Assignment 3

Due: Feb 20, 2014 at 11:55PM.

You may discuss any of the assignments with your classmates (or anyone else) but all work for all assignments must be entirely your own. Any sharing or copying of assignments will be considered cheating.

Problem 1 (30 points)

Write a program that simulates rolling a six-sided dice multiple times and counts how many times each outcome occurs. Repeat your simulation three times for $n=100$, $n = 10000$, $n = 1000000$ times (all in a single program). The program should print three tables (one for each value of $n$) that for each possible roll state the percentage (as a decimal) of times that this number occurred. An example table is shown below.

<table>
<thead>
<tr>
<th>Rolled</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>0.18</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>0.16</td>
</tr>
<tr>
<td>6</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Hint: Your values should be getting closer to 0.166 as $n$ increases.

Problem 2 (35 points)

Write a program that prompts the user for an integer $n$ and then displays the following pattern to the screen:

```
1
2  1
4  2  1
8  4  2  1
16 8  4  2  1
32 16 8  4  2  1
...  
2^n  
...  1
```

Your program should work for $n=1,2,...,10$, and display an error message "Your number is too large or too small" if the user enters any other value. For valid $n$'s your program should output $n+1$ lines.

Hint: You should be using nested loops.

Problem 3 (35 points)

Write a program that reads in a sequence of integers, finds the largest, and counts how many times it occurs. Assume that the user enters a sequence of numbers separated by spaces and that the input ends with a zero.

A sample run of your program may look like this:
Enter your numbers:  4 53 -1 11 45 53 37 -91 9 35 7 0

The largest number is 53
It occurs 2 times.

Hint 1: You should maintain two variables max and count. max should store the largest number you have seen so far (right after you "see" the first number it contains that number). count should keep track of how many times max occurs.

Hint 2: You should use sentinel controlled execution for determining when to end the loop.

Extra challenge (do not submit, but we can discuss is in class and/or office hours): Rewrite your program to find the second largest number as well as the largest number.

Grading

Does the program compile? If not, you will loose all the points for that problem.

Is the program properly documented? (worth 30% of each problem)

Proper documentation includes:

- preamble with the name of the author, date of creation and brief description of the program;
- appropriately chosen variable names, i.e., descriptive names;
- comments inside the code describing steps need to be taken to accomplish the goal of the program;
- appropriate formatting, indentation and use of white space to make the code readable.

Remember that the code is read by humans and it should be easy to read for people who were not involved in its development.

Is the program well developed? (worth 40% of each problem) Make sure you create variables of appropriate types, use control statements (conditionals and loops) that are appropriate for the task, accomplish your task in a well designed and simple way (not a convoluted algorithm that happens to produce the correct output for some unknown reason). You should also design a friendly and informative user interface.

Is the program correct? (worth 30% of each problem). Make sure that your program produces valid results that follow the specification of the problem every time it is run. At this point you can assume a "well behaved user" who enters the type of data that you request. If the program is not completely correct, you get credit proportional to how well it is developed and how close you got it to the completely correct code.

What and how to submit?

You should submit three source code files combined into a single zip file to NYU Classes. Do not submit all the files that Eclipse creates, just the source code files that have .java extensions. Name your classes Problem1, Problem2 and Problem3 (this means that your files are going to be Problem1.java, Problem2.java and Problem3.java).