ALGORITHM 524
MP, A Fortran Multiple-Precision Arithmetic Package [A1]

RICHARD P. BRENT
Australian National University

Key Words and Phrases: arithmetic, multiple precision, extended precision, floating point, elementary function evaluation, Euler's constant, gamma function, polyalgorithm, software package, Fortran, machine-independent software, special function evaluation, Bessel functions, exponential integral, logarithmic intergral, Bernoulli numbers, zeta function, portable software
CR Categories: 3.15, 4.49, 5.11, 5.12, 5.15, 5.19, 5.25
Language: Fortran

DESCRIPTION

The design of the package and the theoretical background for the algorithms used are given in [1]. Details of calling sequences, etc., are given in the comments included here and in [2].

REFERENCES

1. BRENT, R.P. A Fortran multiple-precision arithmetic package *ACM Trans. Math. Software* 4, 1 (March 1978), 57–70.
2. BRENT, R.P. MP users guide. Tech Rep. 54, Computer Centre, Australian National U., Canberra, Australia, Sept. 1976.

ALGORITHM

[Only that portion of the listing which gives the introductory comments and a small example program is printed here. The complete listing, together with a Users' Guide giving further details, is available from the ACM Algorithms Distribution Service (see inside back cover for order form).]

Received July 1975; revised October 1975, March 1976, and October 1976.

Author's address: Computer Centre, Australian National University, Box 4, Canberra, ACT 2600, Australia.

```
C ABCDEFGHIJKLMNOPQRSTUVWXYZ$Ø123456789+-*/=(),.                    MPØØØØ1Ø
C                                                                   MPØØØØ2Ø
C $$                    ******  COMMENTS  ******                   MPØØØØ3Ø
C                                                                   MPØØØØ4Ø
C DESCRIPTION OF MP (VERSION OF 17 FEBRUARY 1977)                   MPØØØ51
C ****************                                                  MPØØØØ6Ø
C                                                                   MPØØØØ7Ø
C MP IS A MULTIPLE-PRECISION ARITHMETIC PACKAGE.                    MPØØØØ8Ø
C IT IS ALMOST COMPLETELY MACHINE-INDEPENDENT, AND SHOULD           MPØØØØ9Ø
C RUN ON ANY MACHINE WITH AN ANSI STANDARD FORTRAN COMPILER,        MPØØØ1ØØ
C SUFFICIENT MEMORY, AND A WORDLENGTH OF AT LEAST 16 BITS.          MPØØØ11Ø
C SOME MODIFICATIONS WOULD BE NECESSARY FOR A WORDLENGTH            MPØØØ12Ø
C OF LESS THAN 16 BITS.                                             MPØØØ13Ø
C                                                                   MPØØØ14Ø
C MP WAS WRITTEN BY R. BRENT (COMPUTER CENTRE, AUSTRALIAN           MPØØØ15Ø
C NATIONAL UNIVERSITY), IN NOVEMBER 1973 (VERSION 1).               MPØØØ16Ø
C CORRESPONDENCE SHOULD BE SENT TO R. P. BRENT, COMPUTER            MPØØØ17Ø
C CENTRE, ANU, BOX 4, CANBERRA, ACT 26ØØ, AUSTRALIA.               MPØØØ18Ø
C                                                                   MPØØØ19Ø
C MP HAS BEEN TESTED ON A UNIVAC 11Ø8  (E LEVEL FORTRAN V),         MPØØØ2ØØ
C A UNIVAC 11ØØ/42 (E AND T LEVEL FORTRAN V, ASCII FORTRAN,         MPØØØ21Ø
C AND RALPH), A PDP 1Ø (FORTRAN 1Ø(/NOOPT) AND FORTRAN 4Ø),         MPØØØ22Ø
C AN IBM 36Ø/5Ø (FORTRAN G AND FORTRAN H, OPT = 2),                 MPØØØ23Ø
C AN IBM 36Ø/91 AND 37Ø/168 (FORTRAN H EXTENDED, OPT = 2),          MPØØØ24Ø
C A CYBER 76 (FTN 4.2, OPT = 1) AND A PDP 11/45 (DOS).              MPØØØ25Ø
C THESE MACHINES HAVE EFFECTIVE INTEGER WORDLENGTHS RANGING         MPØØØ26Ø
C FROM 16 TO 48 BITS.                                               MPØØØ27Ø
C                                                                   MPØØØ28Ø
C MP WORKS WITH NORMALIZED FLOATING-POINT NUMBERS.                  MPØØØ29Ø
C THE BASE (B) AND NUMBER OF DIGITS (T) ARE                         MPØØØ3ØØ
C ARBITRARY (SUBJECT TO SOME RESTRICTIONS GIVEN BELOW),             MPØØØ31Ø
C AND MAY BE VARIED DYNAMICALLY.                                    MPØØØ32Ø
C                                                                   MPØØØ33Ø
C T-DIGIT FLOATING-POINT NUMBERS ARE STORED IN INTEGER ARRAYS OF    MPØØØ34Ø
C DIMENSION T+2, WITH THE FOLLOWING CONVENTIONS -                   MPØØØ35Ø
C   WORD 1 = SIGN (Ø, -1 OR +1)                                     MPØØØ36Ø
C   WORD 2 = EXPONENT (TO BASE B)                                   MPØØØ37Ø
C   WORDS 3 TO T+2 = NORMALIZED FRACTION                            MPØØØ38Ø
C NOTE THAT WORDS 2 TO T+2 ARE UNDEFINED IF SIGN = Ø.               MPØØØ39Ø
C                                                                   MPØØØ4ØØ
C ARITHMETIC IS ROUNDED, AND FOUR GUARD DIGITS ARE USED             MPØØØ41Ø
C FOR ADDITION AND MULTIPLICATION, SO THE CORRECTLY ROUNDED         MPØØØ42Ø
C RESULT IS USUALLY PRODUCED.   DIVISION, SQRT ETC ARE DONE         MPØØØ43Ø
C BY NEWTONS METHOD, BUT GIVE THE EXACT RESULT IF IT CAN BE         MPØØØ44Ø
C REPRESENTED WITH T-2 DIGITS. OTHER ROUTINES (MPSIN, MPLN ETC)     MPØØØ45Ø
C USUALLY GIVE A RESULT Y = F(X) WHICH COULD BE OBTAINED            MPØØØ46Ø
C BY MAKING AN O(B**(1-T)) PERTURBATION IN X, EVALUATING F          MPØØØ47Ø
C EXACTLY, THEN MAKING AN O(B**(1-T)) PERTURBATION IN Y.            MPØØØ48Ø
C                                                                   MPØØØ49Ø
C EXPONENTS CAN LIE IN THE RANGE -M, ... , +M INCLUSIVE,            MPØØØ5ØØ
C WHERE M IS SET BY THE USER.                                       MPØØØ51Ø
C ON UNDERFLOW DURING AN ARITHMETIC OPERATION, THE RESULT           MPØØØ52Ø
C IS SET TO ZERO BY SUBROUTINE MPUNFL.                              MPØØØ53Ø
C ON OVERFLOW SUBROUTINE MPOVFL IS CALLED AND EXECUTION             MPØØØ54Ø
C IS TERMINATED WITH AN ERROR MESSAGE.                              MPØØØ55Ø
C ERROR MESSAGES ARE PRINTED ON LOGICAL UNIT LUN, WHERE LUN         MPØØØ56Ø
C IS SET BY THE USER, AND THEN EXECUTION IS TERMINATED              MPØØØ57Ø
C BY A CALL TO SUBROUTINE MPERR.  IT IS ASSUMED THAT LOGICAL        MPØØØ58Ø
C RECORDS OF UP TO 8Ø CHARACTERS MAY BE WRITTEN ON UNIT LUN.        MPØØØ59Ø
C A WORKING ARRAY OF SIZE MXR (SEE BELOW) MUST BE PROVIDED          MPØØØ6ØØ
C IN COMMON.                                                        MPØØØ61Ø
C                                                                   MPØØØ62Ø
C THE PARAMETERS B, T, M, LUN AND MXR ARE PASSED TO THE UTILITY     MPØØØ63Ø
```

```
C ROUTINES IN COMMON, TOGETHER WITH A WORKING ARRAY R WHICH              MP000640
C MUST BE SUFFICIENTLY LARGE (SEE BELOW).  MOST ROUTINES                 MP000650
C USE THE STATEMENTS -                                                   MP000660
C      COMMON B, T, M, LUN, MXR, R                                       MP000670
C      INTEGER B, T, R(1)                                                MP000680
C AND IT IS ASSUMED THAT R IS DIMENSIONED SUFFICIENTLY LARGE             MP000690
C IN THE CALLING PROGRAM, AND THAT MXR IS SET TO THE                     MP000700
C DIMENSION OF R IN THE CALLING PROGRAM.                                 MP000710
C WARNING - IT IS ASSUMED THAT THE COMPILER PASSES ADDRESSES OF          MP000720
C *******   ARRAYS USED AS ARGUMENTS IN SUBROUTINE CALLS (I.E., CALL     MP000730
C           BY REFERENCE), AND DOES NOT CHECK FOR ARRAY BOUNDS           MP000740
C           VIOLATIONS (EITHER FOR ARGUMENTS OR FOR ARRAYS IN COMMON).   MP000750
C           APART FROM THESE VIOLATIONS, MP IS WRITTEN ENTIRELY IN       MP000760
C           ANSI STANDARD FORTRAN (ANSI X3.9-1966).  THIS HAS BEEN       MP000770
C           CHECKED BY THE PFORT VERIFIER.                               MP000780
C                                                                        MP000790
C RESTRICTIONS -                                                         MP000800
C      B (THE BASE) MUST BE AT LEAST 2,                                  MP000810
C      T (NUMBER OF DIGITS) MUST BE AT LEAST 2,                          MP000820
C      M (EXPONENT RANGE) MUST BE GREATER THAN T AND LESS                MP000830
C        THAN 1/4 THE LARGEST MACHINE-REPRESENTABLE INTEGER,             MP000840
C      8*B**2-1  MUST BE NO GREATER THAN THE LARGEST MACHINE-            MP000850
C        REPRESENTABLE INTEGER, AND THE INTEGERS 0, 1, ... , B           MP000860
C        MUST BE EXACTLY REPRESENTABLE AS SINGLE-PRECISION FLOATING-     MP000870
C        POINT NUMBERS, AND                                              MP000880
C      B**(T-1) SHOULD BE AT LEAST 10**7.                                MP000890
C                                                                        MP000900
C B AND T MAY BE SET TO GIVE THE EQUIVALENT OF A SPECIFIED               MP000910
C NUMBER OF DECIMAL PLACES BY CALLING MPSET (SEE BELOW), OR MAY          MP000920
C BE SET DIRECTLY BY THE USER.  IF MPSET IS NOT CALLED, THE USER         MP000930
C MUST REMEMBER TO INITIALIZE M, LUN AMD MXR (SEE ABOVE) AS WELL         MP000940
C AS B AND T BEFORE CALLING ANY MP ROUTINES.                             MP000950
C                                                                        MP000960
C FOR EFFICIENCY CHOOSE B FAIRLY LARGE, SUBJECT TO THE RESTRICTIONS      MP000970
C GIVEN ABOVE.  FOR EXAMPLE, IF THE WORDLENGTH IS                        MP000980
C      48 BITS, COULD USE B = 419430 OR 1000000,                         MP000990
C      36 BITS, COULD USE B = 65536 OR 10000,                            MP001000
C      32 BITS, COULD USE B = 16384 OR 10000,                            MP001010
C      24 BITS, COULD USE B = 1024 OR 1000,                              MP001020
C      18 BITS, COULD USE B = 128 OR 100,                                MP001030
C      16 BITS, COULD USE B = 64 OR 10.                                  MP001040
C                                                                        MP001050
C AVOID MULTIPLICATION OR DIVISION BY MP NUMBERS, AS                     MP001060
C THESE TAKE O(T**2) OPERATIONS, WHEREAS MULT./DIV. BY                   MP001070
C INTEGERS TAKE O(T) OPERATIONS.                                         MP001080
C                                                                        MP001090
C MP NUMBERS USED AS ARGUMENTS OF SUBROUTINES NEED NOT BE                MP001100
C DISTINCT.  FOR EXAMPLE,                                                MP001110
C       CALL MPADD (X, Y, Y)  OR  CALL MPEXP (X, X) ARE OK.              MP001120
C HOWEVER, DISTINCT ARRAYS WHICH OVERLAP SHOULD NOT BE USED.             MP001130
C                                                                        MP001140
C FOR ADDITIONAL DETAILS SEE - A FORTRAN MULTIPLE-PRECISION              MP001150
C ARITHMETIC PACKAGE (BY R. P. BRENT), TO APPEAR IN ACM                  MP001160
C TRANSACTIONS ON MATHEMATICAL SOFTWARE (AVAILABLE AS A                  MP001170
C CARNEGIE-MELLON UNIV. COMPUTER SCIENCE DEPT. REPORT,                   MP001180
C PITTSBURGH, PENNSYLVANIA, MAY 1976) AND THE MP USERS GUIDE.            MP001190
C                                                                        MP001200
C SUMMARY OF MP ROUTINES                                                 MP001210
C **********************                                                MP001220
C                                                                        MP001230
C BASIC ARITHMETIC - MPADD, MPADDI, MPADDQ, MPDIV, MPDIVI,               MP001240
C                    MPMUL, MPMULI, MPMULQ, MPREC, MPSUB                 MP001250
C                                                                        MP001260
```

```
C POWERS AND ROOTS - MPPWR, MPPWR2, MPQPWR, MPROOT, MPSQRT                MPØØ127Ø
C                                                                         MPØØ128Ø
C ELEMENTARY FUNCTIONS - MPASIN, MPATAN, MPCOS, MPCOSH, MPEXP,            MPØØ129Ø
C                    MPLN, MPLNGS, MPLNI, MPSIN, MPSINH, MPTAN,           MPØØ13ØØ
C                    MPTANH                                               MPØØ131Ø
C                                                                         MPØØ132Ø
C SPECIAL FUNCTIONS - MPBESJ, MPDAW, MPEI, MPERF, MPERFC,                 MPØØ133Ø
C                    MPGAM, MPGAMQ, MPLI, MPLNGM                          MPØØ134Ø
C                                                                         MPØØ135Ø
C CONSTANTS - MPBERN, MPEPS, MPEUL, MPMAXR, MPMINR, MPPI,                 MPØØ136Ø
C                    MPPIGL, MPZETA                                       MPØØ137Ø
C                                                                         MPØØ138Ø
C INPUT AND OUTPUT - MPDUMP, MPIN, MPINE, MPOUT, MPOUTE, MPOUT2           MPØØ139Ø
C                                                                         MPØØ14ØØ
C CONVERSION - MPCDM, MPCIM, MPCMD, CPCMDE, MPCMEF, MPCMI,                MPØØ141Ø
C                    MPCMIM, MPCMR, MPCMRE, MPCQM, MPCRM                  MPØØ142Ø
C                                                                         MPØØ143Ø
C COMPARISON - MPCMPA, MPCMPI, MPCMPR, MPCOMP                             MPØØ144Ø
C                                                                         MPØØ145Ø
C GENERAL UTILITY ROUTINES - MPABS, MPCLR, MPCMF, MPMAX, MPMIN,           MPØØ146Ø
C                    MPNEG, MPPACK, MPPOLY, MPSET, MPSTR, MPUNPK          MPØØ147Ø
C                                                                         MPØØ148Ø
C ERROR DETECTION AND HANDLING - MPCHK, MPERR, MPOVFL, MPUNFL             MPØØ149Ø
C                                                                         MPØØ15ØØ
C TEST PROGRAMS - EXAMPLE, TEST, TESTV, TEST2                             MPØØ151Ø
C                                                                         MPØØ152Ø
C MISCELLANEOUS ROUTINES USED BY THE ABOVE - MPADD2, MPADD3,              MPØØ153Ø
C                    MPART1, MPBES2, MPERF2, MPERF3, MPEXP1,              MPØØ154Ø
C                    MPEXT, MPGCD, MPHANK, MPLNS, MPL235,                 MPØØ155Ø
C                    MPMLP, MPMUL2, MPNZR, MPSIN1, MP4ØD,                 MPØØ156Ø
C                    MP4ØE, MP4ØF, MP4ØG, TIMEMP                          MPØØ157Ø
C                                                                         MPØØ158Ø
C LIST OF MP ROUTINES                                                     MPØØ159Ø
C ******************                                                      MPØØ16ØØ
C                                                                         MPØØ161Ø
C THE ROUTINES PROVIDED IN MP ARE LISTED BELOW.  FOR MORE DETAILS OF      MPØØ162Ø
C CALLING SEQUENCES, RESTRICTIONS, ACCURACY, AND ERROR CONDITIONS,        MPØØ163Ø
C SEE THE COMMENTS IN EACH ROUTINE.  SPACE REQUIRED (I.E. DIMENSION       MPØØ164Ø
C OF R IN COMMON) IS T+4 WORDS UNLESS NOTED BELOW.  THE ROUTINES          MPØØ165Ø
C INDICATED BY * ARE THOSE MOST LIKELY TO BE OF INTEREST TO THE USER.     MPØØ166Ø
C FOR FAST EXECUTION THE ROUTINES WHICH SHOULD BE OPTIMIZED ARE           MPØØ167Ø
C MPNZR, MPMLP, MPDIVI, MPADD2, MPADD3, AND MPMUL2.                       MPØØ168Ø
C                                                                         MPØØ169Ø
C IN THE LIST BELOW AN MP NUMBER MEANS A MULTIPLE-PRECISION               MPØØ17ØØ
C NUMBER AS DESCRIBED ABOVE, M(T) MEANS THE TIME TO MULTIPLY TWO          MPØØ171Ø
C T-DIGIT MP NUMBERS (SEE MPMUL), AN INTEGER MEANS A SINGLE-PRECISION     MPØØ172Ø
C INTEGER, A RATIONAL NUMBER MEANS THE RATIO OF TWO INTEGERS.             MPØØ173Ø
C X, Y, ... ARE MP NUMBERS,  I, J, ... ARE INTEGERS,                      MPØØ174Ø
C AND RI, RX, ... ARE SINGLE-PRECISION REAL NUMBERS.                      MPØØ175Ø
C TIME BOUNDS SUCH AS O(T**2) ARE AS T TENDS TO INFINITY                  MPØØ176Ø
C WITH EVERYTHING ELSE FIXED.                                             MPØØ177Ø
C                                                                         MPØØ178Ø
C *  EXAMPLE   A SMALL MAIN PROGRAM GIVING AN EXAMPLE OF THE USE OF MP.   MPØØ179Ø
C *  MPABS     COMPUTES ABSOLUTE VALUE OF AN MP NUMBER                    MPØØ18ØØ
C              CALL MPABS (X, Y) MEANS Y = ABS(X)                         MPØØ181Ø
C *  MPADD     ADDS TWO MP NUMBERS                                        MPØØ182Ø
C              CALL MPADD (X, Y, Z) MEANS Z = X + Y                       MPØØ183Ø
C *  MPADDI    ADDS AN MP NUMBER TO AN INTEGER,                           MPØØ184Ø
C              GIVING A MULTIPLE-PRECISION RESULT, SPACE = 2T+6           MPØØ185Ø
C              CALL MPADDI (X, IY, Z) MEANS Z = X + IY                    MPØØ186Ø
C *  MPADDQ    ADDS A RATIONAL NUMBER TO AN MP NUMBER,                    MPØØ187Ø
C              SPACE = 2T+6                                               MPØØ188Ø
C              CALL MPADDQ (X, I, J, Y) MEANS Y = X + I/J                 MPØØ189Ø
```

```
C     MPADD2    ROUTINE CALLED BY MPADD AND MPSUB                                MP00190
C     MPADD3    ROUTINE CALLED BY MPADD2                                         MP00191
C     MPART1    COMPUTES ARCTAN(1/N) FOR N .GT. 1 (CALLED BY MPPI)               MP00192
C               SPACE = 2T+6                                                     MP00193
C *   MPASIN    COMPUTES ARCSIN OF AN MP NUMBER,                                 MP00194
C               USING AN O(M(T)T) METHOD, SPACE = 5T+12                          MP00195
C               CALL MPASIN (X, Y) MEANS Y = ARCSIN(X)                           MP00196
C *   MPATAN    COMPUTES ARCTAN OF AN MP NUMBER                                  MP00197
C               USING AN O(T.M(T)) ALGORITHM, SPACE = 5T+12                      MP00198
C               CALL MPATAN (X, Y) MEANS Y = ARCTAN(X)                           MP00199
C *   MPBERN    COMPUTES BERNOULLI NUMBERS B2, B4, B6, ...                       MP00200
C               SPACE = 8T+18                                                    MP00201
C *   MPBESJ    COMPUTES BESSEL FUNCTION J(NU,X) FOR MP X                        MP00202
C               AND SMALL INTEGER NU, SPACE = 14T+156                            MP00203
C               CALL MPBESJ (X, NU, Y) MEANS Y = J(NU,X)                         MP00204
C     MPBES2    ROUTINE CALLED BY MPBESJ (USES BACKWARD RECURRENCE               MP00205
C               TO EVALUATE J(NU,X)), SPACE = 8T+18                              MP00206
C     MPCDM     CONVERTS DOUBLE-PRECISION TO MULTIPLE-PRECISION                  MP00207
C     MPCHK     PRINTS ERROR MESSAGE ON UNIT LUN IF B, T, M OR MXR               MP00208
C               IS ILLEGAL, OR ON UNIT 6 IF LUN IS ILLEGAL                       MP00209
C               (LUN SHOULD BE IN RANGE 1 TO 99)                                 MP00210
C *   MPCIM     CONVERTS INTEGER TO MULTIPLE-PRECISION                           MP00211
C               CALL MPCIM (IX, Z) MEANS Z = IX                                  MP00212
C     MPCLR     SETS SOME DIGITS OF AN MP NUMBER TO ZERO                         MP00213
C     MPCMD     CONVERTS AN MP NUMBER TO DOUBLE-PRECISION REAL                   MP00214
C     MPCMDE    CONVERTS AN MP NUMBER TO (DOUBLE-PRECISION)                      MP00215
C               FRACTION AND (DECIMAL) EXPONENT,                                 MP00216
C               SPACE = 6T+14                                                    MP00217
C     MPCMEF    CONVERTS MP NUMBER TO FRACTION AND (DECIMAL)                     MP00218
C               EXPONENT, SPACE = 5T+12                                          MP00219
C *   MPCMF     FINDS FRACTIONAL PART OF AN MP NUMBER                            MP00220
C     MPCMI     CONVERTS AN MP NUMBER TO AN INTEGER                              MP00221
C *   MPCMIM    CONVERTS AN MP NUMBER TO A MULTIPLE-PRECISION INTEGER            MP00222
C *   MPCMPA    COMPARES ABSOLUTE VALUES OF TWO MP NUMBERS                       MP00223
C               MPCMPA (X, Y) RETURNS SIGN(ABS(X)-ABS(Y))                        MP00224
C *   MPCMPI    COMPARES AN MP NUMBER WITH AN INTEGER, SPACE = 2T+6              MP00225
C               MPCMPI (X, I) RETURNS SIGN(X-I)                                  MP00226
C *   MPCMPR    COMPARES AN MP NUMBER WITH A REAL, SPACE = 2T+6                  MP00227
C               MPCMPR (X, RI) RETURNS SIGN(X-RI)                                MP00228
C *   MPCMR     CONVERTS AN MP NUMBER TO (SINGLE-PRECISION) REAL                 MP00229
C               CALL MPCMR (X, RZ) MEANS RZ = SNGL(X)                            MP00230
C *   MPCMRE    CONVERTS AN MP NUMBER TO EXPONENT AND                            MP00231
C               (SINGLE-PRECISION) FRACTION, I.E. F*10**I                        MP00232
C               SPACE = 6T+14                                                    MP00233
C *   MPCOMP    COMPARES TWO MP NUMBERS                                          MP00234
C               MPCOMP (X, Y) RETURNS SIGN(X-Y)                                  MP00235
C *   MPCOS     COMPUTES COSINE OF AN MP NUMBER, USING AN                        MP00236
C               O(M(T)T/LOG(T)) METHOD, SPACE = 5T+12                            MP00237
C               CALL MPCOS (X, Y) MEANS Y = COS(X)                               MP00238
C *   MPCOSH    COMPUTES HYPERBOLIC COSINE OF AN MP NUMBER                       MP00239
C               USING MPEXP, SPACE = 5T+12                                       MP00240
C               CALL MPCOSH (X, Y) MEANS Y = COSH(X)                             MP00241
C *   MPCQM     CONVERTS A RATIONAL NUMBER TO MULTIPLE-PRECISION                 MP00242
C               CALL MPCQM (I, J, Q) MEANS Q = I/J                               MP00243
C     MPCRM     CONVERTS REAL TO MULTIPLE-PRECISION                              MP00244
C               CALL MPCRM (RX, Z) MEANS Z = RX                                  MP00245
C *   MPDAW     COMPUTES DAWSONS INTEGRAL, DAW(X) = EXP(-X**2)*(INTEGRAL         MP00246
C               FROM 0 TO X OF EXP(U**2) DU),  SPACE = 5T+17                     MP00247
C               CALL MPDAW (X, Y) MEANS Y = DAW(X)                               MP00248
C *   MPDIV     DIVIDES TWO MP NUMBERS, SPACE = 4T+10                            MP00249
C               CALL MPDIV (X, Y, Z) MEANS Z = X/Y                               MP00250
C *   MPDIVI    DIVIDES AN MP NUMBER BY AN INTEGER                               MP00251
C               USING AN O(T) METHOD (MUCH FASTER THAN MPDIV)                    MP00252
```

```
C                CALL MPDIVI (X, IY, Z) MEANS Z = X/IY                    MPØØ253Ø
C     MPDUMP     DUMPS AN MP NUMBER (USEFUL FOR DEBUGGING)                MPØØ254Ø
C                CALL MPDUMP (X) DUMPS THE MP NUMBER X ON UNIT LUN        MPØØ255Ø
C *   MPEI       EVALUATES EXPONENTIAL INTEGRAL OF AN MP NUMBER,          MPØØ256Ø
C                SPACE = 19T+31                                           MPØØ257Ø
C                CALL MPEI (X, Y) MEANS Y = EI(X)                         MPØØ258Ø
C *   MPEPS      COMPUTES THE (MULTIPLE-PRECISION) MACHINE PRECISION      MPØØ259Ø
C                CALL MPEPS (X) MEANS X = Ø.5*B**(1-T) IF B EVEN          MPØØ26ØØ
C *   MPERF      COMPUTES ERROR FUNCTION OF AN MP NUMBER,                 MPØØ261Ø
C                SPACE = 5T+12                                            MPØØ262Ø
C                CALL MPERF (X, Y) MEANS Y = ERF(X)                       MPØØ263Ø
C *   MPERFC     COMPUTES COMPLEMENTARY ERROR FUNCTION OF AN MP NUMBER,   MPØØ264Ø
C                SPACE = 12T+26                                           MPØØ265Ø
C                CALL MPERFC (X, Y)  MEANS Y = ERFC(X)                    MPØØ266Ø
C     MPERF2     COMPUTES EXP(X*X)*(INTEGRAL FROM Ø TO X OF               MPØØ267Ø
C                EXP(-U*U) DU), CALLED BY MPERF, SPACE = 5T+12            MPØØ268Ø
C     MPERF3     ROUTINE CALLED BY MPERF, MPDAW AND MPERFC,               MPØØ269Ø
C                SPACE = 4T+1Ø                                            MPØØ27ØØ
C     MPERR      ERROR HANDLING ROUTINE (TERMINATES EXECUTION AT          MPØØ271Ø
C                PRESENT BUT MAY EASILY BE MODIFIED).                     MPØØ272Ø
C *   MPEUL      RETURNS EULERS CONSTANT (GAMMA = Ø.57721566...) TO       MPØØ273Ø
C                MULTIPLE-PRECISION ACCURACY, SPACE = 5T+14               MPØØ274Ø
C                CALL MPEUL (G) MEANS G = Ø.57721566...                   MPØØ275Ø
C *   MPEXP      COMPUTES EXPONENTIAL OF A MULTIPLE-PRECISION             MPØØ276Ø
C                NUMBER, USING AN O(SQRT(T)M(T)) METHOD, SPACE = 4T+1Ø    MPØØ277Ø
C                CALL MPEXP (X, Y) MEANS Y = EXP(X)                       MPØØ278Ø
C     MPEXP1     COMPUTES EXP(X)-1 FOR ABS(X) .LT. 1  (CALLED BY          MPØØ279Ø
C                MPEXP, MPSINH AND MPTANH),  SPACE = 3T+8                 MPØØ28ØØ
C     MPEXT      A ROUNDING ROUTINE CALLED BY MPDIV AND MPSQRT            MPØØ281Ø
C *   MPGAM      COMPUTES GAMMA FUNCTION OF AN MP ARGUMENT,               MPØØ282Ø
C                SPACE SAME AS FOR MPLNGM (IN WORST CASE)                 MPØØ283Ø
C                CALL MPGAM (X, Y) MEANS Y = GAMMA(X)                     MPØØ284Ø
C     MPGAMQ     COMPUTES GAMMA FUNCTION OF A RATIONAL ARGUMENT,          MPØØ285Ø
C                USING AN O(T**2) METHOD, SPACE = 6T+12                   MPØØ286Ø
C                CALL MPGAMQ (I, J, X) MEANS X = GAMMA(I/J)               MPØØ287Ø
C     MPGCD      DIVIDES TWO INTEGERS BY THEIR GREATEST COMMON DIVISOR    MPØØ288Ø
C                (CALLED BY MPMULQ, MPGAMQ, ETC)                          MPØØ289Ø
C     MPHANK     ROUTINE CALLED BY MPBESJ (EVALUATES HANKELS ASYMPTOTIC   MPØØ29ØØ
C                SERIES FOR BESSEL FUNCTIONS), SPACE = 11T+24             MPØØ291Ø
C *   MPIN       CONVERTS FIXED-POINT NUMBER READ UNDER A1 FORMAT         MPØØ292Ø
C                TO MULTIPLE-PRECISION, SPACE = 3T+11                     MPØØ293Ø
C *   MPINE      SAME AS MPIN BUT RESULT IS MULTIPLIED BY A POWER OF      MPØØ294Ø
C                TEN (USEFUL FOR READING IN FLOATING-POINT NUMBERS),      MPØØ295Ø
C                SPACE = 5T+12                                            MPØØ296Ø
C *   MPLI       EVALUATES LOGARITHMIC INTEGRAL LI(X), SPACE = 19T+31     MPØØ297Ø
C                CALL MPLI (X, Y) MEANS Y = LI(X)                         MPØØ298Ø
C *   MPLN       COMPUTES NATURAL LOG OF AN MP NUMBER,                    MPØØ299Ø
C                USING AN O(SQRT(T).M(T)) METHOD, SPACE = 6T+14           MPØØ3ØØØ
C                CALL MPLN (X, Y) MEANS Y = LN(X)                         MPØØ3Ø1Ø
C     MPLNGM     COMPUTES LN(GAMMA(X)) FOR POSITIVE MP X,                 MPØØ3Ø2Ø
C                USING STIRLINGS APPROXIMATION,                           MPØØ3Ø3Ø
C                SPACE = 11T+24+NL*((T+3)/2), WHERE NL IS THE NUMBER      MPØØ3Ø4Ø
C                OF TERMS USED IN THE ASYMPTOTIC EXPANSION,               MPØØ3Ø5Ø
C                NL .LE. (2 + T*LN(B)/8)                                  MPØØ3Ø6Ø
C                CALL MPLNGM (X, Y) MEANS Y = LN(GAMMA(X))                MPØØ3Ø7Ø
C     MPLNGS     COMPUTES NATURAL LOG OF AN MP NUMBER, USING              MPØØ3Ø8Ø
C                THE GAUSS-SALAMIN ALGORITHM.  RECOMMENDED FOR            MPØØ3Ø9Ø
C                TESTING MPLN AND MPLNI ONLY (UNLESS T LARGE).            MPØØ31ØØ
C                SPACE = 6T+26                                            MPØØ311Ø
C *   MPLNI      COMPUTES NATURAL LOG OF AN INTEGER, USING AN             MPØØ312Ø
C                O(T**2) METHOD (FASTER THAN MPLN), SPACE = 3T+8          MPØØ313Ø
C                CALL MPLNI (N, X) MEANS X = LN(N)                        MPØØ314Ø
C     MPLNS      COMPUTES LN(1+X) FOR SMALL MP X, SPACE = 5T+12           MPØØ315Ø
```

```
C     MPL235    COMPUTES NATURAL LOG OF AN INTEGER WHOSE PRIME          MPØØ316Ø
C               FACTORS ARE 2, 3 AND/OR 5 (CALLED BY MPLNI),            MPØØ317Ø
C               SPACE = 3T+8                                            MPØØ318Ø
C *   MPMAX     COMPUTES THE MAXIMUM OF TWO MP NUMBERS                  MPØØ319Ø
C               CALL MPMAX (X, Y, Z) MEANS Z = MAX(X,Y)                 MPØØ32ØØ
C *   MPMAXR    COMPUTES THE LARGEST POSITIVE MP NUMBER                 MPØØ321Ø
C               CALL MPMAXR (X) MEANS X = MP NUMBER WITH EXPONENT M     MPØØ322Ø
C               AND ALL DIGITS B-1                                      MPØØ323Ø
C *   MPMIN     COMPUTES THE MINIMUM OF TWO MP NUMBERS                  MPØØ324Ø
C               CALL MPMIN (X, Y, Z) MEANS Z = MIN(X,Y)                 MPØØ325Ø
C *   MPMINR    RETURNS THE SMALLEST NORMALIZED POSITIVE MP NUMBER      MPØØ326Ø
C               CALL MPMINR (X) MEANS X = B**(-M-1)                     MPØØ327Ø
C     MPMLP     INNER LOOP ROUTINE CALLED BY MPMUL                      MPØØ328Ø
C *   MPMUL     MULTIPLIES TWO MP NUMBERS                               MPØØ329Ø
C               USING AN M(T) = O(T**2) ALGORITHM                       MPØØ33ØØ
C               CALL MPMUL (X, Y, Z) MEANS Z = X*Y                      MPØØ331Ø
C *   MPMULI    MULTIPLIES AN MP NUMBER BY AN                           MPØØ332Ø
C               INTEGER USING AN O(T) METHOD (FASTER THAN MPMUL)        MPØØ333Ø
C               CALL MPMULI (X, IY, Z) MEANS Z = X*IY                   MPØØ334Ø
C *   MPMULQ    MULTIPLIES MP NUMBER BY A RATIONAL NUMBER               MPØØ335Ø
C               CALL MPMULQ (X, I, J, Y) MEANS Y = X*I/J                MPØØ336Ø
C     MPMUL2    ROUTINE CALLED BY MPMULI                                MPØØ337Ø
C *   MPNEG     REVERSES SIGN OF AN MP NUMBER                           MPØØ338Ø
C               CALL MPNEG (X, Y) MEANS Y = -X                          MPØØ339Ø
C     MPNZR     NORMALIZES AND ROUNDS OR TRUNCATES (CALLED BY           MPØØ34ØØ
C               MPADD2, MPDIVI, MPMUL AND MPMUL2)                       MPØØ341Ø
C *   MPOUT     CONVERTS MULTIPLE-PRECISION TO A FORM SUITABLE FOR      MPØØ342Ø
C               PRINTING UNDER A1 FORMAT (CORRESPONDS TO F OR I         MPØØ343Ø
C               FORMATS), SPACE = 3T+11                                 MPØØ344Ø
C *   MPOUTE    SIMILAR TO MPOUT BUT GIVES (DECIMAL) EXPONENT AND       MPØØ345Ø
C               FRACTION (CORRESPONDS TO E FORMAT), SPACE = 6T+14       MPØØ346Ø
C     MPOUT2    SAME AS MPOUT BUT ANY BASE FROM 2 TO 16 MAY BE          MPØØ347Ø
C               USED FOR OUTPUT REPRESENTATION, SPACE = 3T+11           MPØØ348Ø
C     MPOVFL    ROUTINE CALLED ON MULTIPLE-PRECISION OVERFLOW           MPØØ349Ø
C               (CALLS MPERR AT PRESENT BUT EASILY MODIFIED)            MPØØ35ØØ
C *   MPPACK    PACKS MP NUMBERS INTO ARRAYS OF DIMENSION               MPØØ351Ø
C               (T+3)/2  (USEFUL TO SAVE SPACE),                        MPØØ352Ø
C               UNPACKING MAY BE PERFORMED WITH MPUNPK                  MPØØ353Ø
C *   MPPI      RETURNS PI TO MULTIPLE-PRECISION ACCURACY,              MPØØ354Ø
C               USING AN O(T**2) METHOD, SPACE = 3T+8                   MPØØ355Ø
C               CALL MPPI (X) MEANS X = 3.14159265...                   MPØØ356Ø
C     MPPIGL    RETURNS PI TO MULTIPLE-PRECISION ACCURACY,              MPØØ357Ø
C               USING GAUSS-LEGENDRE O(LOG(T)M(T)) METHOD,              MPØØ358Ø
C               RECOMMENDED FOR TESTING MPPI ONLY, SPACE = 6T+14        MPØØ359Ø
C *   MPPOLY    EVALUATES A POLYNOMIAL WITH INTEGER COEFFICIENTS,       MPØØ36ØØ
C               SPACE = 3T+8                                            MPØØ361Ø
C *   MPPWR     RAISES MP NUMBER TO INTEGER POWER,                      MPØØ362Ø
C               SPACE = 4T+1Ø                                           MPØØ363Ø
C               CALL MPPWR (X, N, Y) MEANS Y = X**N                     MPØØ364Ø
C *   MPPWR2    RAISES NONNEGATIVE MP NUMBER TO MP POWER,               MPØØ365Ø
C               SPACE = 7T+16                                           MPØØ366Ø
C               CALL MPPWR2 (X, Y, Z) MEANS Z = X**Y                    MPØØ367Ø
C *   MPQPWR    RAISES RATIONAL NUMBER TO RATIONAL POWER,               MPØØ368Ø
C               SPACE = 4T+1Ø                                           MPØØ369Ø
C               CALL MPQPWR (I, J, K, L, X) MEANS X = (I/J)**(K/L)      MPØØ37ØØ
C *   MPREC     FORMS RECIPROCAL OF MP NUMBER,                          MPØØ371Ø
C               USING NEWTONS METHOD, SPACE = 4T+1Ø                     MPØØ372Ø
C               CALL MPREC (X, Y) MEANS Y = 1/X                         MPØØ373Ø
C *   MPROOT    COMPUTES THE N-TH ROOT OF AN MP NUMBER                  MPØØ374Ø
C               USING NEWTONS METHOD, SPACE = 4T+1Ø                     MPØØ375Ø
C               CALL MPROOT (X, N, Y) MEANS Y = X**(1/N)                MPØØ376Ø
C *   MPSET     SETS THE BASE B AND DIGITS T ETC GIVEN THE              MPØØ377Ø
```

```
C                EQUIVALENT NUMBER OF DECIMAL PLACES REQUIRED                MP00378Ø
C                WARNING - MAY CAUSE AN INTEGER OVERFLOW,                    MP00379Ø
C                *******   FOR DETAILS SEE COMMENTS IN MPSET                 MP00380Ø
C *  MPSIN       COMPUTES SINE OF AN MP NUMBER,                              MP00381Ø
C                USING AN O(M(T)T/LOG(T)) METHOD, SPACE = 5T+12              MP00382Ø
C                CALL MPSIN (X, Y) MEANS Y = SIN(X)                          MP00383Ø
C *  MPSINH      COMPUTES HYPERBOLIC SINE OF AN MP NUMBER,                   MP00384Ø
C                USING MPEXP, SPACE = 5T+12                                  MP00385Ø
C                CALL MPSINH (X, Y) MEANS Y = SINH(X)                        MP00386Ø
C    MPSIN1      COMPUTES SIN(X) OR COS(X) FOR ABS(X) .LE. 1, CALLED         MP00387Ø
C                BY MPSIN, MPCOS AND MPTAN, SPACE = 3T+8                     MP00388Ø
C *  MPSQRT      COMPUTES SQUARE ROOT OF A NONNEGATIVE MP NUMBER,            MP00389Ø
C                USING NEWTONS METHOD, SPACE = 4T+1Ø                         MP00390Ø
C                CALL MPSQRT (X, Y) MEANS Y = SQRT(X)                        MP00391Ø
C *  MPSTR       STORES ONE MP NUMBER IN ANOTHER                             MP00392Ø
C                CALL MPSTR (X, Y) MEANS Y = X                               MP00393Ø
C *  MPSUB       SUBTRACTS ONE MP NUMBER FROM ANOTHER                        MP00394Ø
C                CALL MPSUB (X, Y, Z) MEANS Z = X - Y                        MP00395Ø
C *  MPTAN       COMPUTES TAN OF AN MP NUMBER,                               MP00396Ø
C                USING MPSIN1, SPACE = 6T+2Ø                                 MP00397Ø
C                CALL MPTAN (X, Y) MEANS Y = TAN(X)                          MP00398Ø
C *  MPTANH      COMPUTES HYPERBOLIC TAN OF AN MP NUMBER,                    MP00399Ø
C                USING MPEXP, SPACE = 5T+12                                  MP00400Ø
C                CALL MPTANH (X, Y) MEANS Y = TANH(X)                        MP00401Ø
C    MPUNFL      ROUTINE CALLED ON MULTIPLE-PRECISION UNDERFLOW              MP00402Ø
C                (SETS RESULT TO ZERO AT PRESENT BUT EASILY MODIFIED)        MP00403Ø
C *  MPUNPK      UNPACKS AN ARRAY FORMED BY MPPACK TO GIVE AN MP             MP00404Ø
C                NUMBER IN STANDARD FORMAT                                   MP00405Ø
C *  MPZETA      COMPUTES RIEMANN ZETA FUNCTION FOR POSITIVE                 MP00406Ø
C                INTEGER ARGUMENTS                                           MP00407Ø
C                SPACE = 8T+18+NL*((T+3)/2), WHERE NL IS THE                 MP00408Ø
C                NUMBER OF TERMS USED IN THE ASYMPTOTIC                      MP00409Ø
C                EXPANSION, NL .LE. (1 + Ø.1*T*LN(B))                        MP00410Ø
C                CALL MPZETA (N, X) MEANS X = ZETA(N)                        MP00411Ø
C *  MP4ØD       OUTPUT ROUTINE CALLED BY TEST PROGRAM,                      MP00412Ø
C                USEFUL FOR EASY FIXED-POINT OUTPUT,                         MP00413Ø
C                SPACE = 3T+N+14 FOR N DECIMAL PLACE OUTPUT                  MP00414Ø
C                CALL MP4ØD (N, X) WRITES X TO N DECIMAL PLACES ON UNIT      MP00415Ø
C                LUN, ASSUMING ABS(X) .LT. 1Ø                                MP00416Ø
C    MP4ØE       OUTPUT ROUTINE CALLED BY MP4ØD                              MP00417Ø
C *  MP4ØF       OUTPUT ROUTINE CALLED BY TEST2 PROGRAM,                     MP00418Ø
C                USEFUL FOR EASY FLOATING-POINT OUTPUT,                      MP00419Ø
C                SPACE = 6T+N+17 FOR N SIGNIFICANT FIGURE OUTPUT             MP00420Ø
C                CALL MP4ØF (N, X) WRITES X TO N SIGNIFICANT FIGURES         MP00421Ø
C                (IN DECIMAL EXPONENT AND FRACTION FORM) ON UNIT LUN         MP00422Ø
C    MP4ØG       OUTPUT ROUTINE CALLED BY MP4ØF                              MP00423Ø
C *  TEST        A MAIN PROGRAM WHICH TESTS SOME OF THE ROUTINES IN MP       MP00424Ø
C                WHILE COMPUTING VARIOUS CONSTANTS TO 4Ø DECIMAL PLACES      MP00425Ø
C    TESTV       A VERSION OF TEST WITH VARIABLE-PRECISION COMPUTATION       MP00426Ø
C                AND OUTPUT                                                  MP00427Ø
C *  TEST2       ANOTHER TEST PROGRAM WHICH TESTS ROUTINES                   MP00428Ø
C                NOT CALLED BY TEST OR TESTV                                 MP00429Ø
C    TIMEMP      A MACHINE-DEPENDENT FUNCTION CALLED BY TESTV, SHOULD        MP00430Ø
C                BE MODIFIED BY THE USER BEFORE TESTV IS RUN.                MP00431Ø
C                                                                            MP00432Ø
C INDEX                                                                      MP00433Ø
C *****                                                                      MP00434Ø
C                                                                            MP00435Ø
C THE STARTING LINE SEQUENCE NUMBERS (GIVEN IN COLUMNS 73-8Ø)                MP00436Ø
C OF THE MP ROUTINES ARE AS FOLLOWS.                                         MP00437Ø
C                                                                            MP00438Ø
C     COMMENTS     MP00003Ø                                                  MP00439Ø
C     EXAMPLE      MP00546Ø                                                  MP00440Ø
```

```
C     MPABS      MP00619Ø                    MP00441Ø
C     MPADD      MP00627Ø                    MP00442Ø
C     MPADDI     MP00635Ø                    MP00443Ø
C     MPADDQ     MP00651Ø                    MP00444Ø
C     MPADD2     MP00663Ø                    MP00445Ø
C     MPADD3     MP00723Ø                    MP00446Ø
C     MPART1     MP00813Ø                    MP00447Ø
C     MPASIN     MP00861Ø                    MP00448Ø
C     MPATAN     MP00899Ø                    MP00449Ø
C     MPBERN     MP00961Ø                    MP00450Ø
C     MPBESJ     MP01Ø76Ø                    MP00451Ø
C     MPBES2     MP01190Ø                    MP00452Ø
C     MPCDM      MP01258Ø                    MP00453Ø
C     MPCHK      MP01325Ø                    MP00454Ø
C     MPCIM      MP01376Ø                    MP00455Ø
C     MPCLR      MP0140ØØ                    MP00456Ø
C     MPCMD      MP01416Ø                    MP00457Ø
C     MPCMDE     MP01456Ø                    MP00458Ø
C     MPCMEF     MP01479Ø                    MP00459Ø
C     MPCMF      MP01550Ø                    MP00460Ø
C     MPCMI      MP01582Ø                    MP00461Ø
C     MPCMIM     MP01625Ø                    MP00462Ø
C     MPCMPA     MP01648Ø                    MP00463Ø
C     MPCMPI     MP01664Ø                    MP00464Ø
C     MPCMPR     MP01680Ø                    MP00465Ø
C     MPCMR      MP01696Ø                    MP00466Ø
C     MPCMRE     MP01731Ø                    MP00467Ø
C     MPCOMP     MP01753Ø                    MP00468Ø
C     MPCOS      MP01788Ø                    MP00469Ø
C     MPCOSH     MP01817Ø                    MP00470Ø
C     MPCQM      MP01843Ø                    MP00471Ø
C     MPCRM      MP01863Ø                    MP00472Ø
C     MPDAW      MP01926Ø                    MP00473Ø
C     MPDIV      MP01990Ø                    MP00474Ø
C     MPDIVI     MP02Ø38Ø                    MP00475Ø
C     MPDUMP     MP02151Ø                    MP00476Ø
C     MPEI       MP02177Ø                    MP00477Ø
C     MPEPS      MP02295Ø                    MP00478Ø
C     MPERF      MP02324Ø                    MP00479Ø
C     MPERFC     MP02381Ø                    MP00480Ø
C     MPERF2     MP02431Ø                    MP00481Ø
C     MPERF3     MP02484Ø                    MP00482Ø
C     MPERR      MP02539Ø                    MP00483Ø
C     MPEUL      MP02558Ø                    MP00484Ø
C     MPEXP      MP02634Ø                    MP00485Ø
C     MPEXP1     MP02737Ø                    MP00486Ø
C     MPEXT      MP02800Ø                    MP00487Ø
C     MPGAM      MP02824Ø                    MP00488Ø
C     MPGAMQ     MP02905Ø                    MP00489Ø
C     MPGCD      MP03Ø30Ø                    MP00490Ø
C     MPHANK     MP03Ø56Ø                    MP00491Ø
C     MPIN       MP03139Ø                    MP00492Ø
C     MPINE      MP03257Ø                    MP00493Ø
C     MPLI       MP033Ø2Ø                    MP00494Ø
C     MPLN       MP03338Ø                    MP00495Ø
C     MPLNGM     MP03394Ø                    MP00496Ø
C     MPLNGS     MP03481Ø                    MP00497Ø
C     MPLNI      MP03558Ø                    MP00498Ø
C     MPLNS      MP03663Ø                    MP00499Ø
C     MPL235     MP0373ØØ                    MP00500Ø
C     MPMAX      MP03783Ø                    MP00501Ø
C     MPMAXR     MP03795Ø                    MP00502Ø
C     MPMIN      MP0381ØØ                    MP00503Ø
```

```
C      MPMINR        MPØ3823Ø                                                MPØØ5Ø4Ø
C      MPMLP         MPØ3839Ø                                                MPØØ5Ø5Ø
C      MPMUL         MPØ3849Ø                                                MPØØ5Ø6Ø
C      MPMULI        MPØ3924Ø                                                MPØØ5Ø7Ø
C      MPMULQ        MPØ3934Ø                                                MPØØ5Ø8Ø
C      MPMUL2        MPØ396ØØ                                                MPØØ5Ø9Ø
C      MPNEG         MPØ4Ø48Ø                                                MPØØ51ØØ
C      MPNZR         MPØ4Ø56Ø                                                MPØØ511Ø
C      MPOUT         MPØ4137Ø                                                MPØØ512Ø
C      MPOUTE        MPØ4152Ø                                                MPØØ513Ø
C      MPOUT2        MPØ4184Ø                                                MPØØ514Ø
C      MPOVFL        MPØ4323Ø                                                MPØØ515Ø
C      MPPACK        MPØ4344Ø                                                MPØØ516Ø
C      MPPI          MPØ4371Ø                                                MPØØ517Ø
C      MPPIGL        MPØ4397Ø                                                MPØØ518Ø
C      MPPOLY        MPØ4442Ø                                                MPØØ519Ø
C      MPPWR         MPØ447ØØ                                                MPØØ52ØØ
C      MPPWR2        MPØ4513Ø                                                MPØØ521Ø
C      MPQPWR        MPØ4544Ø                                                MPØØ522Ø
C      MPREC         MPØ4617Ø                                                MPØØ523Ø
C      MPROOT        MPØ4698Ø                                                MPØØ524Ø
C      MPSET         MPØ48Ø4Ø                                                MPØØ525Ø
C      MPSIN         MPØ4881Ø                                                MPØØ526Ø
C      MPSINH        MPØ4948Ø                                                MPØØ527Ø
C      MPSIN1        MPØ4986Ø                                                MPØØ528Ø
C      MPSQRT        MPØ5Ø44Ø                                                MPØØ529Ø
C      MPSTR         MPØ5Ø68Ø                                                MPØØ53ØØ
C      MPSUB         MPØ5Ø89Ø                                                MPØØ531Ø
C      MPTAN         MPØ5Ø98Ø                                                MPØØ532Ø
C      MPTANH        MPØ5161Ø                                                MPØØ533Ø
C      MPUNFL        MPØ52ØØØ                                                MPØØ534Ø
C      MPUNPK        MPØ5215Ø                                                MPØØ535Ø
C      MPZETA        MPØ5244Ø                                                MPØØ536Ø
C      MP4ØD         MPØ5359Ø                                                MPØØ537Ø
C      MP4ØE         MPØ5376Ø                                                MPØØ538Ø
C      MP4ØF         MPØ539ØØ                                                MPØØ539Ø
C      MP4ØG         MPØ54Ø8Ø                                                MPØØ54ØØ
C      TEST          MPØ5422Ø                                                MPØØ541Ø
C      TESTV         MPØ56Ø3Ø                                                MPØØ542Ø
C      TEST2         MPØ5794Ø                                                MPØØ543Ø
C      TIMEMP        MPØ6414Ø                                                MPØØ544Ø
C                                                                            MPØØ545Ø
C $$                    ******  EXAMPLE  ******                              MPØØ546Ø
C                                                                            MPØØ547Ø
C THIS PROGRAM COMPUTES PI AND EXP(PI*SQRT(163/9)) TO 1ØØ                    MPØØ548Ø
C DECIMAL PLACES, AND EXP(PI*SQRT(163)) TO 9Ø DECIMAL PLACES,                MPØØ549Ø
C AND WRITES THEM ON LOGICAL UNIT 6.  EXECUTION                              MPØØ55ØØ
C TIME ON A UNIVAC 11Ø8 (WITH FORTRAN SE1D) IS 1.Ø51 SECONDS.                MPØØ551Ø
C                                                                            MPØØ552Ø
C TO RUN EXAMPLE THE FOLLOWING MP ROUTINES ARE REQUIRED - MPABS,             MPØØ553Ø
C MPADD, MPADDI, MPADD2, MPADD3, MPART1, MPCHK, MPCIM, MPCLR, MPCMF,         MPØØ554Ø
C MPCMI, MPCMPR, MPCMR, MPCOMP, MPCQM, MPCRM, MPDIVI, MPERR,                 MPØØ555Ø
C MPEXP, MPEXP1, MPGCD, MPLNI, MPL235, MPMAXR, MPMLP, MPMUL,                 MPØØ556Ø
C MPMULI, MPMULQ, MPMUL2, MPNZR, MPOUT, MPOUT2, MPOVFL, MPPI,               MPØØ557Ø
C MPPWR, MPQPWR, MPREC, MPROOT, MPSET, MPSTR, MPSUB, MPUNFL.                 MPØØ558Ø
C                                                                            MPØØ559Ø
C CORRECT OUTPUT (EXCLUDING HEADINGS) IS AS FOLLOWS                          MPØØ56ØØ
C                                                                            MPØØ561Ø
C                  3.14159265358979323846264338327950288419716939937510 MPØØ562Ø
C                    582Ø974944592Ø78164Ø628620899862803482534211706Ø8Ø MPØØ563Ø
C              64Ø32Ø.ØØØØØØØØØ6Ø4863735Ø49Ø16Ø3947174181881853947577148 MPØØ564Ø
C                    576Ø3665918194652218258286942536340815822646477590 MPØØ565Ø
C 262537412640768743.99999999999925ØØ725971981856888793538563373369908 6 MPØØ566Ø
C                    27Ø753741Ø37821Ø64791Ø1186Ø7312951181346              MPØØ567Ø
```

```
C                                                                       MPØØ568Ø
C CERTAIN PARAMETERS AND WORKING SPACE IN COMMON.                       MPØØ569Ø
      COMMON B, T, M, LUN, MXR, R                                       MPØØ57ØØ
C                                                                       MPØØ571Ø
C MPEXP REQUIRES 4T+1Ø WORDS AND WE HAVE T .LE. 62 IF WORDLENGTH        MPØØ572Ø
C AT LEAST 16 BITS, SO 4T+1Ø .LE. 258.  DIMENSIONS CAN BE REDUCED       MPØØ573Ø
C IF WORDLENGTH IS GREATER THAN 16 BITS.                                MPØØ574Ø
      INTEGER B, T, R(258)                                              MPØØ575Ø
C                                                                       MPØØ576Ø
C VARIABLES NEED T+2 .LE. 64 WORDS AND ALLOW 11Ø CHARACTERS FOR         MPØØ577Ø
C DECIMAL OUTPUT                                                        MPØØ578Ø
      INTEGER PI(64), X(64), C(11Ø)                                     MPØØ579Ø
C                                                                       MPØØ58ØØ
C CALL MPSET TO SET OUTPUT LOGICAL UNIT = 6 AND EQUIVALENT              MPØØ581Ø
C NUMBER OF DECIMAL PLACES TO AT LEAST 11Ø.  THE LAST TWO               MPØØ582Ø
C PARAMETERS ARE THE DIMENSIONS OF PI (OR X) AND R.                     MPØØ583Ø
      CALL MPSET (6, 11Ø, 64, 258)                                      MPØØ584Ø
C                                                                       MPØØ585Ø
C COMPUTE MULTIPLE-PRECISION PI                                         MPØØ586Ø
      CALL MPPI(PI)                                                     MPØØ587Ø
C                                                                       MPØØ588Ø
C CONVERT TO PRINTABLE FORMAT (F11Ø.1ØØ) AND WRITE                      MPØØ589Ø
      CALL MPOUT (PI, C, 11Ø, 1ØØ)                                     MPØØ59ØØ
      WRITE (LUN, 1Ø) B, T, C                                           MPØØ591Ø
   1Ø FORMAT (32H1EXAMPLE OF MP PACKAGE,   BASE =, I9,                  MPØØ592Ø
     $  12H,   DIGITS =, I4 /// 11H PI TO 1ØØD //                       MPØØ593Ø
     $  11X, 6ØA1 / 21X, 5ØA1)                                          MPØØ594Ø
C                                                                       MPØØ595Ø
C SET X = SQRT(163/9), THEN MULTIPLY BY PI                              MPØØ596Ø
      CALL MPQPWR (163, 9, 1, 2, X)                                     MPØØ597Ø
      CALL MPMUL (X, PI, X)                                             MPØØ598Ø
C                                                                       MPØØ599Ø
C SET X = EXP(X)                                                        MPØØ6ØØØ
      CALL MPEXP (X, X)                                                 MPØØ6Ø1Ø
C                                                                       MPØØ6Ø2Ø
C CONVERT TO PRINTABLE FORMAT AND WRITE                                 MPØØ6Ø3Ø
      CALL MPOUT (X, C, 11Ø, 1ØØ)                                      MPØØ6Ø4Ø
      WRITE (LUN, 2Ø) C                                                 MPØØ6Ø5Ø
   2Ø FORMAT (/ 28H EXP(PI*SQRT(163/9)) TO 1ØØD //                      MPØØ6Ø6Ø
     $        11X, 6ØA1 / 21X, 5ØA1)                                   MPØØ6Ø7Ø
C                                                                       MPØØ6Ø8Ø
C SET X = X**3 = EXP(PI*SQRT(163))                                      MPØØ6Ø9Ø
      CALL MPPWR (X, 3, X)                                              MPØØ61ØØ
C                                                                       MPØØ611Ø
C WRITE IN FORMAT F11Ø.9Ø                                               MPØØ612Ø
      CALL MPOUT (X, C, 11Ø, 9Ø)                                       MPØØ613Ø
      WRITE (LUN, 3Ø) C                                                 MPØØ614Ø
   3Ø FORMAT (/ 25H EXP(PI*SQRT(163)) TO 9ØD //                         MPØØ615Ø
     $        1X, 7ØA1 / 21X, 4ØA1)                                    MPØØ616Ø
      STOP                                                              MPØØ617Ø
      END                                                               MPØØ618Ø
```