Submission Due: Apr./10/2023 23:59

Homework 1: A Lottery Simulator

Overview and Specification

Your goal in this homework is to construct a program that simulates the process of winning lottery tickets. The entire procedure of replicating a lottery round is as follows:

1. Accept a natural number N, M from a user as (console) input. N represents the user's money. M represents the number of tickets you will issue for customers. When N or M is equal to 0, your program should be terminated.

2. The price of lottery tickets is 10, and you cannot buy more lottery tickets than you currently have. The current holding amount is the sum of the existing holding amount and the entered holding amount. If you enter more lottery tickets than you have, you will receive the input again. (Initial funding is zero at the start of the program)

3. Randomly determine a series of winning numbers for the round (**6 regular numbers + 1 bonus number**) using ***LotteryGenerator***.

   - The range of possible choices is **from 1 to 20 (integers)**. Therefore, it is not possible to select the series of 2 13 0 5 6 30 + 40 as winning numbers because 0, 30, and 40 are out of the range.
   - The winning numbers (including the bonus number) **cannot** be repeated. For example, the combination of 2 3 14 5 6 6 + 9 is impossible because the number 6 is repeated twice. Similarly, 2 13 4 5 16 8 + 5 is also infeasible due to the duplication of the number 5.

4. Issue M lottery tickets at random by employing the ***LotteryGenerator*** class again.

   - Each ticket consists of a series of six numbers from 1 to 20 (integers).
   - Similar to the case of the winning numbers, it is not allowed to repeat the same number in the seqeunce, e.g., 7 2 13 4 5 5 is not possible due to the overlap of the number 5.
   - However, it is okay to generate the same number sequence for two different tickets when they are accidentally matched.

5. Print out the winning numbers and the results of different tickets computed by the ***LotteryChecker*** class which should be implemented according to the rank rule below:

   - **1st place:** 6 numbers and regular winning numbers are matched or 5 numbers and regular winning numbers are matched and the remaining number is matched the bonus number.

- **2nd place:** 5 numbers and regular winning numbers are matched or 4 numbers and regular winning numbers are matched and the remaining number is matched the bonus number.
- **3rd place:** 4 numbers and regular winning numbers are matched.
- **4th place:** 3 numbers and regular winning numbers are matched.
- **Lose:** All the other scenarios not corresponding to the aforementioned cases.

  (All cases don't care about order.)

6. Print out the final remaining amount and how many times each rank has been won.
7. Go back to 1 and repeat the procedure.

Requirements

1. Your program **must** contain the implementation of the following classes:
   - ***Main***: It functions as the starting point of your program. It is responsible for all the procedures related to console input and output, in addition to the interaction among other classes.
   - ***LotteryGenerator***: A class whose job is to generate (1) a series of winning numbers (i.e., a target) (2) a set of number sequences for the tickets (i.e., trials) bought by customers.
   - ***LotteryChecker***: This class is dedicated for deriving the rank of a ticket.
2. Your program should be based on console input and output. A detailed explanation for how to accept input and print output is specified in the last section.

Scoring Criteria (5 points in total)

- **(2 points)** Your program should satisfy all the requirements listed above, ensuring its accurate working.
- **(2 points)** The role of each class should be distinctive following its definition. For instance, the logic of generating a winning number should be part of LotteryGenerator, not LotteryChecker.
- **(1 point)** There **should not** exist any duplication or redundancy (between classes or in a class) in your code.

An Example of Correct Execution & Formatting

- **Input format**: When your program needs to accept some input from a user, print out "<<" to indicate that it is waiting for the user.
- **Output format**: Always append ">>" in front of your output to indicate that the program is in its printing mode. First, print out "Round Winning Number : " and the corresponding winning numbers. Second, print out "Lottery number : " and the number sequence of each lottery ticket, followed by the rank of the ticket represented by using the format of "Winner (nth place)" or "Lose". Third, print out remaining money and the number of lottery tickets won. See the below example for details.
- **Example**

```
Input your money and number of lottery tickets:
<< 50 5
>> Round Winning number : 1 3 4 5 6 8 + 9
>> Lottery number : 8 5 4 3 1 6 Winner (1st place)
>> Lottery number : 6 1 3 4 5 9 Winner (1st place)
>> Lottery number : 4 5 6 8 9 2 Winner (2nd place)
>> Lottery number : 1 5 4 6 8 10 Winner (2nd place)
>> Remaining money : 0
>> 1st place: 2
>> 2nd place: 2
>> 3rd place: 0
>> 4th place: 0

Input your money and number of lottery tickets:
<< 40 5
Input your money and number of lottery tickets:
<< 40 2
>> Round Winning number : 20 3 14 5 6 8 + 9
>> Lottery number : 1 5 8 14 9 2 Winner (3rd place)
>> Lottery number : 14 5 7 9 8 13 Winner (4th place)
>> Remaining money : 20
>> 1st place: 0
>> 2nd place: 0
>> 3rd place: 1
>> 4th place: 1
```

Input your money and number of lottery tickets:
<< 5 1
>> Round Winning number : 1 4 10 9 4 3 + 13
>> Lottery number : 2 6 17 19 20 9 Lose
>> Remaining money : 15
>> 1st place: 0
>> 2nd place: 0
>> 3rd place: 0
>> 4th place: 0

Input your money and number of lottery tickets:
<< 0 10
>> End of program