

**CONVEX FORTRAN Master Index**  
Document No. 720-000033-200

---

---

First Edition  
October 1988

*Sharon Lammie*

**CONVEX Computer Corporation**  
Richardson, Texas

---

xobaf

*CONVEX FORTRAN Master Index*

First Edition

© 1988 CONVEX Computer Corporation  
All rights reserved.

This document is copyrighted. This document may not, in whole or part, be copied, duplicated, reproduced, translated, stored electronically, or reduced to machine-readable form without prior written consent from CONVEX Computer Corporation.

Although the material contained herein has been carefully reviewed, CONVEX Computer Corporation (CONVEX) does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE PROGRAM DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS PROGRAM. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

CONVEX and the CONVEX logo ("C") are registered trademarks of CONVEX Computer Corporation.

COVUE is a trademark of CONVEX Computer Corporation.

Cray is a registered trademark of Cray Research, Inc.

Sun FORTRAN is a trademark of Sun Microsystems, Inc.

UNIX is a trademark of AT&T Bell Laboratories.

VAX and VMS are trademarks of Digital Equipment Corporation.

Printed in the United States of America

# Index

\$ descriptor lrm-8-16  
: descriptor lrm-8-17

## A

absolute value ug-C-6  
ACCEPT statement lrm-7-1, lrm-7-11  
access directly ug-2-5  
ACCESS keyword lrm-7-18  
access modes ug-2-5  
access sequentially ug-2-5  
accessing files lrm-7-3  
actual arguments lrm-9-1  
*adb* debugger ug-7-5  
adjustable arrays lrm-9-2  
algebraic simplification ug-4-14  
alternate return arguments lrm-9-5, lrm-9-9  
ANSI standard formatting lrm-1-4  
apostrophe descriptor lrm-8-5  
arc cosine ug-C-5  
arc cosine (degree) ug-C-5  
arc sine ug-C-5  
arc sine (degree) ug-C-5  
arc tangent ug-C-5  
arc tangent (degree) ug-C-5  
arc tangent, two arguments ug-C-5  
arc tangent, two arguments (degree) ug-C-6  
argument packets ug-5-1, ug-5-6  
argument pointer ug-5-1  
argument-passing mechanisms ug-5-2  
arguments lrm-9-8, lrm-9-9  
arguments, alternate return lrm-9-9  
arguments, character lrm-9-3  
arguments, dummy lrm-9-9  
arithmetic expression, data type lrm-2-9  
arithmetic expression, data types lrm-2-8  
arithmetic expressions lrm-2-8  
arithmetic IF statement lrm-6-3  
arithmetic operators lrm-2-8  
array arguments lrm-9-2  
array declaration lrm-2-7  
array storage lrm-2-8  
array subscripts lrm-2-7  
array table ug-1-10  
arrays lrm-2-7  
ASCII character set lrm-E-1  
assembly-language debugger ug-7-5  
ASSIGN statement lrm-5-3  
assigned GOTO statement lrm-6-2  
assignment statement lrm-5-1  
assignment substitution ug-4-12  
ASSOCIATEVARIABLE keyword lrm-7-20  
assumed-length character argument lrm-3-4  
assumed-size arrays lrm-9-3  
auxiliary input/output statements lrm-7-18  
auxiliary I/O operations ug-C-17

## B

B descriptors lrm-8-12  
BACKSPACE statement lrm-7-28  
BEGIN\_TASKS directive lrm-C-4  
bitwise AND ug-C-10  
bitwise circular shift ug-C-12

bitwise clear ug-C-12  
bitwise complement ug-C-10  
bitwise extract ug-C-11  
bitwise OR ug-C-10  
bitwise shift ug-C-10  
bitwise test ug-C-11  
bitwise XOR ug-C-10  
blank common storage lrm-3-1  
BLANK keyword lrm-7-20, lrm-F-2  
BLOCK DATA statement, definition  
lrm-10-1  
BLOCK DATA statement in subprograms  
lrm-9-9  
block IF statement lrm-6-4  
blockdata subprogram lrm-10-1  
BLOCKSIZE keyword lrm-7-20  
BN descriptor lrm-8-12  
branch optimization ug-4-19  
built-in functions lrm-9-6  
built-in functions %REF and %VAL lrm-9-6  
BYTE lrm-2-1, ug-A-1  
BZ descriptor lrm-8-12

## C

C interface ug-1-11  
CALL statement lrm-9-10  
calling conventions ug-5-1, ug-C-1  
calling utility routines ug-6-1  
carriage-control characters lrm-8-24  
CARRIAGECONTROL keyword lrm-7-20  
*-cfc* option lrm-G-1  
CHAR function ug-3-4  
character arguments lrm-9-3  
character constants lrm-2-6, ug-3-1  
character conversions lrm-5-1  
character data ug-3-1  
character descriptor A lrm-8-4  
character equivalence lrm-3-5  
character expressions lrm-2-10  
CHARACTER FUNCTION statement  
lrm-9-9  
character I/O ug-3-3  
character library functions ug-3-4  
character relationals ug-C-12  
character representation ug-A-4  
character strings, concatenating ug-3-3  
character substrings lrm-2-11, ug-3-2  
CHARACTER type-declaration statements  
lrm-3-3  
character variables, declaring ug-3-2  
character-per-column formatting lrm-1-3  
characters, carriage-control lrm-8-24  
character-valued function ug-5-1  
CLOSE statement lrm-7-25  
code motion ug-4-10  
coding statements lrm-1-1  
colon descriptor lrm-8-17  
comma field separator lrm-8-18  
comment line lrm-1-2  
common block storage lrm-3-1  
common blocks lrm-3-6  
COMMON statement lrm-3-1

COMMON statements lrm-2-6  
 common subexpression elimination ug-4-13  
 compiler directives lrm-1-4, lrm-C-1  
 compiler features ug-1-1  
 compiler messages ug-1-8, ug-B-1  
 compiler options lrm-B-1, ug-1-2  
 compiling FORTRAN-66 programs lrm-F-1  
 compiling programs lrm-B-1, ug-1-2  
 COMPLEX lrm-2-1  
 complex conjugate ug-C-8  
 complex descriptor lrm-8-4  
 complex programmed operators ug-C-14  
 complex representation ug-A-4  
 COMPLEX\*16 lrm-2-1  
 COMPLEX\*16 constants lrm-2-4  
 COMPLEX\*8 lrm-2-1, lrm-2-9  
 COMPLEX\*8 constants lrm-2-4  
 computed GOTO statement lrm-6-2  
 concatenation lrm-2-10  
 conditional induction variables ug-4-6  
 constant expressions lrm-2-11  
 constant propagation and folding ug-4-7,  
 ug-4-13  
 constants lrm-2-3  
 contact, reporting problems lrm-K-1, ug-D-1  
 continuation indicator lrm-1-3  
 continuation line lrm-1-3  
 CONTINUE statement lrm-6-9  
 control statement lrm-6-1  
 conversion of data types lrm-2-2  
 CONVEX FORTRAN ug-1-1  
 CONVEX math library ug-C-1  
 copy propagation ug-4-8  
 COVUEshell lrm-1-7, lrm-7-18  
 Cray compatibility lrm-G-1  
 cross-reference generator ug-7-1  
 csd debugger ug-7-4

## D

D descriptor lrm-8-9  
 D indicator lrm-1-4  
 data item runtime, examples ug-C-15  
 data representation ug-A-1  
 data representations ug-5-5  
 DATA statement lrm-4-1  
 DATA statement form lrm-4-1  
 DATA statement, implied-DO lrm-4-2  
 data types lrm-2-1  
 data types, arithmetic expressions lrm-2-8  
 data types, conversion lrm-2-2  
 data types, equivalenced lrm-3-4  
 data-type length specifiers lrm-3-3  
 date ug-6-3  
 -dc option lrm-1-4  
 dead-code elimination ug-4-8  
 debug statements lrm-1-4  
 debugger, assembly-language ug-7-5  
 debugger, symbolic ug-7-4  
 debugging programs ug-7-1  
 DECODE statement lrm-7-16  
 default descriptor values lrm-8-17

DEFAULTFILE keyword lrm-7-21  
 descriptors lrm-8-3  
 diagnostic messages ug-1-7  
 dimension declarator lrm-2-7  
 DIMENSION statement lrm-3-4  
 direct access lrm-7-4, ug-2-5  
 direct-access file ug-2-6  
 direct-access WRITE statements lrm-7-13  
 directives, compiler lrm-1-4, lrm-C-1  
 DISPOSE keyword lrm-7-21  
 DO loops, extended range lrm-6-8  
 DO statement lrm-6-8, lrm-6-9  
 DO statements lrm-6-6  
 DO WHILE statement lrm-6-8, lrm-6-9  
 dollar sign descriptor lrm-8-16  
 DOUBLE PRECISION lrm-2-1  
 double-Precision constants lrm-2-3  
 dummy argument lrm-9-1  
 dummy argument, NAMELIST statement  
 lrm-3-8  
 dummy arguments lrm-9-1, lrm-9-8, lrm-9-9  
 dummy arguments, arrays lrm-9-2  
 dynamic loop selection ug-4-18

## E

E descriptor lrm-8-9  
 edit descriptors lrm-8-3  
 ENCODE statement lrm-7-15  
 ENCODE statement, example lrm-7-16  
 END DO statement lrm-6-9  
 END specifier ug-8-1  
 END statement lrm-6-10  
 END statement in subprograms lrm-9-9  
 ENDFILE record lrm-7-2  
 ENDFILE statement lrm-7-28  
 end-of-file specifier lrm-7-7  
 END\_TASKS directive lrm-C-4  
 entry points, I/O list element transmission  
 ug-C-16  
 entry points, I/O list initialization ug-C-16  
 entry points, scalar intrinsics ug-C-2  
 entry points, vector intrinsics ug-C-2  
 ENTRY statement lrm-9-11  
 EQUIVALENCE statement lrm-3-4  
 EQUIVALENCE statements lrm-2-6  
 equivalencing arrays lrm-3-5  
 ERR keyword lrm-7-21  
 ERR specifier ug-8-1  
 error messages ug-1-7  
 error reporting lrm-K-1, ug-D-1  
 error specifier lrm-7-7  
 error utility ug-B-1  
 error-processing utilities ug-8-4  
 errors, runtime ug-8-1  
 errsns ug-6-4  
 errtrap utility ug-8-5  
 examples ug-5-7  
 exception, runtime ug-8-1  
 exceptions ug-8-3  
 executable program lrm-1-1  
 executing programs ug-1-7

*exit* ug-6-4  
 exponentiation programmed operators  
     ug-C-13  
 expressions lrm-2-8  
 expressions, constant lrm-2-11  
 external file type ug-2-5  
 external READ statements lrm-7-9  
 EXTERNAL statement lrm-3-9

## F

F descriptor lrm-8-8  
*fc* command line lrm-B-1, ug-1-2  
 field descriptors lrm-8-3  
 field separators, external lrm-8-18  
*.fil* files ug-4-15  
 FILE keyword lrm-7-21  
 file positioning lrm-8-3  
 file type ug-2-5  
 file-naming conventions ug-1-1  
 file-positioning statements lrm-7-28  
 files lrm-7-2  
 files, accessing lrm-7-3  
 files, FORTRAN source ug-1-1  
 FIND statement lrm-7-17  
 fix-to-float conversion ug-C-7  
 floating-point data representation ug-A-2  
 floating-point, IEEE ug-A-3  
 floating-point, native ug-A-3  
 floating-point representation, IEEE lrm-B-3,  
     ug-1-1, ug-1-4  
 float-to-fix conversion ug-C-6  
 force directives lrm-C-5  
 FORCE\_PARALLEL directive lrm-C-5  
 FORCE\_VECTOR directive lrm-C-5  
 FORM keyword lrm-7-22  
 format code separators lrm-F-3  
 FORMAT control lrm-8-2  
 format specifications lrm-8-1  
 format specifier lrm-7-6  
 FORMAT statement lrm-8-1  
 formats, variable lrm-8-19  
 formatted I/O ug-2-4  
 formatted records lrm-7-2  
 formatting, list-directed lrm-8-20  
 FORTRAN argument packets ug-5-1  
 FORTRAN character set lrm-1-1, lrm-E-1  
 FORTRAN intrinsic library ug-C-1  
 FORTRAN I/O library ug-C-15  
 FORTRAN-66 compatibility lrm-F-1  
 FORTRAN-77 formatting lrm-1-4  
*fpp* lrm-J-1  
 FUNCTION statement lrm-9-9  
 function subprograms lrm-9-9  
 function-naming convention ug-C-1  
 functions lrm-9-5

## G

G descriptor lrm-8-10  
 generic and intrinsic functions, table lrm-A-1  
*gerror* utility ug-8-7  
 global optimization ug-4-7

GOTO statement lrm-6-1

## H

H descriptor lrm-8-5  
 hexadecimal constants lrm-2-4, lrm-2-5  
 hoisting ug-4-20  
 Hollerith constants lrm-2-5  
 Hollerith representation ug-A-5  
 hyperbolic cosine ug-C-6  
 hyperbolic sine ug-C-6  
 hyperbolic tangent ug-C-6

## I

I descriptor lrm-8-6  
 ICHAR function ug-3-4  
*idate* ug-6-4  
 IEEE floating-point ug-A-3  
 IEEE floating-point representation lrm-B-3,  
     ug-1-1, ug-1-4  
 IEEE/native conversions ug-C-12  
*ierrno* utility ug-8-7  
 IF statements lrm-6-3  
 IF THEN statement lrm-6-4  
 imaginary part of complex ug-C-8  
 IMPLICIT NONE statement lrm-3-2  
 IMPLICIT statement lrm-3-2  
 implied-DO list lrm-7-5  
 #include statement lrm-1-6  
 INCLUDE statement lrm-1-6  
 INDEX function ug-3-5  
 Inf message lrm-8-9, lrm-8-10, lrm-8-11  
 Inf operand ug-A-3  
 initial line lrm-1-3  
 inline substitution ug-4-14  
 inlining, how to use ug-4-15  
 inlining, restrictions on ug-4-17  
 inlining, when to use ug-4-15  
 input, list-directed lrm-8-20  
 input/output ug-2-1  
 input/output lists lrm-7-4  
 input/output statements lrm-7-1  
 input/output statements, auxiliary lrm-7-18  
 INQUIRE statement lrm-7-25  
 instruction scheduling ug-4-19  
 integer constants lrm-2-3  
 integer conversion ug-C-7  
 integer descriptor lrm-8-4  
 integer part of real ug-C-8  
 integer representation ug-A-1  
 INTEGER\*1 lrm-2-1  
 INTEGER\*2 lrm-2-1  
 INTEGER\*4 lrm-2-1  
 INTEGER\*8 lrm-2-1  
 internal direct-access WRITE statements  
     lrm-7-14  
 internal file type ug-2-5  
 internal files lrm-7-2, ug-2-6  
 internal READ statements lrm-7-11  
 intrinsic functions lrm-9-5  
 intrinsic runtimes ug-C-3  
 INTRINSIC statement lrm-3-9

invariant computation ug-4-10  
 invoking the compiler lrm-B-1, ug-1-2  
 I/O error processing ug-8-1  
 I/O forms ug-2-4  
 I/O list element transmission ug-C-16  
 I/O list initialization ug-C-16  
 I/O list termination ug-C-17  
 I/O operation ug-C-15  
 I/O runtime naming convention ug-C-16  
 I/O statement format lrm-7-4  
 I/O statements lrm-7-1  
 IOSTAT keyword lrm-7-22  
 IOSTAT specifier ug-8-2  
 -is option ug-4-16

## K

keywords, OPEN statement lrm-7-18

## L

L descriptor lrm-8-5  
*ld* ug-1-6  
 LEN function ug-3-4  
 lexical comparison functions ug-3-5  
 libraries, runtime ug-1-7  
 limits, system lrm-D-1  
 list-directed character input lrm-8-20  
 list-directed complex input lrm-8-20  
 list-directed formatting lrm-8-20  
 list-directed input lrm-8-20  
 list-directed I/O ug-2-4  
 list-directed, null value lrm-8-21  
 list-directed output lrm-8-23  
 list-directed, slashes lrm-8-21  
 LNBLNK function ug-3-4  
 loader, UNIX ug-1-6  
 loading programs ug-1-6  
 %LOC ug-5-4  
 %LOC built-in function lrm-9-7  
 local optimization ug-4-12  
 logical constants lrm-2-6  
 logical descriptor lrm-8-4  
 logical elements lrm-2-10  
 logical entities lrm-2-9  
 logical expressions lrm-2-10  
 logical IF statement lrm-6-3  
 logical names ug-2-1  
 logical operator XOR lrm-2-10  
 logical records ug-2-6  
 logical representation ug-A-1  
 LOGICAL\*1 lrm-2-1  
 LOGICAL\*2 lrm-2-1  
 LOGICAL\*4 lrm-2-1  
 LOGICAL\*8 lrm-2-1  
*longjmp* utility ug-8-4  
 loop distribution ug-4-2  
 loop interchange ug-4-3  
 loop replication ug-4-17  
 loop table ug-1-8  
 loop unrolling ug-4-17

## M

machine-dependent optimization ug-4-18  
 main program lrm-1-1, lrm-3-7  
 matching paired vector references ug-4-20  
 maximum ug-C-8  
 MAXREC keyword lrm-7-22  
 MAX\_TRIPS directive lrm-C-1  
 messages ug-1-7, ug-B-1  
 minimum ug-C-8  
 multiple statements lrm-1-4  
*mvbits* ug-6-5

## N

NAME keyword lrm-7-21  
 namelist input lrm-8-21  
 namelist specifier lrm-7-8  
 NAMELIST statement lrm-3-8  
 namelist-directed formatting lrm-8-21  
 namelist-directed I/O ug-2-6  
 namelist-directed output lrm-8-23  
 NaN message lrm-8-9, lrm-8-10, lrm-8-11  
 NaN operand ug-A-3  
 native floating-point ug-A-3  
 nearest integer ug-C-7  
 nested block IF statement lrm-6-6  
 nested DO loops lrm-6-7  
 NEXT\_TASK directive lrm-C-4  
 NML keyword lrm-7-8  
 non-FORTRAN-to-FORTRAN calling  
     sequence ug-5-4  
 nonrepeatable descriptors lrm-8-2  
 NO\_RECURRENCE directive lrm-C-1,  
     ug-4-5  
 NO\_SIDE\_EFFECTS directive lrm-C-2  
 NOSPANBLOCKS keyword lrm-7-23  
 numeric type-declaration statements lrm-3-3

## O

O descriptor lrm-8-7  
 octal constants lrm-2-4, lrm-2-5  
 OPEN statement lrm-7-18, ug-2-2  
 OPEN statement keywords lrm-F-2  
 operator precedence lrm-2-8  
 optimization ug-4-1  
 optimization directives lrm-C-3  
 optimization report ug-1-7, ug-1-8  
 options, compiler lrm-B-1, ug-1-2  
 OPTIONS statement lrm-1-5

## P

P descriptor lrm-8-13  
 packets, argument ug-5-1  
 paired vector references ug-4-20  
 parallel processing ug-4-6  
 parallelization ug-4-6  
 parallelization directives lrm-C-3  
 PARAMETER statement lrm-3-6  
 PARAMETER statement, alternate lrm-3-7  
 PAUSE statement lrm-6-10  
*perror* utility ug-8-7

*pmd* utility ug-7-2  
 pointer, argument ug-5-1  
 positive difference ug-C-9  
 post-mortem dump ug-7-2  
 precedence, operator lrm-2-8  
 preconnection of units ug-2-1  
 preprocessor lrm-J-1  
 PRINT statement lrm-7-1, lrm-7-14  
 problem reporting ug-D-1  
 procedure names ug-5-5  
 procedures as dummy arguments lrm-9-4  
 program elements lrm-1-1  
 program, executable lrm-1-1  
 program interfaces ug-1-11  
 program, main lrm-1-1, lrm-3-7  
 PROGRAM statement lrm-1-1, lrm-3-7  
 PROGRAM statement in subprograms lrm-9-9  
 program unit lrm-1-1  
 PSTRIIP directive lrm-C-7

## Q

Q descriptor lrm-8-16

## R

R descriptor lrm-8-14  
*ran* ug-6-4  
 READ statement lrm-7-1, lrm-7-8  
 READ statement, direct lrm-7-10  
 READ statement, external lrm-7-9, lrm-7-10  
 READ statement, internal lrm-7-11  
 READ statement, sequential lrm-7-9  
 READONLY keyword lrm-7-23  
 REAL lrm-2-1  
 real constants lrm-2-3  
 real data representation ug-A-2  
 real descriptor lrm-8-4  
 real part of complex ug-C-8  
 REAL\*4 lrm-2-1, ug-C-8  
 REAL\*4 Constants lrm-2-3  
 REAL\*8 lrm-2-1, lrm-2-9  
 REAL\*8 constants lrm-2-3  
 REAL\*8 product of Real\*4's ug-C-9  
 REAL\*8 to REAL\*4 conversion ug-C-8  
 RECL keyword lrm-7-23  
 record specifier lrm-7-6  
 records lrm-7-1  
 records, logical ug-2-6  
 RECORDSIZE keyword lrm-7-23  
 RECORDTYPE keyword lrm-7-23  
 recurrence ug-4-4  
 recurrences, array references ug-4-4  
 reductions ug-4-5  
 redundant-assignment elimination ug-4-9, ug-4-12  
 redundant-subexpression elimination ug-4-9  
 redundant-use elimination ug-4-12  
 %REF ug-5-3  
 %REF built-in function lrm-9-6  
 register allocation ug-4-19  
 relational expressions lrm-2-9

remainder ug-C-9  
 repeatable descriptors lrm-8-2, lrm-8-3  
 report, optimization ug-1-8  
 reporting problems lrm-K-1, ug-D-1  
 RETURN, alternate lrm-9-10  
 RETURN statement lrm-6-10, lrm-9-11, lrm-9-12  
 RETURN statement, definition lrm-9-12  
 RETURN statement in subprograms lrm-9-9  
 return values ug-5-6  
 REWIND statement lrm-7-28  
 RINDEX function ug-3-5  
 Rop message lrm-8-9, lrm-8-10, lrm-8-11  
 Rop (reserved operand) ug-A-3  
 ROW-WISE directive lrm-C-7  
 runtime data items ug-C-14  
 runtime error messages ug-1-10, ug-B-2  
 runtime errors and exceptions ug-8-1  
 runtime formats lrm-8-19  
 runtime interface ug-1-11  
 runtime libraries ug-1-7, ug-C-1  
 runtime messages ug-B-1  
 runtime prefixes ug-C-1  
 runtime stack ug-5-4  
 runtime utilities ug-6-1

## S

S descriptor lrm-8-13  
 SAVE statement lrm-3-9  
 SCALAR directive lrm-C-5  
 scalar truncation ug-C-1  
 scalar/vector intrinsic example ug-C-3  
 scale factor lrm-8-13  
*secnds* ug-6-4  
 SELECT directive lrm-C-8  
 semantic differences with vectorization ug-4-3  
 sequential access lrm-7-3, ug-2-5  
 sequential READ statements lrm-7-9  
 sequential WRITE statement, unformatted lrm-7-13  
 sequential-access file ug-2-6  
 sequential-access WRITE statements lrm-7-12  
*setjmp* utility ug-8-4  
*-sfc* option lrm-I-1  
 signal handling examples ug-8-8  
*signal* utility ug-8-5  
 signals and exceptions ug-8-2  
 slash descriptor lrm-8-17  
 source files ug-1-1  
 SP descriptor lrm-8-13  
 span-dependent instructions ug-4-19  
 specification statements lrm-3-1  
 specifiers lrm-7-5  
 SS descriptor lrm-8-13  
 stack, runtime ug-5-4  
 statement field lrm-1-4  
 statement function reference lrm-9-8  
 statement functions lrm-9-7  
 statement label lrm-1-3  
 statement label assignment lrm-5-3

statements, executable lrm-1-2  
 statements, multiple lrm-1-4  
 statements, nonexecutable lrm-1-2  
 STATUS keyword lrm-7-24, lrm-F-3  
 status specifier lrm-7-7  
 STOP statement lrm-6-10  
 storage, array lrm-2-8  
 strength reduction ug-4-11  
 strength reduction and the code generator  
   ug-4-20  
 string index ug-C-12  
 string length ug-C-12  
 string-manipulation programmed operators  
   ug-C-14  
 strip mining ug-4-2  
 strip mining, parallel lrm-C-6  
 strip-mine directives lrm-C-6  
 SU descriptor lrm-8-14  
 subprogram lrm-1-1  
 subprogram calling conventions ug-5-1  
 subprograms lrm-9-1  
 SUBROUTINE statement in subprograms  
   lrm-9-9  
 subroutine subprograms lrm-9-10  
 Sun FORTRAN compatibility lrm-I-1  
 symbolic Names lrm-2-1  
 SYNCH\_PARALLEL directive lrm-C-8  
 system errors ug-B-2  
 system limits lrm-D-1  
 system utilities ug-6-1  
*system* utility ug-6-3

## T

T descriptors lrm-8-15  
 tab character lrm-1-4  
 tab-key formatting lrm-1-4  
 table, array ug-1-10  
 tangent ug-C-5  
 tasking directives lrm-C-3, lrm-C-4  
*time* ug-6-4  
 TL descriptor lrm-8-15  
 TR descriptor lrm-8-15  
*traceback* utility ug-8-6  
 transfer of sign ug-C-10  
*traper* utility ug-8-6  
 tree-height reduction ug-4-20  
 trouble reports lrm-K-1, ug-D-1  
 TYPE keyword lrm-7-24  
 TYPE statement lrm-7-1, lrm-7-14  
 type-declaration statements lrm-3-2

## U

unconditional GOTO statement lrm-6-1  
 unformatted I/O ug-2-4  
 unformatted records lrm-7-2  
 UNIT keyword lrm-7-24  
 unit specifier lrm-7-6  
 units lrm-7-3  
 units, input/output ug-2-1  
 UNIX utilities ug-6-1  
 UNROLL directive lrm-C-9

utilities ug-6-1  
 utility routines, how to call ug-6-1

## V

%VAL ug-5-3  
 %VAL built-in function lrm-9-6  
 variable formats lrm-8-19  
 variables lrm-2-6  
 VAX FORTRAN compatibility lrm-H-1  
 VAX FORTRAN records lrm-H-3  
 VAX-11 FORTRAN system utilities ug-6-3  
 vector mask programmed operators: ug-C-14  
 vectorization ug-4-1  
 vectorization directives lrm-C-3  
 vectorization, nested DO loops ug-4-2  
 vectorization restriction ug-4-3  
 vectorizer limitations ug-4-4  
*vers* command lrm-K-1, ug-D-1  
 version of software, how to find lrm-K-1,  
   ug-D-1  
*-vfc* option lrm-H-1  
 VMS FORTRAN compatibility lrm-H-1  
 VSTRIP directive lrm-C-7

## W

*which* lrm-K-1, ug-D-1  
 WRITE statement lrm-7-1, lrm-7-12,  
   lrm-8-23  
 WRITE statement, list-directed output for-  
   mats lrm-8-23  
 WRITE statements, direct access lrm-7-13  
 WRITE statements, internal lrm-7-14

## X

X descriptor lrm-8-14  
 X format edit descriptor lrm-F-3

## Z

Z descriptor lrm-8-7