

Assignment 2: Areas of a rectangles

Due: Tuesday 10/22/09 (before class)

In this assignment, you will write a program that can take in series of dimensions, and print out the computed area in the format described below. The area of a rectangle is computed as: $area = length \times width$.

Your program will not just perform one computation per run, as we did in assignment 1. Instead, it will continue to ask the user for input and print out results until it gets an illegal input value, which will be used as a signal that the program should cease operation. In our case, any strictly negative number (zero is a valid dimension) will be a flag to stop execution.

It is **required** that your program immediately cease asking for input and doing area computations when the *first* entry is negative. Therefore, if the first dimension is given as negative, then the second will not be asked for, and in no case will the area computed from negative value(s) be printed. You should make your operation match the example runs given below exactly (note that of course your program must work with any input, not just the exact input shown below).

Your routine will print a header (only once) describing what is being done, and a footer indicating how many valid areas were computed (see below for the exact form of these header/footer lines). Notice that the area I/O is indented three spaces past the header/footer lines.

This whole assignment should be written in a single file, called `area_rect.f90`, which the TA will compile using on a UTSA Linux machine (eg. `main204`), using the command:

```
gfortran -Wall -g -o xarea area_rect.f90
```

At the top of the program, put a comment that identifies you and the assignment. You should e-mail the source code for `area_rect.f90` to `yhan@cs.utsa.edu` before class on the due date. Be sure to follow the required coding style.

You may want to tackle this problem in steps:

1. Get program working that computes only one area per execution, just like last assignment
2. Add a loop that causes the program to loop forever, and make sure things work (make sure you solve a new problem each time!). You can break out of an *infinite loop* by holding the **control/CTRL** key, and hitting `c` (control-C).
3. Add code so that loop exits on negative input, and avoids unnecessary print/read(s)
4. Add code to count valid areas computed
5. Fix format of all I/O, including header and footer

Collaboration: As I have given you an outline, and sample output, students should not collaborate in any way on this assignment. If you need debugging help, ask the TA or myself. Do not consult with anyone on detailed implementation issues.

In the following printouts, all output including and following ‘!’ is a comment added by me, that does not occur in the printout (so your program should not include such comments). Note that the header printout is 75 columns, so you’ll need to use the string line extension we covered in class to abide by the max 80 columns coding style rule!

Example run:

```
c2d>gfortran -Wall -g -o xarea area_rect.f90
c2d>./xarea
COMPUTING AREA FROM LENGTH AND WIDTH (enter a negative value to quit):
! NOTICE: blank line after header
    Enter the length:
2.0
    Enter the width:
4.0
Area =    8.000000

    Enter the length:
3.2
    Enter the width:
-1
! go directly to footer after neg dim
! NOTICE: blank line before footer
DONE.  NUMBER OF AREAS COMPUTED =          1

c2d>./xarea
COMPUTING AREA FROM LENGTH AND WIDTH (enter a negative value to quit):

    Enter the length:
0.0
    Enter the width:
2.1
Area =    0.000000

    Enter the length:
4.8
    Enter the width:
3.2
Area =   15.36000

    Enter the length:
8
    Enter the width:
2
Area =   16.00000

    Enter the length:
-8
! does not ask for width if len neg

DONE.  NUMBER OF AREAS COMPUTED =          3
```