Read chapter 3.

For the following two problems, you just have to write the relevant code to accomplish the given task. There is no need to write a complete program.

1. Write code which prompts the user for a number $x$ and an exponent $y$, both of which are real numbers. The goal is to print $x^y$. However, if $x$ is a negative number, it is an error in FORTRAN to calculate $x^y$ if the exponent is not an integer or an integer expression\(^1\). Nevertheless, there really shouldn’t be a problem if the user entered an integer value for $y$—the fact that we stored the exponent as a real variable is our problem. So, your code should check if the exponent is actually an integer (look up the INT () and/or AINT () functions in Table 6.1). If $y$ contains an integer value, the user should obtain a numeric result regardless of the sign of $x$. If the user enters a negative $x$ and a non-integer $y$, print an error message that says the value can’t be computed. And, of course, if $x$ is positive, $x^y$ should be displayed regardless of $y$.

```fortran
if (x .lt. 0.0) then
    if (y .ne. aint(y)) then
        print *, 'Can’t calculate exponential with given values.’
    else
        print *, x**int(y)
    endif
else
    print *, x**y
endif
```

(continued on back)

\(^1\)The g77 compiler will actually forgive this error. Contrary to what it says in the book on page 56, our compiler will allow you to write things such as \((-2.0) ** 2.0\). However, for the sake of this problem we will not take advantage of this feature.
2. (Taken from problem 10, page 158 of the text.) Print the message “LEAP YEAR” if the integer variable \textbf{YEAR} is the number of a leap year. A leap year is a multiple of 4 and if it is a multiple of 100 it must also be a multiple of 400. (So, for example, 1800 and 1900 would not be leap years because they are multiples of 100 but not 400. If you want to check if a given year is a leap year, you can enter the command “\texttt{cal YYYY}” at the command-line prompt, where \texttt{YYYY} is the given year. See how many days February has. \texttt{cal 1752} is interesting—the month of September is not a bug.)

You’ll want to look up the \texttt{MOD()} function (again, Table 6.1 is the place to get a brief description of functions).

\begin{verbatim}
if (mod(year,4) .eq. 0 .and.
   & (mod(year,100) .eq. 0 .eqv. mod(year,400) .eq. 0))
   & print *, ’LEAP YEAR’

or

if (mod(year,4) .eq. 0 .and. mod(year,100) .ne. 0
   & .or. mod(year,400) .eq. 0)
   & print *, ’LEAP YEAR’

or, less elegantly and breaking things up to separate statements,

if (mod(year,4) .eq. 0 .and. (mod(year,100) .ne. 0)) then
   print *, ’LEAP YEAR’
else if ((mod(year,100) .eq. 0 .and. mod(year,400) .eq. 0)) then
   print *, ’LEAP YEAR’
else

Other variation are certainly possible.
\end{verbatim}