```
REM Basic Program No. 1
REM Note: "Comments" in basic begin with the characters "REM"
REM Purpose: This program reads a number and computes the sum of
REM          the numbers from 1 through the number read.
REM Coded by S. G. Wright - 8/30/93
REM Revised by S. G. Wright - 8/31/93
REM ---------------------------------------------------------------------------
REM
REM Open the input and output files
REM
   OPEN "INPUT" FOR INPUT AS 5
   OPEN "OUTPUT" FOR OUTPUT AS 6
REM
REM Read the number being input.
REM
   INPUT #5, NUM
   PRINT #6, USING "& ###########"; "The number entered was: "; NUM
REM
REM Test if the number is negative.  If it is, output message.
REM Otherwise continue and compute the summation.
REM
   IF NUM < 0 THEN
      PRINT #6, USING "&"; "The number read was negative."
   ELSE
      ISUM = 0
      FOR I = 1 TO NUM
         ISUM = ISUM + I
      NEXT
      PRINT #6, USING "& #############"; "The summation is:"; ISUM
   END IF
REM
REM Close the files that were opened for input & output.
REM
   CLOSE #5
   CLOSE #6
   END
```

```
/* C Example Program No. 1                                          */
/* Note: "Comments" in C are bracketed with slash-asterisk          */
/*       and asterisk-slash.                                        */
/* Purpose: This program reads a number and computes the sum of     */
/*          the numbers from 1 through the number read.             */
/*          The number is read from an external file named INPUT.   */
/*          The output is written to an external file named OUTPUT. */
/* Hardware & Software: Written in Symantec's C 6.0 for the         */
/*                      Apple Macintosh.                            */
/* Coded by S. G. Wright - 8/31/93                                  */
/* Revised by S. G. Wright - 9/1/93                                 */
/*------------------------------------------------------------------*/

/* The following two items of "front-matter" are required for       */
/* almost any C program to have access to several basic functions   */
/* and input/output capabilities.  Other items may also need to be  */
/* "included" depending on functionality needed.                    */

#include <stdio.h>
#include <stdlib.h>

/* The program must be named "main" */

int main ()
{ /* This left brace signifies the beginning of the program instructions */

/* The following statements declare the names and the types of */
/* variables that are used in this program                     */

 FILE     *inputFile;
 FILE     *outputFile;
 int       I;
 int       ISUM;
 int       NUM;

/* Open the files for input and output */

 inputFile = fopen ("INPUT","r");
 outputFile = fopen ("OUTPUT","w");

/* Read & write to the output file the number being input */

 fscanf(inputFile,"%d",&NUM);
 fprintf(outputFile,"The number read was: %d\n",NUM);

/* Test if the number is negative.  If it is, output a message.  */
/* Otherwise, continue and compute the summation.                */

 if (NUM < 0)
  fprintf(outputFile,"The number read was negative\n",NUM);
 else
 { /* left brace begins a block of instructions */
  ISUM = 0;
  for (I = 1; I < (NUM + 1); I++)
   ISUM = ISUM + I;
  fprintf(outputFile,"The summation is: %d\n",ISUM);

 } /* right brace ends a block of instructions */

/* close the files that were opened for input & output */

 fclose(inputFile);
 fclose(outputFile);

} /* this right brace signifies the end of the program instructions */
```

```
C FORTRAN Example Program No. 1
C Note: "Comments" in FORTRAN begin with the character "C"
C       in column 1.
C Purpose: This program reads a number and computes the sum of
C          the numbers from 1 through the number read.
C          The number is read from an external file named INPUT.
C          The output is written to an external file named OUTPUT.
C Hardware & Software: Written in MicroSoft FORTRAN for the PC.
C Coded by S. G. Wright - 8/31/93
C Revised by S. G. Wright - 9/1/93
C---------------------------------------------------------------------
C
C     The program instructions begin with a "PROGRAM" statement.
C
      PROGRAM EXAMPLE1
C
C     Open the files for input and output.
C
      OPEN (5,FILE='INPUT')
      OPEN (6,FILE='OUTPUT')
C
C     Read & write to the output file the number being input
C
      READ (5,*) NUM
      WRITE (6,2000) NUM
 2000 FORMAT (1X,'The number read was:',I10)
C
C     Test if the number is negative.  If it is, output a message.
C     Otherwise, continue and compute the summation.
C
      IF (NUM .LT. 0) THEN
         WRITE (6,'(1X,''The number read was negative'')')
      ELSE
         ISUM = 0
         DO 100 I = 1,NUM
            ISUM = ISUM + I
  100    CONTINUE
         WRITE (6,2020) ISUM
 2020    FORMAT (1X,'The summation is:',I10)
      ENDIF
C
      END
```

```
{ Pascal Example Program No. 1                                              }
{ Note: "Comments" in Pascal are bracketed with "braces"                    }
{ Purpose: This program reads a number and computes the sum of              }
{          the numbers from 1 through the number read.                      }
{          The number is read from an external file named INPUT.            }
{          The output is written to an external fiel named OUTPUT.          }
{ Hardware & Software: Written in Borland Turbo Pascal 7.0 on a PC          }
{ Coded by S. G. Wright - 8/30/93                                           }
{ Revised by S. G. Wright - 9/2/93                                          }
{---------------------------------------------------------------------------}

{ The program instructions begin with the "program" statement }

 program EXAMPLE1;

{ The following statements, beginning with the keyword "var" declare the }
{ names and types of variables that are used in this program.            }

 var

    inputFile  : Text;
    outputFile : Text;
    i          : integer;
    isum       : integer;
    num        : integer;

{ The program instructions start with the keyword "begin" }

 begin

{ Open the input and output files. }

    assign(inputFile,'INPUT');
    assign(outputFile,'OUTPUT');
    reset(inputFile);
    rewrite(outputFile);

{ Read & write to the output file the number being input. }

    readln(inputFile, num);
    writeln(outputFile,'The number read was:', num : 5);

{ Test if the number is negative.  If it is, output a message. }
{ Otherwise, continue and compute the summation.               }

    if (num < 0) then
       writeln(outputFile,'The number read was negative.')
    else
       begin { A block of statements starts with the keyword "begin" }
          isum := 0;
          for i := 1 to num do { beginning of "for" loop }
           isum := isum + i;   { end of "for" loop       }
          writeln(outputFile,'The summation is:',isum);
       end; { A block of statements ends with the keyword "end" }

{ Close the files that were opened for input & output. }

    close(inputFile);
    close(outputFile);

 end. { The program ends with an "end" statement. }
```