VARIABLE(I,J,K) is equivalent to GRID(I,J,K) only when K is such that I+NK=J.

VEC is a synonym for VECTOR.
VEC(N:) defines a vector with N components.
If there are no additional arguments, the values of the components are set equal to zero.
VEC(:I,J,...,K) defines a vector and specifies the components. The components are preset by the element list I, J,...,K. If any argument of the element list is structured, then the elements of that structured object are used. If a complex element is encountered, then a complex vector is defined.
VEC(N:I,J,...,K) defines an N-component vector with preset els. The components are specified by the element list I,J,...,K. If any argument of the element list is structured, then the elements of that structured object are used. If a complex element is encountered, then a complex vector is defined. If all N components are not specified, the unspecified components are set equal to zero.

VECTOR(N:) defines an N-component vector.
If no further arguments are given, all the components are set equal to zero.
A vector is a 1-dimensional member of the matrix/vector family.
A shortened form is VEC.
VECTOR(:I,J,...,K) defines a vector by presetting the components. The components are specified by the element list I,J,,...,K. If any argument in the element list is structured, then the elements of that structured object are used. If a complex element is encountered, a complex vector is defined.
VECTOR(N:I,J,...,K) defines an N-component vector with preset els. The components are preset by the element list I,J,,...,K. If any argument in the element list is structured, then the elements of that structured object are used. If a complex element is encountered, a complex vector is defined. If all N components are not preset by the element list, the unspecified components are set to zero.

VERSIONS refers to the various versions of the SPEAKEASY processor. These versions differ in their space requirements and in their efficiency of operation. The present versions are:
STANDARD
BABY
GRAPHEZ
CONSOLE
TSO
For further detail about these versions, refer to the SPEAKEASY-3 manual.

VFAM(X) defines a member of the matrix/vector family. The resultant has the same structure as X. If X is a 1-dimensional array or a vector, the result is a vector. If X is a 2-dimensional array or a matrix, the result is a matrix.

VLABEL='ANY MESSAGE' labels the vertical scale. The desired label is enclosed between apostrophes.
VLABEL is available only in version GRAPHEZ.

VOCABULARY generates a list of all currently defined words. The list is the actual list used by the processor in deciphering the user's SPEAKEASY program. The printout in response to the word VOCABULARY will change whenever a new word is added to the language.

VSCALE=(BOTTOM,TOP) specifies the vertical limits of a graph. BOTTOM refers to the lower limit and TOP the upper limit of the vertical scale. Default values of the limits on the vertical scale are 0 and 8. The scale is labeled at inch marks.
VSCALE is used for versions with graphical output.

VSIZE=Y specifies the vertical size of the graph. Y is in inches. The default is 8 inches. Y cannot exceed 10.
VSIZE is used for versions with graphical output.

WHERE is a synonym for WHEREVER.
WHERE(X) is a conditional statement for array operations. X may be a structured object. The associated statement Y must be an equation and the class of the result must be the same as that of the test expression X. The statement Y is executed only if the test expression X is true (nonzero).
WHERE is a retricted word.

WHEREVER(X) Y is a conditional statement for array operations. X may be a structured object. The associated statement Y must be an equation and the class of the result must be the same as that of the test expression X. The statement Y is executed only if the test expression X is true (nonzero).
WHEREVER is a restricted word.
A shortened form is WHERE.

WHOLE(X) returns the integer part of X.
WHOLE(X) defines an object with the same structure as X but the elements of the result are the integer part of the corresponding elements of X.
A synonym is INTPART.

WRITE(F:X) prints the defined object X in the format F. The WRITE statement enables the user to print information in a specific format.
The format F is specified by a statement of the form F='(FORMAT)' where FORMAT is the standard 360-FORTRAN IV format excluding fixed-point form (integers).
X is the defined object to be printed in the specific format.
WRITE is not available in all versions of SPEAKEASY.

ZEROS(F:X) finds the zeros of the function F. The result is a 1-dimensional array. X is a 1-dimensional array of the same length as F. ZEROS defines a new object whose elements correspond to the zeros of the function F. The method of trapezoidal interpolation is used.
A synonym is ROOTS.

APPENDIX

HELP documents are written as follows. Each word in the HELP documents is written in a specific format for retrieval of various levels of information. The first line of each document may be obtained by itself. Consequently, it is a one-line definition of the word. The information in each document is structured in levels so that each paragraph goes into greater detail. Eventually, the printing of a document will be halted after each paragraph so that the user will have the option of stopping or requesting further information. If further information is requested, the next paragraph will be printed.

To facilitate obtaining output on various devices, the length of each line is limited to 72 characters. For uniformity, a new paragraph is indented two spaces. Thus a line that begins a new paragraph is limited to 70 characters.

The first line of each document consists of (1) the word being defined (in capital letters), (2) the arguments of the word (in capital letters), and (3) a concise definition of the word. Since the first line is the beginning of a paragraph, it is indented two spaces so 70 characters are left for the word, argument list, and definition. Ideally this should be a complete sentence. Sometimes abbreviations must be used in order to convey the meaning of the word in so little space.

In addition to the restriction that the rest of the paragraph and the following paragraphs have lines of 72 characters or less, there is the convention that the word being defined and any of the arguments appear in capital letters throughout the document. If a word has more than one calling sequence, it is listed with each calling sequence as a new paragraph. The first line of this new paragraph begins with the word being defined and follows the same format as the first line of the document.

The last line of the document should have the initials of the contributor in columns 68—72. The contributor assumes responsibility for the validity of operation of the word described.

The documents created in this way are placed into a staging data set for validation before introducing them into the system library. However, the newly defined documents are available to SPEAKEASY users at Argonne.

HELP documents are written in upper and lower case. Consequently, words may be put in only on devices that have both upper and lower case, e.g., on the IBM-2741 terminal. The command TMODE LOWER gives lower case letters in the RESCUE EDIT mode.

The definition of each word must be preceded by a card of the form

/TO NEWWORD

The / is in column 1. NEWWORD is the first 8 characters of the word being put into the HELP-document library. This card is in capital letters.

An example follows.

```
//JFHELP JOB (F88888,1,1,1),CLASS=C,REGION=85K
//      EXEC SOS
//LIBOUT DD DSN=SPEAK.NEW.HELP,DISP=OLD
/TO MATRIX
  MATRIX(N,M:) defines an N-by-M matrix.
If no additional arguments are present, the matrix has all
elements set to zero.
  A shortened form is MAT.
  MATRIX(N,M:I,J,...,K) defines an N-by-M matrix with preset elements.
The elements are set row by row by use of the values I,J,...,K or
the elements of I,J,...,K if they are structured objects.  If
a complex element is encountered, then a complex matrix is
defined.  If all the elements are not specified by the element
list, the unspecified elements are set to zero.
/TO SUMSQCOL
  SUMSQCOLS(X) sums the squares of the elements in each column.
SUMSQCOLS(X) defines a 2-dimensional object that is a row
and is a member of the same family as X.  The element in each
column is the sum of the squares of elements in the corres-
ponding columns of X.
```