

## PASCAL QUICK REFERENCE

---

### OPERATORS (IN ORDER OF PRECEDENCE, EXTEND IN BOLD)

|             |  |
|-------------|--|
| Unary       | <b>NOT</b> <b>ADR</b> <b>ADS</b>   |
| Multiplying | <b>*</b> / <b>DIV</b> <b>MOD</b> <b>AND</b>  |
| Adding      | <b>+</b> - <b>OR</b> <b>XOR</b>  |
| Relational  | <b>=</b> <b>&lt;&gt;</b> <b>&lt;=</b> <b>&gt;=</b> <b>&lt;</b> <b>&gt;</b> <b>IN</b> |

---

### SET OPERATORS (APPLY DIFFERENTLY TO SETS)

|                               |                                     |
|-------------------------------|-------------------------------------|
| <b>+</b> Set union            | <b>-</b> Set difference             |
| <b>*</b> Set intersection     | <b>IN</b> Test set membership       |
| <b>=</b> Test set equality    | <b>&lt;&gt;</b> Test set inequality |
| <b>&lt;=</b> and <b>&gt;=</b> | Test subset and superset            |
| <b>&lt;</b> and <b>&gt;</b>   | Test proper subset and superset     |

---

### STATEMENT EXAMPLES (EXTEND LEVEL UNDERLINED)

|                      |  |
|----------------------|--|
| <b>Assignment</b>    | A := B + C;  |
| <b>Procedure</b>     | ProcedureName(P1, P2);   |
| <b><u>BREAK</u></b>  | BREAK Label2;  |
| or                   | IF A = B THEN BREAK Label1;  |
| <b>CASE</b>          | CASE expression OF<br>Case label: Statement;<br>{Case label can be a range}<br>Case label: Statement;<br>OTHERWISE {optional}<br>Statement<br>END; |
| <b><u>CYCLE</u></b>  | CYCLE Label2;  |
| <b>FOR</b>           | FOR I := 1 TO 10 DO  |
| or                   | FOR J := 10 DOWNTO 1 DO  |
| <b>GOTO</b>          | GOTO Label2  |
| <b>IF</b>            | IF I := Expression THEN Statement<br>ELSE I := Expression  |
| <b>REPEAT</b>        | REPEAT<br>Statement;<br>Statement<br>UNTIL Expression;   |
| <b><u>RETURN</u></b> | RETURN   |
| <b>WHILE</b>         | WHILE Expression DO Statement  |
| <b>WITH</b>          | WITH Recordname DO Statement(FieldA, FieldB)   |

---

### CONSTANTS

|                      |                  |
|----------------------|------------------|
| MAXINT = 32767       | MAXWORD = 65535  |
| MAXINT4 = 2147483647 | SINT = -127..127 |

---

---

## PREDECLARED TYPES

### TYPE

BOOLEAN = (FALSE,TRUE);  
INTEGER2 = INTEGER;  
STRING = SUPER PACKED ARRAY [1..\*] OF CHAR;  
SINT = -127..127;  
LSTRING = SUPER PACKED ARRAY [ 0..\*] OF CHAR;  
BYTE = WRD(0)..255;  
ADAPQQQ = ARRAY [WRD(0)..32765] OF BYTE;  
ADRMEM = ADR OF ADAPQQQ;  
ADSMEM = ADS OF ADAPQQQ;  
INTEGER1 = SINT;

---

## VARIABLE DECLARATION EXAMPLES SHOWING USE OF TYPES

VAR NUMBER : INTEGER;  
LARGENUMBER : WORD;  
HUGENUMBER : INTEGER4;  
SINGLE\_PRECISION\_DECIMAL : REAL4;  
DOUBLE\_PRECISION\_DECIMAL : REAL8;  
ALSO\_LEGAL : REAL; {Set precision with \$REAL}  
CHARACTER : CHAR;  
CONDITION : BOOLEAN;  
ENUMERATED\_COLORS : (RED, ORANGE, YELLOW,  
GREEN, BLUE, VIOLET);  
SCORE : 0..10; {subrange type}  
MY\_ARRAY : ARRAY [1..10] OF INTEGER;  
SUPER : SUPER ARRAY [1..\*] OF REAL;  
INFORMATION : RECORD  
    FIRST\_ITEM : LSTRING(100);  
    SECOND\_ITEM : INTEGER;  
    THIRD : ARRAY [1..9] OF SINT  
    END  
VARIANT\_RECORD : RECORD  
    X,Y : REAL;  
    S : SHAPE;  
    CASE S OF  
        SQUARE : (SIZE, ANGLE: REAL);  
        CIRCLE : (DIAMETER: REAL)  
    END;  
HUES : SET OF ENUMERATED\_COLORS;  
PALETTE : FILE OF ENUMERATED\_COLORS;  
PTR : WORD;  
AD : ADS OF WORD;  
AR : ADR OF INTEGER;

---

---

**RESERVED WORDS (\*INDICATES AN ATTRIBUTE)**

|            |                |            |
|------------|----------------|------------|
| ADR        | FUNCTION       | PUBLIC *   |
| ADS        | GOTO           | PURE *     |
| AND        | IF             | READONLY * |
| ARRAY      | IMPLEMENTATION | RECORD     |
| BEGIN      | IN             | REPEAT     |
| BREAK      | INTERFACE      | RETURN     |
| CASE       | INTERRUPT *    | SET        |
| CONST      | LABEL          | STATIC *   |
| CONSTS     | MOD            | THEN       |
| CYCLE      | MODULE         | TO         |
| DIV        | NIL            | TYPE       |
| DO         | NOT            | UNIT       |
| DOWNT0     | OF             | UNTIL      |
| ELSE       | OR             | USES       |
| END        | ORIGIN         | VALUE      |
| EXTERN *   | OTHERWISE      | VAR        |
| EXTERNAL * | PACKED         | VARS       |
| FILE       | PORT           | WHILE      |
| FOR        | PROCEDURE      | WITH       |
| FORWARD *  | PROGRAM        | XOR        |

---

**METACOMMANDS** ENCLOSE IN COMMENTS, E.G. {\$DEBUG+}  
(Number, +, or - is default setting)

|  |  |
|--|--|
| <b>\$BRAVE+</b>  | Sends error messages and warnings to the video display.                |
| <b>\$DEBUG-</b>  | Turns on or off all the debug checking.                                |
| <b>\$ENTRY-</b>  | Generates procedure entry/exit calls for Pascal error routines.        |
| <b>\$ERRORS:25</b>   | Sets the number of errors allowed per page. Default is 25.             |
| <b>\$GOTO-</b>   | Flags GOTO statements as "considered harmful."                         |
| <b>\$IF &lt;constant&gt;</b><br><b>\$THEN&lt;text&gt;</b><br><b>\$END</b>                                | Allows conditional compilation of source.<br><br>{or}                  |
| <b>\$IF &lt;constant&gt;</b><br><b>\$THEN&lt;text1&gt;</b><br><b>\$ELSE&lt;text2&gt;</b><br><b>\$END</b> |  |
| <b>\$INCLUDE:'&lt;filename&gt;'</b>  | Switches compilation from current source file to source file named.    |
| <b>\$INCONST:&lt;identifier&gt;</b>  | Allows interactive setting of constant values at compile time.         |
| <b>\$INDEXCK+</b>  | Checks for array index values in range, including super array indexes. |

---

---

|                                |   |
|--------------------------------|---|
| <b>\$INITCK-</b>               | Checks for uninitialized values.  |
| <b>\$INTEGER:2</b>             | Sets the length of the INTEGER type.  |
| <b>\$LINE-</b>                 | Generates line number calls for the Pascal error routines.  |
| <b>\$LINESIZE:131</b>          | Sets width of listing.  |
| <b>\$LIST+</b>                 | Turns on or off source listing.   |
| <b>\$MATHCK-</b>               | Checks for mathematical errors such as overflow and division by zero.                                     |
| <b>\$MESSAGE:&lt;text&gt;</b>  | Allows the display of a message on the video display to indicate which version of a program is compiling. |
| <b>\$NILCK+</b>                | Checks for bad pointer values.  |
| <b>\$OCODE+</b>                | Turns on disassembled object code listing.  |
| <b>\$PAGE+</b>                 | Sets page number for next page.   |
| <b>\$PAGE:&lt;n&gt;</b>        | Sets page number for next page.   |
| <b>\$PAGEIF:&lt;n&gt;</b>      | Skips to next page if less than a specified number of lines are left on the page.                         |
| <b>\$PAGESIZE:55</b>           | Sets length of listing in lines.  |
| <b>\$POP</b>                   | Restores saved value of all metacommands.   |
| <b>\$PUSH</b>                  | Saves current value of all metacommands.  |
| <b>\$RANGECK-</b>              | Checks for subrange validity.   |
| <b>\$REAL:4</b>                | Sets the length of the REAL type.   |
| <b>\$ROM-</b>                  | Gives a warning on static initialization.   |
| <b>\$RUNTIME-</b>              | Determines context of run-time errors.  |
| <b>\$SIMPLE</b>                | Disables global optimizations.  |
| <b>\$SIZE</b>                  | Minimizes size of code generated.   |
| <b>\$SKIP:&lt;n&gt;</b>        | Skips specified number of lines or to end of page.  |
| <b>\$SPEED</b>                 | Minimizes execution time of code.   |
| <b>\$STACKCK-</b>              | Checks for stack overflow at procedure or function entry.   |
| <b>\$SUBTITLE:&lt;text&gt;</b> | Sets page subtitle.   |
| <b>\$SYMTAB+</b>               | Sends symbol table to listing file.   |
| <b>\$TITLE:&lt;text&gt;</b>    | Sets page title.  |
| <b>\$WARN+</b>                 | Writes warning messages to listing file.  |

---

---

## INTRINSIC PREDECLARED PROCEDURES AND FUNCTIONS

PROCEDURE **ABORT** (CONST MESS: STRING; ERR1, ERR2: WORD);

FUNCTION **ABS** (X: NUMERIC): NUMERIC;

FUNCTION **ARCTAN** (X: REAL): REAL;

PROCEDURE **ASSIGN** (VAR F: FILE OF..; CONSTS N: STRING);

FUNCTION **BYLONG** (HI: WORD or INTEGER or INTEGER4; LO: WORD or INTEGER or INTEGER4);INTEGER4

FUNCTION **BYWORD** (PAR1, PAR2): WORD;

FUNCTION **CHR** (X: ORDINAL): CHAR;

PROCEDURE **CLOSE** (VAR F: FILE OF..);

PROCEDURE **CONCAT** (VARS D: LSTRING; CONSTS S: STRING);

PROCEDURE **COPYLST** (CONSTS S: STRING; VARS D:LSTRING);

PROCEDURE **COPYSTR** (CONSTS S: STRING; VARS D: STRING);

FUNCTION **COS** (X: NUMERIC): REAL;

FUNCTION **DECODE** (CONST LSTR: LSTRING, X: M: N): BOOLEAN;

PROCEDURE **DELETE** (VARS D: LSTRING; I, N: INTEGER);

PROCEDURE **DISCARD** (VAR F: FILE OF..);

PROCEDURE **DISPOSE** (VARS P: POINTER);

PROCEDURE **DISPOSE** (VARS P: POINTER; T1, T2,..TN: TAGS);

FUNCTION **ENCODE** (VAR LSTR: LSTRING, X:M:N): BOOLEAN;

FUNCTION **EOF**: BOOLEAN;

FUNCTION **EOF** (VAR F: FILE OF ..): BOOLEAN;

FUNCTION **EOLN**: BOOLEAN;

FUNCTION **EOLN** (VAR F: FILE OF ..): BOOLEAN;

PROCEDURE **EVAL** (EXPRESSION, EXPRESSION,..);

FUNCTION **EXP** (X: NUMERIC): REAL;

PROCEDURE **FILLC** (D: ADRMEM; N: WORD; C: CHAR);

PROCEDURE **FILLSC** (D: ADSMEM; N: WORD; C: CHAR);

FUNCTION **FLOAT** (X: INTEGER): REAL;

FUNCTION **FLOAT4** (X: INTEGER4): REAL;

PROCEDURE **GET** (VAR F: FILE OF..);

FUNCTION **HIBYTE** (I: INTEGER or WORD): BYTE;

FUNCTION **HIWORD** (I: INTEGER4): WORD;

PROCEDURE **INSERT** (CONSTS S: STRING; VARS D: LSTRING; I: INTEGER);

FUNCTION **LN** (X: REAL): REAL;

FUNCTION **LOBYTE** (I: INTEGER or WORD): BYTE;

FUNCTION **LOWER** (EXPRESSION): VALUE;

FUNCTION **LOWORD** (I: INTEGER4): WORD;

---

---

PROCEDURE **MOVEL** (S, D: ADRMEM; N: WORD);  
PROCEDURE **MOVER** (S, D: ADRMEM; N: WORD);  
PROCEDURE **MOVESL** (S, D: ADSMEM; :N: WORD);  
PROCEDURE **MOVESR** (S, D: ADSMEM; N: WORD);  
PROCEDURE **NEW** (VARS P: POINTER);  
PROCEDURE **NEW** (VARS P: POINTER; T1, T1..TN: TAGS);  
FUNCTION **ODD** (X: ORDINAL): BOOLEAN;  
FUNCTION **ORD** (X: VALUE): INTEGER;  
PROCEDURE **PACK** (CONSTS A: UNPACKED; I: INDEX;  
                  VARS Z: PACKED);  
PROCEDURE **PAGE**;  
PROCEDURE **PAGE** (VAR F: FILE OF ..);  
FUNCTION **POSITN** (CONSTS PAT: STRING; CONSTS S:  
                  STRING; I: INTEGER): INTEGER  
FUNCTION **PRED** (X: ORDINAL): ORDINAL;  
PROCEDURE **READ** (VAR F: FILE OF..; P1, P2,..PN);  
PROCEDURE **READFN** (VAR F: FILE OF..; P1, P2,..PN);  
PROCEDURE **READLN** (VAR F: FILE OF..; P1, P2..PN);  
PROCEDURE **READSET** (VAR F: FILE OF..; VAR L: LSTRING;  
                  CONST S: SETOFCHAR);  
PROCEDURE **RESET** (VAR F: FILE OF..);  
PROCEDURE **REWRITE** (F);  
FUNCTION **RESULT** (FUNCTION-IDENTIFIER): VALUE;  
FUNCTION **RETYPE** (TYPE-IDENT, EXPRESSION):TYPE-IDENT;  
FUNCTION **ROUND** (X: REAL): INTEGER;  
FUNCTION **ROUND4** (X: REAL): INTEGER4;  
FUNCTION **SCANEQ** (LEN: INTEGER; PART: CHAR;  
                  CONSTS S: STRING; I: INTEGER): INTEGER;  
FUNCTION **SCANNE** (LEN: INTEGER; PAT: CHAR; CONSTS S:  
                  STRING; I: INTEGER): INTEGER;  
PROCEDURE **SEEK** (VAR F: FILE OF..; N: INTEGER4);  
FUNCTION **SIN** (X: NUMERIC): REAL;  
FUNCTION **SIZEOF** (VARIABLE: WORD);  
FUNCTION **SIZEOF** (VARIABLE, TAG1, TAG2,..TAGN): WORD;  
FUNCTION **SQR** (X: NUMERIC): NUMERIC;  
FUNCTION **SQRT** (X): REAL  
FUNCTION **SUCC** (X: ORDINAL): ORDINAL;  
FUNCTION **TRUNC** (X: REAL): INTEGER;  
FUNCTION **TRUNC4** (X: REAL): INTEGER4;  
PROCEDURE **UNPACK** (CONSTS Z: PACKED;  
                  VARS A: UNPACKED; I: INDEX);  
FUNCTION **UPPER** (EXPRESSION): VALUE;  
FUNCTION **WRD** (X: VALUE): WORD;  
PROCEDURE **WRITE** (VAR F: FILE OF..; P1, P2, .. PN);  
PROCEDURE **WRITELN** (VAR F: FILE OF..; P1, P2, .. PN);

---

---

## OTHER PROCEDURES AND FUNCTIONS

FUNCTION **ACSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **ACDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **AISRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **AIDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **ALLHQQ** (SIZE: WORD): WORD; EXTERN;  
FUNCTION **ALLMQQ** (wants: WORD) :ADSMEM; EXTERN;  
FUNCTION **ANSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **ANDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **ASSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **ASSRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **ATSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **ATDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **A2SRQQ** (A, B: REAL4): REAL4; EXTERN;  
FUNCTION **A2DRQQ** (A, B: REAL8): REAL8; EXTERN;  
PROCEDURE **BEGOQQ**; EXTERN;  
PROCEDURE **BEGXQQ**; EXTERN;  
  
FUNCTION **CHSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **CHDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **CNSRQQ** (CONSTS A: REAL4): REAL4;  
FUNCTION **CNDRQQ** (CONSTS A: REAL8): REAL8;  
FUNCTION **DISMQQ** (block: ADSMEM); EXTERN;  
PROCEDURE **ENDOQQ**; EXTERN;  
PROCEDURE **ENDXQQ**; EXTERN;  
FUNCTION **EXSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **EXDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **FREET** (SIZE: WORD): WORD; EXTERN;  
FUNCTION **FREMQQ** (block: ADSMEM): WORD; EXTERN;  
FUNCTION **GETMQQ** (wants: WORD): ADSMEM; EXTERN;  
FUNCTION **GTUQQ** (LEN: WORD; LOC: ADSMEM):  
WORD; EXTERN;  
FUNCTION **LADDOK** (A, B: INTEGER4; VAR C: INTEGER4):  
BOOLEAN; EXTERN;  
FUNCTION **LDSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **LDDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
FUNCTION **LMULOK** (A, B: INTEGER4; VAR C: INTEGER4):  
BOOLEAN; EXTERN;  
FUNCTION **LNSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
FUNCTION **LNDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
PROCEDURE **MARKAS** (VAR HEAPMARK: INTEGER4); EXTERN;  
FUNCTION **LOCKED** (VARS SEMAPHORE: WORD): BOOLEAN;  
EXTERN;

---

---

FUNCTION **MDSRQQ** (CONSTS A, B: REAL4): REAL4;  
 EXTERN;  
 FUNCTION **MDDRQQ** (CONSTS A, B: REAL8): REAL8;  
 EXTERN;  
 FUNCTION **MEMAVL**: WORD; EXTERN;  
 FUNCTION **MNSRQQ** (CONSTS A, B: REAL4): REAL4;  
 EXTERN;  
 FUNCTION **MNDRQQ** (CONSTS A, B: REAL8): REAL8;  
 EXTERN;  
 FUNCTION **MXSRQQ** (CONSTS A, B: REAL4): REAL4;  
 EXTERN;  
 FUNCTION **MXDRQQ** (CONSTS A, B: REAL8): REAL8;  
 EXTERN;  
 FUNCTION **PISRQQ** (CONSTS A: REAL4; CONSTS B:  
 INTEGER4): REAL4; EXTERN;  
 FUNCTION **PIDRQQ** (CONSTS A: REAL8; CONSTS B:  
 INTEGER4): REAL8; EXTERN;  
 PROCEDURE **PLYUQQ**; EXTERN;  
 FUNCTION **PREALLOCHEAP** (VARS CBALLOC: WORD):  
 WORD; EXTERN;  
 FUNCTION **PREALLOCHEAP** (CPARA: WORD); WORD:  
 EXTERN;  
 FUNCTION **PRSRQQ** (A, B: REAL4): REAL4; EXTERN;  
 FUNCTION **PRDRQQ** (A, B: REAL8): REAL8; EXTERN;  
 PROCEDURE **PTYUQQ** (LEN: WORD; LOC: ADSMEM); EXTERN;  
 PROCEDURE **PUT** (VAR F: FILE OF..);  
 PROCEDURE **RELEAS** (VAR HEAPMARK: INTEGER4); EXTERN;  
 FUNCTION **SADDOK** (A, B: INTEGER; VAR C: INTEGER):  
 BOOLEAN; EXTERN;  
 FUNCTION **SHSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
 FUNCTION **SHDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
 FUNCTION **SMULOK** (A, B: INTEGER; VAR C: INTEGER):  
 BOOLEAN; EXTERN;  
 FUNCTION **SNSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
 FUNCTION **SNDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
 FUNCTION **SRSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
 FUNCTION **SRDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
 FUNCTION **THSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
 FUNCTION **THDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
 FUNCTION **TNSRQQ** (CONSTS A: REAL4): REAL4; EXTERN;  
 FUNCTION **TNDRQQ** (CONSTS A: REAL8): REAL8; EXTERN;  
 FUNCTION **UADDOK** (A, B: WORD; VAR C: WORD):  
 BOOLEAN; EXTERN;  
 FUNCTION **UMULOK** (A, B: WORD; VAR C: WORD):  
 BOOLEAN; EXTERN;  
 PROCEDURE **UNLOCK** (VARS SEMAPHORE: WORD); EXTERN;

---