

Best part that works (circle one when done): Part A Part B Part C Part D

Time: 50 minutes

This is a multi-section exam. You may not discuss it with anyone until Friday Sep 14 2012.

You may only use your homework, labs, paper-based notes, the official class notes from the web (and reference sites directly linked from the course website), or your textbook for the practicum. You may **not** receive help from anyone, or use any other resource on the web.

Write a C++ program called main.cpp that first prints out your name and alphacode. Your program is to then read a sequence of room measurements, separated by whitespace, from the standard input and output some results to the screen. Specific details are below.

Grading: A partially working solution that compiles and produces some subset of the correct output is far better (i.e. worth more points) than a solution that does not compile.

(30 pts) Part A – Read in a single room dimension and compute the number of square feet in the room.

- Each room dimension is made up of 3 elements: <unit> <length1> <length2>
- For now assume that the <unit> will always be the letter F (for feet). The input dimensions, <length1> and <length2>, will be whole numbers.
- Recall that square feet can be computed by multiplying length1 and length 2.
- For full credit, all parts **must** output, when finished, this message: Bye!
- Example run (where user enters **F 2 3**):

```

mid@ubuntu: ~/ic210/miniPract
File Edit View Terminal Help
mid@ubuntu:~/ic210/miniPract$ ./a.out
Welcome to MIDN W.T. Door's (12XXXX) program.
Enter room dimensions: F 2 3
Found a room with area 6
Bye!
mid@ubuntu:~/ic210/miniPract$
    
```

STOP – when finished with each part, save a backup copy of your working code, e.g. in the terminal do:
 cp main.cpp partA.cpp

(20 pts) Part B – Add the ability to handle dimensions in inches OR feet. Measurements in inches are represented by the letter I for the units. You will still calculate area in terms of square feet.

NOTE: for lengths in inches, square feet = <length1> * <length2> / 144

```

mid@ubuntu:~/ic210/miniPract$ ./a.out
Welcome to MIDN W.T. Door's (12XXXX) program.
Enter room dimensions: I 12 18
Found a room with area 1.5
Bye!
mid@ubuntu:~/ic210/miniPract$ ./a.out
Welcome to MIDN W.T. Door's (12XXXX) program.
Enter room dimensions: F 2 3
Found a room with area 6
Bye!
mid@ubuntu:~/ic210/miniPract$
    
```

STOP – once working, make a backup before continuing: cp main.cpp partB.cpp

(35 pts) Part C – Modify your program so that it is capable of reading in multiple room dimensions and compute the number of square feet in each room. When there are no more room dimensions, the user enters the letter X. Here is an example:

```
mid@ubuntu:~/ic210/miniPract$ ./a.out
Welcome to MIDN W.T. Door's (12XXXX) program.
Enter room dimensions: F 2 3   I 12 18   I 12 12   X
Found a room with area 6
Found a room with area 1.5
Found a room with area 1
Bye!
mid@ubuntu:~/ic210/miniPract$
```

- In this example, the user entered this: F 2 3 I 12 18 I 12 12 X (This represents 3 different rooms. The user included extra spaces to make it easier to see each room separately, but these spaces will not affect your program.)
- Note 1: you do **not** know in advance how many rooms there will be, but you may assume there will be at least one room.
- Note 2: the user will NOT enter anything else after entering the X. You will need to think carefully about how to handle this, but it is similar to problems we have considered before.
- Note 3: for full credit, your program **must** output the final Bye! Message when finished. Maximum of 20 points for this part without handling this correctly.

STOP – once working, make a backup before continuing: `cp main.cpp partC.cpp`

(15 pts) Part D – Also output the total square feet (sum of all the areas). Example:

```
mid@ubuntu:~/ic210/miniPract$ ./a.out
Welcome to MIDN W.T. Door's (12XXXX) program.
Enter room dimensions: F 2 3   I 12 18   I 12 12   X
Found a room with area 6
Found a room with area 1.5
Found a room with area 1
Total square feet is: 8.5
Bye!
mid@ubuntu:~/ic210/miniPract$
```

STOP – once working, make a backup before continuing: `cp main.cpp partD.cpp`

When finished:

1. Go to IC210 on Blackboard. Go to Syllabus, and click on “MiniPracticum”. Submit the `partX.cpp` file for the best part (A, B, C, or D) that actually works.
2. Turn in the following things, stapled in the following order:
 - a. This paper (the assignment). On other side of this paper, write your name and circle one option for “Best part that works” at top right.
 - b. Printout of your `partX.cpp` (for the best part that works).
 - c. Printout of a screenshot with your program running (for the best part that works). Use the same sample input as shown above.
 - d. If you started on one part but did not get it working, also print that out and turn it in so your instructor can see your progress.