PROGRAMS FOR DIGITAL SIGNAL PROCESSING

Programs for Digital Signal Processing

Edited by the

Digital Signal Processing Committee

IEEE Acoustics, Speech, and Signal Processing Society

Committee members:

Clifford J. Weinstein (Chairman), M.I.T. Lincoln Laboratory
James W. Cooley, IBM T. J. Watson Research Center
Ronald E. Crochiere, Bell Laboratories
Marie T. Dolan, Bell Laboratories
Joseph R. Fisher, Signal Processing Systems, Inc.
Howard D. Helms, American Telephone and Telegraph Company
Leland B. Jackson, University of Rhode Island
James F. Kaiser, Bell Laboratories
James H. McClellan, Massachusetts Institute of Technology
Carol A. McGonegal, Bell Laboratories
Russell M. Mersereau, Georgia Institute of Technology
Alan V. Oppenheim, Massachusetts Institute of Technology
Lawrence R. Rabiner, Bell Laboratories
Charles M. Rader, M.I.T. Lincoln Laboratory
Ronald W. Schafer, Georgia Institute of Technology
Harvey F. Silverman, IBM T. J. Watson Research Center
Kenneth Steiglitz, Princeton University
Jose M. Tribolet, Instituto Superior Tecnico
John W. Woods, Rensselaer Polytechnic Institute

A volume published under the sponsorship of the
IEEE Acoustics, Speech, and Signal Processing Society.

IEEE PRESS

The Institute of Electrical and Electronics Engineers, Inc. New York

Programs for Digital Signal Processing

Edited by the

Digital Signal Processing Committee
IEEE Acoustics, Speech, and Signal Processing Society

IEEE International Standard Book Numbers: Clothbound: 0-87942-127-4
Paperbound: 0-87942-128-2

Library of Congress Catalog Card Number 79-89028

Sole Worldwide Distributor (Exclusive of the IEEE):

JOHN WILEY & SONS, INC.
605 Third Ave.
New York, NY 10016

Wiley Order Numbers: Clothbound: 0-471-05962-5
Paperbound: 0-471-05961-7

Preface

During the past fifteen years, digital signal processing has been an extremely active and dynamic field. Advances in integrated circuit technology and in processor architecture have greatly enlarged the scope of the technical areas to which digital signal processing techniques can be applied. Research in fundamental signal processing techniques and algorithms has led to dramatic improvements in the efficiency of signal processing systems.

An important facet of the progress in digital signal processing has been the development of algorithms and their embodiment in computer programs, both for the execution of processing operations on signals and for the design of signal processing filters and systems. The purpose of this book is to make widely available, in a directly usable form, a comprehensive set of computer programs useful for digital signal processing. In addition, the book serves as an outlet for excellent programming effort and enables authors of programs to receive credit for their work.

The programs have been carefully selected to cover a broad spectrum of digital signal processing applications and design techniques. The programs are categorized into eight chapters, and separate summaries (authored by chapter editors who were also asked to make final checks on the documentation and code) are provided with each chapter. The first chapter focuses on the Discrete Fourier Transform (DFT) and presents a variety of Fast Fourier Transform (FFT) and related algorithms. Chapter 2 includes algorithms for the periodogram and correlation methods of power spectrum estimation and for coherence and cross spectrum estimation. A program for high speed, FFT-based convolution is presented in Chapter 3. Chapter 4 presents several algorithms related to the linear prediction techniques of signal processing, including the autocorrelation, covariance, and lattice methods. The design and synthesis of Finite Impulse Response (FIR) digital filters are the subjects of Chapter 5. Algorithms for optimal (minimax), windowed, and maximally-flat filter designs, and a design program which incorporates finite-word-length effects, are included. Chapter 6 presents a comprehensive set of programs dealing with the design and synthesis of Infinite Impulse Response (IIR) digital filters. The first program in that chapter includes most of the classical filter design techniques as well as consideration of finite word length issues such as pairing and ordering in a single but modular package. The other programs in that chapter are specialized either to the finite word length design problem or to filter design based on iterative optimization. Chapter 7 deals with cepstral and homomorphic algorithms, with specific attention to the difficult problem of phase calculation in a homomorphic system. Finally, Chapter 8 presents several programs for interpolation and decimation, the fundamental operations necessary for changing sampling rates in a digital system. Included are multistage implementations and sampling rate conversion by rational ratios.

The book is the culmination of a project undertaken in early 1976 by the Digital Signal Processing Committee of the IEEE Acoustics, Speech, and Signal Processing Society. In contrast to previous books in the IEEE Press reprint series, this book consists largely of material which has not been published before. In addition, a unique and complex set of issues regarding the publication of computer programs had to be dealt with. These issues included: (1) clarity and uniformity of documentation and program presentation; (2) independent review and verification of all programs; (3) portability and machine-independence of code; and (4) availability of computer-readable source programs. The project has therefore been a very ambitious and time-consuming one for the committee, for the program contributors, and for the reviewers.

In order to assess the feasibility of the project, the committee distributed to the digital signal processing community in early 1976 a description of the proposed project and a request for descriptions of candidate programs. The distribution of this request was carried out both through direct mailing and through handouts at the 1976 International Conference on Acoustics, Speech, and Signal Processing. In a short time, enough very promising candidate programs had been identified to convince the committee that the book was not only feasible, but potentially a landmark contribution to the digital signal processing field. In addition, a reasonable organization of the material into the eight sections now included in the book began to emerge.

Because of the large effort to be required of program contributors and of reviewers, it was deemed desirable to pre-screen program candidates based on the offered descriptions, and invite submission of only those programs judged most likely to fit our rather stringent requirements. In cases where two

programs were offered to accomplish the same functions, a choice was generally made based on which of the programs was closer to publishable form in terms of documentation, portability, verification through wide previous use, or other similar criteria.

After this pre-screening, invitations were sent to prospective authors to submit full code and documentation for their programs. It was required of the authors that all programs meet the following conditions:

(1) complete user documentation must be included;

(2) the program must be in FORTRAN and conform to ANSI FORTRAN standards;

(3) the code should be fully commented and debugged, and as clear and understandable as possible;

(4) the code should be transportable and machine-independent;

(5) comprehensive test examples with corresponding input and output should be supplied.

Of course, in addition to printouts, a computer-readable source of all code had to be provided by the authors. This led to the rather onerous task of collecting the submitted tapes and cards, reading them into a central filing system, and distributing them to reviewers in one of several possible formats.

The committee viewed it to be an essential part of the project that computer-readable source material for the programs be made available to readers of the book. The collection of all programs into a central file was clearly necessary to achieve this. In addition, with the cooperation of Reed Crone of IEEE Press and Jack Fraum of the IEEE Service Center, we were able to arrange for IEEE to offer the service of reproducing and distributing, upon request and at a nominal cost, a tape of all programs in the book. For information on how to order such a tape, contact the IEEE Press, 345 East 47th St., New York, NY 10017. It is worth emphasizing that this is a new service which IEEE has not offered in the past, and which can perhaps be utilized again in other projects.

As authors submitted programs, the review process was able to proceed. Each reviewer was asked to act from points of view (1) of a highly critical program user, and (2) of an editorial reviewer of journal-quality technical material. The reviewer was asked to read the program description, compile the program, and attempt to run the test examples described as well as other test examples considered appropriate. The review process was an open one, with reviewers frequently consulting directly with authors on points of difficulty. In most cases it was found that the material submitted by authors was excellent and well-debugged, and the review process went quite smoothly. However, in some cases serious difficulties, particularly in the area of program portability, were uncovered and corrected in the review process. At the end of the book, we have identified the reviewer(s) as well as the computer(s) and operating system(s) on which each program was tested. In addition to giving credit for the efforts of the reviewers, this serves to identify for readers at least one successful user of each program besides the author(s).

Program portability was the objective of major importance in this project. One successful review was not considered sufficient to insure that this goal was met; therefore an extensive parallel effort in program standardization was undertaken. The standardization procedures are described in detail in the standards section.

Much care has been taken to insure the accuracy, clarity, and uniformity of the printed material in the book. All listings of program code have been printed directly from the centrally-collected computer-readable source material, with particular attention to the selection of a readable type font. This was considered necessary to avoid either direct photoreproduction of the assorted type fonts produced by authors or the inevitable errors which would be caused by attempting to transcribe and typeset the listings. The authors' manuscripts documenting their programs were typed into a computer system, after which final editing and correcting were carried out. All the printed material in the book was then produced by computer-driven phototypesetters.

Although this has been a project of the full Digital Signal Processing Committee as listed on the title page, a number of outstanding individual contributions deserve special mention here. The idea for this project was first presented to the committee by Ronald Crochiere in early 1976, and the project was

initiated under the committee chairmanship of Alan Oppenheim. Crochiere was responsible for the initial solicitation and collection of candidate programs from the digital signal processing community. Clifford Weinstein took over as committee chairman in September 1976 and since that time served as general coordinator for the program book project.

The task of collecting submitted tapes and cards from the authors, reading them into a central file, and distributing them for review in specified formats, was initially carried out by Howard Helms and later taken over by Marie Dolan. Helms was also instrumental in arranging for IEEE reproduction of the program tape for users. To insure the standardization and portability of the programs, a standardization subcommittee including James Kaiser, Dolan, Carol McGonegal, Lawrence Rabiner, and Jose Tribolet was made responsible for detailed specification and execution of the standards imposed on the programs in this book. The typing of author manuscripts into a computer file was carried out by Penny Blaine and Carmela Patuto. Art editing and page layout were skillfully handled by Madeline Wilson. Kaiser coordinated the handling of manuscripts and arranged for the phototypesetting of the book. Much of this phototypesetting was done by Yourdon, Inc., New York City. The subcommittee of Kaiser, Dolan, McGonegal, and Rabiner took charge of tracking and carrying out the many details required to bring the book into final form.

Special thanks are due to Marie Dolan and Carol McGonegal, who joined the committee for the express purpose of participating in the program book project and who put forth enormous and very productive effort in seeing it through to completion. We would like to acknowledge the excellent cooperation and assistance provided to this project by IEEE Press Managing editor Reed Crone, and the specific efforts of Jack Fraum in arranging for IEEE reproduction of the program tape. Finally, the Committee must thank the program authors, reviewers, and chapter editors for the large amounts of time and effort they contributed to this project.

Standards

In putting together this collection of programs on digital signal processing, it was immediately recognized that users would be trying to run the different programs on a wide variety of computing machines and under several types of operating systems. For this reason FORTRAN was chosen as the principal language and a great deal of attention was given to the problem of program portability. The goal was for each program to compile and execute, according to its defined performance criteria, on a wide range of computers. Adopted was the PORT Mathematical Subroutine Library approach toward portability of Fox, Hall, and Schryer [1]. "That is, all programs had to be portable in the IFIP sense*, with the exception that the three machine-constant defining functions had to be particularized to the host computer once, at installation". The techniques used in the PORT library and in the programs in this book to make them easily portable were: (1) the programs are written in a subset of ANSI FORTRAN, and (2) the target environment is specified in terms of machine-dependent parameters. In addition to portability, program readability was given consideration in this collection of programs. The goal was to enhance the readability of the program code by standardizing the format of the programs. Following is a discussion of the techniques used to make the programs portable and enhance their readability.

Language

The programs were restricted to the particular portable subset of ANSI FORTRAN known as PFORT (Portable FORTRAN) [3]. The PFORT verifier program [3] checks that individual program units conform to the American National Standard FORTRAN [4]; it also checked that the inter-program-unit communication through the use of COMMON and argument lists is consistent with the standard. During the review each program was passed through the PFORT verifier to guarantee its adherence to this language requirement.

Machine-Dependent Quantities

All machine-dependent quantities (I/O device codes, machine constants, etc.) are specified by the I1MACH, R1MACH and D1MACH FORTRAN function subprograms used in the PORT [1,5] library. Table 1 is a list of all the machine-dependent quantities which are specified by these routines. They are delivered by means of a single integer argument indicating the particular quantity desired. Complete listings of I1MACH, R1MACH and D1MACH are given in the Appendix. Any program in this book which is not truly portable, that is, which uses machine-dependent quantities not given in Table 1, is however still transportable in the IFIP sense. Additional machine-dependent quantities have been kept to a minimum; instructions for changing their values are documented in the program code.

To move the programs to a new environment, only the DATA statements in the I1MACH, R1MACH and D1MACH subprograms need to be changed. Values are provided in these routines for a number of different computing systems. They include the Burroughs 1700 system, the Burroughs 5700/6700/7700 systems, the CDC 6000/7000 series, the CRAY 1, the Data General Eclipse S/200, the Harris Slash 6 and Slash 7, the Honeywell 6000 series, the IBM 360/370 series, the Xerox SIGMA 5/7/9, the SEL Systems 85/86, the DEC PDP 10 (KA and KI processors), the DEC PDP 11, the UNIVAC 1100 series, and the VAX-11.

To aid in adding values for a target environment not provided in these subprograms, a description of the integer and floating-point variables follows. This description appears in the PORT Mathematical Subroutine Library report [1].

---

* The IFIP Working Group (on Numerical Software) (WG2.5) has proposed, in a working draft, the following definitions [2]
Portable. A program is portable over a given range of machines and compilers if, *without any alteration*, it can compile and run to satisfy specified performance criteria on that range.
Transportable. In transferring a program between members of a given range of machines and compilers, some changes may be necessary to the base version before it satisfies specified performance criteria on each of the machines and compilers. The program is transportable if (1) the changes lend themselves to mechanical implementation by a processor, and (2) the changes, ideally, are limited in number, extent, and complexity.

Integer variables: Let the values for integer variables be written in the s-digit, base-a form:

$$\pm(x_{s-1}a^{s-1} + x_{s-2}a^{s-2} + \cdots + x_1a + x_0)$$

where $0 \leq x_i < a$ for $i = 0,\ldots s-1$. Then specify the base a, the maximum number of digits s, and the largest integer a^s-1. Although the quantity a^s-1 can easily be computed from s and the base a, it is provided because a naive evaluation of the formula would cause overflow on most machines and hence result in an incorrect value. (Storage of integers as magnitude and sign or in a complement notation is not specified since PORT subprograms must be independent of the storage mode.)

Floating-point variables: If floating-point numbers are written in the t-digit, base-b form:

$$\pm b^e \left(\frac{x_1}{b} + \frac{x_2}{b^2} + \cdots + \frac{x_t}{b^t} \right)$$

where $0 \leq x_i < b$ for $i=1,\ldots,t$, $0 < x_1$, and $e_{min} \leq e \leq e_{max}$, then for a particular machine, choose values for the parameters t, e_{min}, and e_{max} such that all numbers expressible in this form are representable by the hardware and usable from FORTRAN. Note that the formula is symmetrical under negation but not reciprocation. On some machines a small portion of the range of permissible numbers may be excluded. Also, for 2's complement machines, care must be taken in assigning the values.

For each machine one must specify for real (single-precision) floating-point numbers, the base, b, the number t of base-b digits, the minimum exponent e_{min} and the maximum exponent e_{max}. For double-precision numbers, b remains the same, but t, e_{min}, and e_{max} are replaced by T, E_{min}, and E_{max}.

The 16 parameters discussed above are all integers and are obtained by invoking the function IIMACH with the appropriate argument. The floating-point single-precision and double-precision quantities provided by the functions R1MACH and D1MACH can be derived from the given integer quantities, but are provided for efficiency and convenience.

The single-precision floating-point quantities provided in R1MACH are the smallest positive magnitude, $b^{e_{min}-1}$, the largest magnitude, $b^{e_{max}}(1-b^{-t})$, the smallest relative spacing between values, b^{-t}, the largest relative spacing between values, $b^{(1-t)}$, and the logarithm of the base b, $\log_{10}b$. The relative spacing is $|(y-x)/x|$, when x and y are successive floating-point numbers. Equivalent values for the double-precision floating-point quantities are provided by D1MACH, with e_{min}, e_{max}, and t replaced by E_{min}, E_{max}, and T.

Portable Random Number Generator

In addition to the machine-dependent quantities, the PORT library was the source of the portable random number generator, UNI. UNI, written by Alan Gross, is a uniform generator which will produce the same sequence of values, to the accuracy of the computer, on any computer with 16 or more bits. UNI is implemented as a FORTRAN function subprogram and returns a single real random variate from the uniform [0,1) distribution. A complete listing of UNI is given in the Appendix. There are three local variables (FIRST, CSEED in UNI and TSEED in the routine R1UNIF) which must retain their most recently assigned values. In order to adhere to this requirement on a Data General Computer, the STATIC mode must be set when UNI is compiled.

Program Readability

In order to enhance the readability of the programs, two steps were taken. First, the authors were asked to precede each program unit with a specific comment structure. This structure was designed to allow the reader to easily identify a program unit. Second, the programs (except for 1.7, 1.8, and 5.1) were passed through the POLISH program [6]. POLISH was written at the University of Colorado at Boulder and is intended to improve and standardize the format of FORTRAN programs. It does this by renumbering all statement labels, by inserting CONTINUE statements to force every DO to end on a unique CONTINUE statement, and by systematically spacing and indenting each statement in the program.

References

1. P. A. Fox, A. D. Hall, and N. L. Schryer, "The PORT Mathematical Subroutine Library", *ACM Transactions on Mathematical Software*, Vol. 4, No. 2, pp. 104-126, June 1978.
2. B. Ford and B. T. Smith, "Transportable mathematical software: A substitute for portable mathematical software", Position paper, IFIP Working Group 2.5 (on Numerical Software), 1975.
3. B. G. Ryder, "The PFORT Verifier", *Software-Practice and Experience*, Vol. 4, No. 4, pp. 359-377, Oct.-Dec. 1974.
4. American National Standard X3.9-1966 (ISO 1539-1972), FORTRAN, American National Standards Institute (ANSI), New York, 1966.
5. P. A. Fox, A. D. Hall, and N. L. Schryer, "ALGORITHM 528, Framework for a Portable Library [Z]", *ACM Transactions on Mathematical Software*, Vol. 4, No. 2, pp. 177-188, June 1978.
6. J. Dorrenbacher, D. Paddock, D. Wisneski, and L. D. Fosdick, "POLISH, A FORTRAN Program to Edit FORTRAN Programs", Department of Computer Science, Technical Report No. CU-CS-050-74, University of Colorado, Boulder, Colorado, July 1974.

Table 1

machine-dependent quantity	function (argument)
standard input unit	IIMACH (1)
standard output unit	IIMACH (2)
standard punch unit	IIMACH (3)
standard error message unit	IIMACH (4)
number of bits per integer storage unit	IIMACH (5)
number of characters per integer storage unit	IIMACH (6)
base of an integer, a	IIMACH (7)
number of base a digits, s	IIMACH (8)
largest integer, a^s-1	IIMACH (9)
base of a real number, b	IIMACH (10)
number of single-precision real base b digits, t	IIMACH (11)
smallest single-precision real exponent, e_{min}	IIMACH (12)
largest single-precision real exponent, e_{max}	IIMACH (13)
number of double-precision base b digits, T	IIMACH (14)
smallest double-precision exponent, E_{min}	IIMACH (15)
largest double-precision exponent, E_{max}	IIMACH (16)
smallest positive real magnitude, $b^{e_{min}-1}$	R1MACH (1)
largest positive magnitude, $b^{e_{max}}(1-b^{-t})$	R1MACH (2)
smallest relative spacing between values, b^{-t}	R1MACH (3)
largest relative spacing between values, $b^{(1-t)}$	R1MACH (4)
$\log_{10}b$	R1MACH (5)
smallest positive double-precision magnitude, $b^{E_{min}-1}$	D1MACH (1)
largest positive double-precision magnitude, $b^{E_{max}}(1-b^{-T})$	D1MACH (2)
smallest relative spacing between double-precision values, b^{-T}	D1MACH (3)
largest relative spacing between double-precision values, $b^{(1-T)}$	D1MACH (4)
double-precision, $\log_{10}b$	D1MACH (5)

```
C
C-----------------------------------------------------------------------
C  FUNCTION:  I1MACH
C  THIS ROUTINE IS FROM THE PORT MATHEMATICAL SUBROUTINE LIBRARY
C  IT IS DESCRIBED IN THE BELL LABORATORIES COMPUTING SCIENCE
C  TECHNICAL REPORT #47 BY P.A. FOX, A.D. HALL AND N.L. SCHRYER
C-----------------------------------------------------------------------
C
      INTEGER FUNCTION I1MACH(I)
C
C  I/O UNIT NUMBERS.
C
C    I1MACH( 1) = THE STANDARD INPUT UNIT.
C
C    I1MACH( 2) = THE STANDARD OUTPUT UNIT.
C
C    I1MACH( 3) = THE STANDARD PUNCH UNIT.
C
C    I1MACH( 4) = THE STANDARD ERROR MESSAGE UNIT.
C
C  WORDS.
C
C    I1MACH( 5) = THE NUMBER OF BITS PER INTEGER STORAGE UNIT.
C
C    I1MACH( 6) = THE NUMBER OF CHARACTERS PER INTEGER STORAGE UNIT.
C
C  INTEGERS.
C
C    ASSUME INTEGERS ARE REPRESENTED IN THE S-DIGIT, BASE-A FORM
C
C                SIGN ( X(S-1)*A**(S-1) + ... + X(1)*A + X(0) )
C
C                WHERE 0 .LE. X(I) .LT. A FOR I=0,...,S-1.
C
C    I1MACH( 7) = A, THE BASE.
C
C    I1MACH( 8) = S, THE NUMBER OF BASE-A DIGITS.
C
C    I1MACH( 9) = A**S - 1, THE LARGEST MAGNITUDE.
C
C  FLOATING-POINT NUMBERS.
C
C    ASSUME FLOATING-POINT NUMBERS ARE REPRESENTED IN THE T-DIGIT,
C    BASE-B FORM
C
C       .        SIGN (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C                WHERE 0 .LE. X(I) .LT. B FOR I=1,...,T,
C                0 .LT. X(1), AND EMIN .LE. E .LE. EMAX.
C
C    I1MACH(10) = B, THE BASE.
C
C  SINGLE-PRECISION
C
C    I1MACH(11) = T, THE NUMBER OF BASE-B DIGITS.
C
C    I1MACH(12) = EMIN, THE SMALLEST EXPONENT E.
C
C    I1MACH(13) = EMAX, THE LARGEST EXPONENT E.
C
C  DOUBLE-PRECISION
C
C    I1MACH(14) = T, THE NUMBER OF BASE-B DIGITS.
C
C    I1MACH(15) = EMIN, THE SMALLEST EXPONENT E.
C
C    I1MACH(16) = EMAX, THE LARGEST EXPONENT E.
C
C  TO ALTER THIS FUNCTION FOR A PARTICULAR ENVIRONMENT,
C  THE DESIRED SET OF DATA STATEMENTS SHOULD BE ACTIVATED BY
C  REMOVING THE C FROM COLUMN 1.  ALSO, THE VALUES OF
C  I1MACH(1) - I1MACH(4) SHOULD BE CHECKED FOR CONSISTENCY
C  WITH THE LOCAL OPERATING SYSTEM.
C
      INTEGER IMACH(16),OUTPUT
C
      EQUIVALENCE (IMACH(4),OUTPUT)
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM.
C
C     DATA IMACH( 1) /    7 /
C     DATA IMACH( 2) /    2 /
C     DATA IMACH( 3) /    2 /
C     DATA IMACH( 4) /    2 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   33 /
C     DATA IMACH( 9) / Z1FFFFFFFF /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -256 /
C     DATA IMACH(13) /  255 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) / -256 /
C     DATA IMACH(16) /  255 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   48 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /    8 /
C     DATA IMACH(11) /   13 /
C     DATA IMACH(12) /  -50 /
C     DATA IMACH(13) /   76 /
C     DATA IMACH(14) /   26 /
C     DATA IMACH(15) /  -50 /
C     DATA IMACH(16) /   76 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   48 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /    8 /
C     DATA IMACH(11) /   13 /
C     DATA IMACH(12) /  -50 /
```

```
C     DATA IMACH(13) /  76 /
C     DATA IMACH(14) /  26 /
C     DATA IMACH(15) / -32754 /
C     DATA IMACH(16) /  32780 /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   60 /
C     DATA IMACH( 6) /   10 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   48 /
C     DATA IMACH( 9) / 00007777777777777777B /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   48 /
C     DATA IMACH(12) / -974 /
C     DATA IMACH(13) / 1070 /
C     DATA IMACH(14) /   96 /
C     DATA IMACH(15) / -927 /
C     DATA IMACH(16) / 1070 /
C
C     MACHINE CONSTANTS FOR THE CRAY 1
C
C     DATA IMACH( 1) /   100 /
C     DATA IMACH( 2) /   101 /
C     DATA IMACH( 3) /   102 /
C     DATA IMACH( 4) /   101 /
C     DATA IMACH( 5) /    64 /
C     DATA IMACH( 6) /     8 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    63 /
C     DATA IMACH( 9) /  777777777777777777777B /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    47 /
C     DATA IMACH(12) / -8192 /
C     DATA IMACH(13) /  8190 /
C     DATA IMACH(14) /    95 /
C     DATA IMACH(15) / -8192 /
C     DATA IMACH(16) /  8190 /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     DATA IMACH( 1) /   11 /
C     DATA IMACH( 2) /   12 /
C     DATA IMACH( 3) /    8 /
C     DATA IMACH( 4) /   10 /
C     DATA IMACH( 5) /   16 /
C     DATA IMACH( 6) /    2 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   15 /
C     DATA IMACH( 9) /32767 /
C     DATA IMACH(10) /   16 /
C     DATA IMACH(11) /    6 /
C     DATA IMACH(12) /  -64 /
C     DATA IMACH(13) /   63 /
C     DATA IMACH(14) /   14 /
C     DATA IMACH(15) /  -64 /
C     DATA IMACH(16) /   63 /
C
C     MACHINE CONSTANTS FOR THE HARRIS SLASH 6 AND SLASH 7
C
C     DATA IMACH( 1) /       5 /
```



```
C     DATA IMACH( 2) /       6 /
C     DATA IMACH( 3) /       0 /
C     DATA IMACH( 4) /       6 /
C     DATA IMACH( 5) /      24 /
C     DATA IMACH( 6) /       3 /
C     DATA IMACH( 7) /       2 /
C     DATA IMACH( 8) /      23 /
C     DATA IMACH( 9) / 8388607 /
C     DATA IMACH(10) /       2 /
C     DATA IMACH(11) /      23 /
C     DATA IMACH(12) /    -127 /
C     DATA IMACH(13) /     127 /
C     DATA IMACH(14) /      38 /
C     DATA IMACH(15) /    -127 /
C     DATA IMACH(16) /     127 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /   43 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   63 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9 AND THE SEL SYSTEMS 85/86.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / Z7FFFFFFF /
C     DATA IMACH(10) /   16 /
C     DATA IMACH(11) /    6 /
C     DATA IMACH(12) /  -64 /
C     DATA IMACH(13) /   63 /
C     DATA IMACH(14) /   14 /
C     DATA IMACH(15) /  -64 /
C     DATA IMACH(16) /   63 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR).
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
```

```
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   54 /
C     DATA IMACH(15) / -101 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR).
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   62 /
C     DATA IMACH(15) / -128 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     16-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   16 /
C     DATA IMACH( 6) /    2 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   15 /
C     DATA IMACH( 9) / 32767 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES.
C
C     NOTE THAT THE PUNCH UNIT, I1MACH(3), HAS BEEN SET TO 7
C     WHICH IS APPROPRIATE FOR THE UNIVAC-FOR SYSTEM.
C     IF YOU HAVE THE UNIVAC-FTN SYSTEM, SET IT TO 1.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) /-1024 /
C     DATA IMACH(16) / 1023 /
C
C     MACHINE CONSTANTS FOR THE VAX-11 WITH
C     FORTRAN IV-PLUS COMPILER
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
      IF (I .LT. 1  .OR.  I .GT. 16) GO TO 10
C
      I1MACH=IMACH(I)
      RETURN
C
 10   WRITE(OUTPUT,9000)
 9000 FORMAT(39H1ERROR    1 IN I1MACH - I OUT OF BOUNDS)
C
      STOP
C
      END
```

```
C
C-----------------------------------------------------------------------
C  FUNCTION:  R1MACH
C  THIS ROUTINE IS FROM THE PORT MATHEMATICAL SUBROUTINE LIBRARY
C  IT IS DESCRIBED IN THE BELL LABORATORIES COMPUTING SCIENCE
C  TECHNICAL REPORT #47 BY P.A. FOX, A.D. HALL AND N.L. SCHRYER
C  A MODIFICATION TO THE "I OUT OF BOUNDS" ERROR MESSAGE
C  HAS BEEN MADE BY C. A. MCGONEGAL - APRIL, 1978
C-----------------------------------------------------------------------
C
      REAL FUNCTION R1MACH(I)
C
C  SINGLE-PRECISION MACHINE CONSTANTS
C
C  R1MACH(1) = B**(EMIN-1), THE SMALLEST POSITIVE MAGNITUDE.
C
C  R1MACH(2) = B**EMAX*(1 - B**(-T)), THE LARGEST MAGNITUDE.
C
C  R1MACH(3) = B**(-T), THE SMALLEST RELATIVE SPACING.
C
C  R1MACH(4) = B**(1-T), THE LARGEST RELATIVE SPACING.
C
C  R1MACH(5) = LOG10(B)
C
C  TO ALTER THIS FUNCTION FOR A PARTICULAR ENVIRONMENT,
C  THE DESIRED SET OF DATA STATEMENTS SHOULD BE ACTIVATED BY
C  REMOVING THE C FROM COLUMN 1.
C
C  WHERE POSSIBLE, OCTAL OR HEXADECIMAL CONSTANTS HAVE BEEN USED
C  TO SPECIFY THE CONSTANTS EXACTLY WHICH HAS IN SOME CASES
C  REQUIRED THE USE OF EQUIVALENT INTEGER ARRAYS.
C
      INTEGER SMALL(2)
      INTEGER LARGE(2)
      INTEGER RIGHT(2)
      INTEGER DIVER(2)
      INTEGER LOG10(2)
C
      REAL RMACH(5)
C
      EQUIVALENCE (RMACH(1),SMALL(1))
      EQUIVALENCE (RMACH(2),LARGE(1))
      EQUIVALENCE (RMACH(3),RIGHT(1))
      EQUIVALENCE (RMACH(4),DIVER(1))
      EQUIVALENCE (RMACH(5),LOG10(1))
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM.
C
C     DATA RMACH(1) / Z400800000 /
C     DATA RMACH(2) / Z5FFFFFFFF /
C     DATA RMACH(3) / Z4E9800000 /
C     DATA RMACH(4) / Z4EA800000 /
C     DATA RMACH(5) / Z500E730E8 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700/6700/7700 SYSTEMS.
C
C     DATA RMACH(1) / O1771000000000000 /
C     DATA RMACH(2) / O0777777777777777 /
C     DATA RMACH(3) / O1311000000000000 /
C     DATA RMACH(4) / O1301000000000000 /
C     DATA RMACH(5) / O1157163034761675 /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C     DATA RMACH(1) / 00014000000000000000B /
C     DATA RMACH(2) / 37767777777777777777B /
C     DATA RMACH(3) / 16404000000000000000B /
C     DATA RMACH(4) / 16414000000000000000B /
C     DATA RMACH(5) / 17164642023241175720B /
C
C     MACHINE CONSTANTS FOR THE CRAY 1
C
C     DATA RMACH(1) / 200040000000000000000B /
C     DATA RMACH(2) / 577677777777777777776B /
C     DATA RMACH(3) / 377224000000000000000B /
C     DATA RMACH(4) / 377234000000000000000B /
C     DATA RMACH(5) / 377774642023241175720B /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C     STATIC RMACH(5)
C
C     DATA SMALL/20K,0/,LARGE/77777K,177777K/
C     DATA RIGHT/35420K,0/,DIVER/36020K,0/
C     DATA LOG10/40423K,42023K/
C
C     MACHINE CONSTANTS FOR THE HARRIS SLASH 6 AND SLASH 7
C
C     DATA SMALL(1),SMALL(2) / '20000000, '00000201 /
C     DATA LARGE(1),LARGE(2) / '37777777, '00000177 /
C     DATA RIGHT(1),RIGHT(2) / '20000000, '00000352 /
C     DATA DIVER(1),DIVER(2) / '20000000, '00000353 /
C     DATA LOG10(1),LOG10(2) / '23210115, '00000377 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES.
C
C     DATA RMACH(1) / O402400000000 /
C     DATA RMACH(2) / O376777777777 /
C     DATA RMACH(3) / O714400000000 /
C     DATA RMACH(4) / O716400000000 /
C     DATA RMACH(5) / O776464202324 /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9 AND THE SEL SYSTEMS 85/86.
C
C     DATA RMACH(1) / Z00100000 /
C     DATA RMACH(2) / Z7FFFFFFF /
C     DATA RMACH(3) / Z3B100000 /
C     DATA RMACH(4) / Z3C100000 /
C     DATA RMACH(5) / Z41134413 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA OR KI PROCESSOR).
C
C     DATA RMACH(1) / "000400000000 /
C     DATA RMACH(2) / "377777777777 /
C     DATA RMACH(3) / "146400000000 /
C     DATA RMACH(4) / "147400000000 /
C     DATA RMACH(5) / "177464202324 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN'S SUPPORTING
C     32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1) /    8388608 /
C     DATA LARGE(1) / 2147483647 /
C     DATA RIGHT(1) /  880803840 /
C     DATA DIVER(1) /  889192448 /
C     DATA LOG10(1) / 1067065499 /
C
C     DATA RMACH(1) / O00040000000 /
```

```
C     DATA RMACH(2) / O17777777777 /
C     DATA RMACH(3) / O06440000000 /
C     DATA RMACH(4) / O06500000000 /
C     DATA RMACH(5) / O07746420233 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN'S SUPPORTING
C     16-BIT INTEGERS  (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1),SMALL(2) /   128,     0 /
C     DATA LARGE(1),LARGE(2) / 32767,    -1 /
C     DATA RIGHT(1),RIGHT(2) / 13440,     0 /
C     DATA DIVER(1),DIVER(2) / 13568,     0 /
C     DATA LOG10(1),LOG10(2) / 16282,  8347 /
C
C     DATA SMALL(1),SMALL(2) / O000200, O000000 /
C     DATA LARGE(1),LARGE(2) / O077777, O177777 /
C     DATA RIGHT(1),RIGHT(2) / O032200, O000000 /
C     DATA DIVER(1),DIVER(2) / O032400, O000000 /
C     DATA LOG10(1),LOG10(2) / O037632, O020233 /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES.
C
C     DATA RMACH(1) / O000400000000 /
C     DATA RMACH(2) / O377777777777 /
C     DATA RMACH(3) / O146400000000 /
C     DATA RMACH(4) / O147400000000 /
C     DATA RMACH(5) / O177464202324 /
C
C     MACHINE CONSTANTS FOR THE VAX-11 WITH
C     FORTRAN IV-PLUS COMPILER
C
C     DATA RMACH(1) / Z00000080 /
C     DATA RMACH(2) / ZFFFF7FFF /
C     DATA RMACH(3) / Z00003480 /
C     DATA RMACH(4) / Z00003500 /
C     DATA RMACH(5) / Z209B3F9A /
C
      IF (I .LT. 1  .OR.  I .GT. 5) GOTO 100
C
      R1MACH = RMACH(I)
      RETURN
C
  100 IWUNIT = I1MACH(4)
      WRITE(IWUNIT, 99)
   99 FORMAT(24HR1MACH - I OUT OF BOUNDS)
      STOP
      END
```

```
C
C-----------------------------------------------------------------------
C  FUNCTION:  D1MACH
C  THIS ROUTINE IS FROM THE PORT MATHEMATICAL SUBROUTINE LIBRARY
C  IT IS DESCRIBED IN THE BELL LABORATORIES COMPUTING SCIENCE
C  TECHNICAL REPORT #47 BY P.A. FOX, A.D. HALL AND N.L. SCHRYER
C  A MODIFICATION TO THE "I OUT OF BOUNDS" ERROR MESSAGE
C  HAS BEEN MADE BY C. A. MCGONEGAL - APRIL, 1978
C-----------------------------------------------------------------------
C
      DOUBLE PRECISION FUNCTION D1MACH(I)
C
C  DOUBLE-PRECISION MACHINE CONSTANTS
C
C  D1MACH( 1) = B**(EMIN-1), THE SMALLEST POSITIVE MAGNITUDE.
C
C  D1MACH( 2) = B**EMAX*(1 - B**(-T)), THE LARGEST MAGNITUDE.
C
C  D1MACH( 3) = B**(-T), THE SMALLEST RELATIVE SPACING.
C
C  D1MACH( 4) = B**(1-T), THE LARGEST RELATIVE SPACING.
C
C  D1MACH( 5) = LOG10(B)
C
C  TO ALTER THIS FUNCTION FOR A PARTICULAR ENVIRONMENT,
C  THE DESIRED SET OF DATA STATEMENTS SHOULD BE ACTIVATED BY
C  REMOVING THE C FROM COLUMN 1.
C
C  WHERE POSSIBLE, OCTAL OR HEXADECIMAL CONSTANTS HAVE BEEN USED
C  TO SPECIFY THE CONSTANTS EXACTLY WHICH HAS IN SOME CASES
C  REQUIRED THE USE OF EQUIVALENT INTEGER ARRAYS.
C
      INTEGER SMALL(4)
      INTEGER LARGE(4)
      INTEGER RIGHT(4)
      INTEGER DIVER(4)
      INTEGER LOG10(4)
C
      DOUBLE PRECISION DMACH(5)
C
      EQUIVALENCE (DMACH(1),SMALL(1))
      EQUIVALENCE (DMACH(2),LARGE(1))
      EQUIVALENCE (DMACH(3),RIGHT(1))
      EQUIVALENCE (DMACH(4),DIVER(1))
      EQUIVALENCE (DMACH(5),LOG10(1))
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM.
C
C     DATA SMALL(1) / ZC00800000 /
C     DATA SMALL(2) / Z000000000 /
C
C     DATA LARGE(1) / ZDFFFFFFFF /
C     DATA LARGE(2) / ZFFFFFFFFF /
C
C     DATA RIGHT(1) / ZCC5800000 /
C     DATA RIGHT(2) / Z000000000 /
C
C     DATA DIVER(1) / ZCC6800000 /
C     DATA DIVER(2) / Z000000000 /
C
C     DATA LOG10(1) / ZD00E730E7 /
C     DATA LOG10(2) / ZC77800DC0 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM.
C
C     DATA SMALL(1) / O1771000000000000 /
C     DATA SMALL(2) / O0000000000000000 /
C
C     DATA LARGE(1) / O0777777777777777 /
C     DATA LARGE(2) / O0000777777777777 /
C
C     DATA RIGHT(1) / O1461000000000000 /
C     DATA RIGHT(2) / O0000000000000000 /
C
C     DATA DIVER(1) / O1451000000000000 /
C     DATA DIVER(2) / O0000000000000000 /
C
C     DATA LOG10(1) / O1157163034761674 /
C     DATA LOG10(2) / O0006677466732724 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS.
C
C     DATA SMALL(1) / O1771000000000000 /
C     DATA SMALL(2) / O7770000000000000 /
C
C     DATA LARGE(1) / O0777777777777777 /
C     DATA LARGE(2) / O7777777777777777 /
C
C     DATA RIGHT(1) / O1461000000000000 /
C     DATA RIGHT(2) / O0000000000000000 /
C
C     DATA DIVER(1) / O1451000000000000 /
C     DATA DIVER(2) / O0000000000000000 /
C
C     DATA LOG10(1) / O1157163034761674 /
C     DATA LOG10(2) / O0006677466732724 /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C     DATA SMALL(1) / 00604000000000000000B /
C     DATA SMALL(2) / 00000000000000000000B /
C
C     DATA LARGE(1) / 37767777777777777777B /
C     DATA LARGE(2) / 37167777777777777777B /
C
C     DATA RIGHT(1) / 15604000000000000000B /
C     DATA RIGHT(2) / 15000000000000000000B /
C
C     DATA DIVER(1) / 15614000000000000000B /
C     DATA DIVER(2) / 15010000000000000000B /
C
C     DATA LOG10(1) / 17164642023241175717B /
C     DATA LOG10(2) / 16367571421742254654B /
C
C     MACHINE CONSTANTS FOR THE CRAY 1
C
C     DATA SMALL(1) / 20004000000000000000B /
C     DATA SMALL(2) / 00000000000000000000B /
C
C     DATA LARGE(1) / 57767777777777777777B /
C     DATA LARGE(2) / 00007777777777777776B /
C
C     DATA RIGHT(1) / 37642400000000000000B /
C     DATA RIGHT(2) / 00000000000000000000B /
C
C     DATA DIVER(1) / 37643400000000000000B /
C     DATA DIVER(2) / 00000000000000000000B /
C
C     DATA LOG10(1) / 37777642023241175717B /
C     DATA LOG10(2) / 00000757142174225465 4B /
```

```
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C     STATIC DMACH(5)
C
      DATA SMALL/20K,3*0/,LARGE/77777K,3*177777K/
      DATA RIGHT/31420K,3*0/,DIVER/32020K,3*0/
      DATA LOG10/40423K,42023K,50237K,74776K/
C
C     MACHINE CONSTANTS FOR THE HARRIS SLASH 6 AND SLASH 7
C
      DATA SMALL(1),SMALL(2) / '20000000, '00000201 /
      DATA LARGE(1),LARGE(2) / '37777777, '37777577 /
      DATA RIGHT(1),RIGHT(2) / '20000000, '00000333 /
      DATA DIVER(1),DIVER(2) / '20000000, '00000334 /
      DATA LOG10(1),LOG10(2) / '23210115, '10237777 /
C
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES.
C
      DATA SMALL(1),SMALL(2) / O402400000000, O000000000000 /
      DATA LARGE(1),LARGE(2) / O376777777777, O777777777777 /
      DATA RIGHT(1),RIGHT(2) / O604400000000, O000000000000 /
      DATA DIVER(1),DIVER(2) / O606400000000, O000000000000 /
      DATA LOG10(1),LOG10(2) / O776464202324, O117571775714 /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9 AND THE SEL SYSTEMS 85/86.
C
      DATA SMALL(1),SMALL(2) / Z00100000, Z00000000 /
      DATA LARGE(1),LARGE(2) / Z7FFFFFFF, ZFFFFFFFF /
      DATA RIGHT(1),RIGHT(2) / Z33100000, Z00000000 /
      DATA DIVER(1),DIVER(2) / Z34100000, Z00000000 /
      DATA LOG10(1),LOG10(2) / Z41134413, Z509F79FF /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR).
C
      DATA SMALL(1),SMALL(2) / "033400000000, "000000000000 /
      DATA LARGE(1),LARGE(2) / "377777777777, "344777777777 /
      DATA RIGHT(1),RIGHT(2) / "113400000000, "000000000000 /
      DATA DIVER(1),DIVER(2) / "114400000000, "000000000000 /
      DATA LOG10(1),LOG10(2) / "177464202324, "144117571776 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR).
C
      DATA SMALL(1),SMALL(2) / "000400000000, "000000000000 /
      DATA LARGE(1),LARGE(2) / "377777777777, "377777777777 /
      DATA RIGHT(1),RIGHT(2) / "103400000000, "000000000000 /
      DATA DIVER(1),DIVER(2) / "104400000000, "000000000000 /
      DATA LOG10(1),LOG10(2) / "177464202324, "476747767461 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN'S SUPPORTING
C     32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
      DATA SMALL(1),SMALL(2) /    8388608,           0 /
      DATA LARGE(1),LARGE(2) / 2147483647,          -1 /
      DATA RIGHT(1),RIGHT(2) /  612368384,           0 /
      DATA DIVER(1),DIVER(2) /  620756992,           0 /
      DATA LOG10(1),LOG10(2) / 1067065498, -2063872008 /
C
      DATA SMALL(1),SMALL(2) / O00040000000, O00000000000 /
      DATA LARGE(1),LARGE(2) / O17777777777, O37777777777 /
      DATA RIGHT(1),RIGHT(2) / O04440000000, O00000000000 /
      DATA DIVER(1),DIVER(2) / O04500000000, O00000000000 /
      DATA LOG10(1),LOG10(2) / O07746420232, O20476747770 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN'S SUPPORTING
C     16-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1),SMALL(2) /    128,       0 /
C     DATA SMALL(3),SMALL(4) /      0,       0 /
C
C     DATA LARGE(1),LARGE(2) /  32767,      -1 /
C     DATA LARGE(3),LARGE(4) /     -1,      -1 /
C
C     DATA RIGHT(1),RIGHT(2) /   9344,       0 /
C     DATA RIGHT(3),RIGHT(4) /      0,       0 /
C
C     DATA DIVER(1),DIVER(2) /   9472,       0 /
C     DATA DIVER(3),DIVER(4) /      0,       0 /
C
C     DATA LOG10(1),LOG10(2) /  16282,    8346 /
C     DATA LOG10(3),LOG10(4) / -31493,  -12296 /
C
C     DATA SMALL(1),SMALL(2) / O000200, O000000 /
C     DATA SMALL(3),SMALL(4) / O000000, O000000 /
C
C     DATA LARGE(1),LARGE(2) / O077777, O177777 /
C     DATA LARGE(3),LARGE(4) / O177777, O177777 /
C
C     DATA RIGHT(1),RIGHT(2) / O022200, O000000 /
C     DATA RIGHT(3),RIGHT(4) / O000000, O000000 /
C
C     DATA DIVER(1),DIVER(2) / O022400, O000000 /
C     DATA DIVER(3),DIVER(4) / O000000, O000000 /
C
C     DATA LOG10(1),LOG10(2) / O037632, O020232 /
C     DATA LOG10(3),LOG10(4) / O102373, O147770 /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES.
C
C     DATA SMALL(1),SMALL(2) / O000040000000, O000000000000 /
C     DATA LARGE(1),LARGE(2) / O377777777777, O777777777777 /
C     DATA RIGHT(1),RIGHT(2) / O170540000000, O000000000000 /
C     DATA DIVER(1),DIVER(2) / O170640000000, O000000000000 /
C     DATA LOG10(1),LOG10(2) / O177746420232, O411757177572 /
C
C     MACHINE CONSTANTS FOR THE VAX-11 WITH
C     FORTRAN IV-PLUS COMPILER
C
C     DATA SMALL(1),SMALL(2) / Z00000080, Z00000000 /
C     DATA LARGE(1),LARGE(2) / ZFFFF7FFF, ZFFFFFFFF /
C     DATA RIGHT(1),RIGHT(2) / Z00002480, Z00000000 /
C     DATA DIVER(1),DIVER(2) / Z00002500, Z00000000 /
C     DATA LOG10(1),LOG10(2) / Z209A3F9A, ZCFFA84FB /
C
      IF (I .LT. 1  .OR.  I .GT. 5) GOTO 100
C
      D1MACH = DMACH(I)
      RETURN
C
  100 IWUNIT = I1MACH(4)
      WRITE (IWUNIT, 99)
   99 FORMAT (24H0D1MACH - I OUT OF BOUNDS)
      STOP
      END
```