

COP 2930 - Individual Programming Assignment #2

Due date: Please consult WebCourses for your due date/time

Objectives

1. To give students practice at typing in, compiling and running simple programs.
2. To learn how to read in input from the user.
3. To learn how to use assignment statements and arithmetic expressions to make calculations

Problem A: Biggest Yard (bigyard.py)

You must fence off an area for your dogs to play freely and you have a fixed amount of fencing. You have decided that you want the fenced off area to be in the shape of a rectangle with an integer number of feet for both dimensions. Write a program that outputs all of the possible dimensions of the fenced off area as well as the area of each of those possibilities.

Input Specification

Note: It is guaranteed that whoever uses your program will adhere to these specifications. This means that you do NOT have to check for them!

The amount of fencing will be a positive even integer in between 4 and 100, inclusive.

Output Specification

Write out a single line of the form

X by Y with area Z

where X is the smaller dimension, in feet, Y is the larger dimension in feet, and Z is the corresponding area, for every possible rectangle. Order the rectangles by increasing order of X. Also, the last line may have $X = Y$.

Output Sample

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Run #1

How much fencing do you have, in feet?

10

1 by 4 with area 4

2 by 3 with area 6

Sample Run #2

How much fencing do you have, in feet?

16

1 by 7 with area 7

2 by 6 with area 12

3 by 5 with area 15

4 by 4 with area 16

Problem B: Tennis Scorer (tennis.py)

In the game of tennis, a match is a series of sets and a set is a series of games. A typical match winner is the player who first wins 2 sets. (Thus, all typical matches last either 2 or 3 sets long.) To win a set, a player either has to be the first to win six games and also win by a margin of at least two games, or, if a set is tied at 6 - 6, then the winner of the next game wins the set 7 - 6. (If this confuses you, just ask me in class to explain further!)

Serena and Roger are playing a match.

Write a program that prompts the user to enter who won each game and then reads in each response and determines who won the match. Your program **MUST** end right after the last game of the match is entered and you must display both the winner and how many sets both players won.

Input Specification

Note: It is guaranteed that whoever uses your program will adhere to these specifications. This means that you do NOT have to check for them!

Every line of input will either be "SERENA" or "ROGER", depending on who won the corresponding game.

Output Specification

Produce a single line of output with the following format:

PLAYER won the match 2 sets to X sets.

where PLAYER is the winner of the match and X is the number of sets won by the loser.

Output Sample

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Run #1

Please enter the game winners, in sequence

ROGER

SERENA

ROGER

SERENA

SERENA

SERENA

ROGER

SERENA

ROGER

SERENA

SERENA

SERENA

SERENA

SERENA

SERENA

SERENA

SERENA won the match 2 sets to 0 sets.

Restrictions

Please IDLE 3.6 (or higher) to develop your program. Write each in a separate file with the names specified previously, **bigyard.py** and **tennis.py**.

Each of your programs should include a header comment with the following information: your name, course number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

- 1) Your correctness
- 2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.
- 3) Compatibility to IDLE.