Behavior Research Methods & Instrumentation
1980, Vol. 12 (1), 75

FREEIN: Flexible free-form input subroutine

ANDREW J. MACKINNON
University of Melbourne, Parkville, Victoria 3052, Australia

Difficulties are often experienced in writing interactive programs for inexperienced users. When a program requires input from a terminal, the most desirable means is in free format. That is, the data items are required only to be in the correct order, separated by specified delimiters, with no restrictions on the location of the items on the input line. Users should not have to worry about type conversion; for example, 1 should be regarded as equivalent to 1.00 when a real number is required.

Many FORTRAN compilers support some form of list-directed input. However, this provides only a partial solution in most cases. The error messages produced by many of these routines are obscure, often merely stating that an error has occurred but not specifying its nature or location. More detailed information is vital for inexperienced computer users. In addition, alphanumeric variables (literals) must usually be enclosed in quotation marks. Further, these systems support only standard FORTRAN input lists. It is not possible for the program user to specify that he wishes to use default values, for example, or that the input requirements are not understood.

FREEIN attempts to overcome these difficulties and to give the programmer a greater degree of control over the input.

Input. FREEIN decodes data items separated by one or more blanks or commas. The position of items on the input line is not important. Numbers may be input with or without a decimal point and may contain an optional exponent (e.g., FREEIN recognizes 10, 10.0, and 1E1 as being identical). Literals may be enclosed in quotes and contain any character except the quote character itself. However, any literal beginning with an alphabetic character and not containing a blank or quote is not required to be enclosed in quotes.

The programmer specifies the length of the line to be examined and whether numeric variables only, alphanumeric literals only, or both are allowable input. In specifying the number of items to be input, there are three options. A fixed number of items may be specified. Excess items will be ignored, and, if fewer than the specified number of items are found, a prompt for "MORE" will be issued. Alternatively, FREEIN can count the number of items input and return this value to the calling program along with the data. The third option extends the second by allowing the user to input a variable number of values or an empty line to indicate that default values are required.

Output. Data are returned to the calling program in a floating-point vector. The maximum length of literal variables is determined by the number of characters per floating-point word, usually 8 or 10 characters.

Error Processing. The programmer may select one of three levels of error processing. The first produces no message, except to prompt for corrected input. Alternatively, an error message stating the nature of the error may be printed. The most informative level includes a diagnostic message and prints the input line showing where the error occurred.

FREEIN also allows the programmer to determine what action should occur after an error is detected. In normal "interactive" mode, a prompt to input the information "AGAIN" is issued. However, if desired, FREEIN will return to the calling program with the error condition flagged. This mode of processing is useful if FREEIN is reading data from a file.

Help Processing. FREEIN recognizes the character "?" as a request for help if it is the first character on the line. If the programmer does not wish the program to take any special action as a result of this request, FREEIN will print a message:

SORRY, NO HELP AVAILABLE.

Alternatively it will return to the calling program with the request flagged. Any desired procedure may follow (e.g., a call to a HELP subroutine).

Computer Requirements. As far as possible, FREEIN has been programmed in ANSI FORTRAN. Compiler-specific procedures for bit manipulation are, however, required for literal conversion. In addition, the operation of the subroutine is dependent on the character code. FREEIN has been programmed for the CYBER 70 series computers using FTN 4.6 and for the DEC PDP-11 series computers using the FORTRAN V2.01 compiler. The CYBER version uses a 6-bit ASCII subset character set. The PDP-11 version should be suitable for machines using the ASCII character code as long as they support the BYTE or LOGICAL*1 data type for bit manipulation. The calling sequences for each version and its functioning are identical, except that the PDP version performs two additional tasks: It converts lowercase alphabetics to uppercase, and it detects the presence of control characters and treats their presence as an error condition.

FREEIN is not as efficient as compiler-provided input routines in execution speed or space. The CYBER version requires 3,200 words. This includes the processing routine itself and a separate error processing routine.

Availability. A listing of either version of the program is available free of charge from A. J. Mackinnon, Department of Psychology, University of Melbourne, Parkville, Victoria 3052, Australia. The program contains approximately 500 lines. Of these, approximately half are comment lines describing the use of the subroutine.

(Accepted for publication September 4, 1979.)

0005-7878/80/010075-01$00.35/0