

COP 3223 Program #3: Lotto Simulation + Multiplication Game

Now that we've learned loops and random numbers, we have much more flexibility. One important skill that we learned with random numbers is to run a simulation.

In this assignment you'll write two programs: `lotto.py` and `multgame.py`. Turn in BOTH files as attachments over WebCourses by the designated due date and time.

Arup's Lotto

In Arup's Lotto, two numbers, one from 1 - 30 and a second from 31 - 60 are chosen, with each choice independent from the other and each of the possible numbers is equally likely to be chosen. If you match one number you receive \$5 and if you match both numbers you receive \$500. It costs \$1 per play. Write a program that simulates playing Arup's Lotto one million times. At the end of the simulation, print out the following:

- 1) The number of times exactly one number was matched.
- 2) The number of times exactly two numbers were matched.
- 3) The amount of money lost after buying the million lotto tickets.

For your simulation, you'll choose one winning lotto combination. Then, you'll simulate randomly choosing a million tickets, tallying up the winnings of each ticket.

Note: If you write the simulation properly, there is virtually no chance that you'll make money.

Sample Program Run

```
You matched 1 number 64222 times.  
You matched 2 numbers 1107 times.  
You lost $125390.
```

Grading Note: Most of your grade will be determined by how accurately you've run the simulation as opposed to the output produced. Namely, if you manage to produce output similar to this but don't mimic the process described above, you could still lose many points.

Multiplication Game

Write a program that allows the user to practice multiplication. Ask the user how many problems they want and then give the user that many problems. Each number in the problem should be in between 1 and 10, inclusive. For each question, determine if the user got the question correct. At the end of the game print out the percentage of problems they got correctly, rounded to the nearest tenth of a percent. After each question, print out a message with one of the two following formats, depending on whether or not the user correctly answered the question (where X represents the correct answer to the question:

Correct!

Sorry, the answer is X.

Sample Program Run (User Input in Bold and Italics)

How many problems do you want?

6

What is 1×4 ?

4

Correct!

What is 6×10 ?

60

Correct!

What is 5×10 ?

50

Correct!

What is 10×2 ?

20

Correct!

What is 3×9 ?

24

Sorry, the answer is 27.

What is 5×7 ?

35

Correct!

You got 83.3% of the problems.