CISC106 – Lab 04

This lab is due 06/30/2013 at 11:55 p.m. (submission via Sakai)

- **Please do all of the following problems in ONE file named lab4.py. This is an INDIVIDUAL assignment, please do all work accordingly.**
- **Use comments to separate your program for each problem. For questions where you should write your answers, envelop them as comments.**
- **For now on we will be using the Design recipe to write functions and the assertEqual function to run our test cases. You need to provide at least 3 test cases for each new function you write, when applicable. Please keep in mind that those elements are worth at least half of the question, so you may not want to forget them!**
- **Make sure the file cisc106.py (attached) is in the same directory of your lab4.py while running it. Also make sure you include it in your lab4.py program. You can do this by adding the following line at the very beginning of your file.**

  from cisc106 import *

- **The problems are worth 90 points + 10 points for attending the lab session.**
- **You do not need to make tests for functions that use random generation (unless they are already provided, on which case you should copy them into your file).**

Problem 1: (10 points)

Python can generate random numbers. There's a whole library dedicated to generating random numbers. It's called **random**. So if you want to generate random numbers, you must first import the random number library by placing at the top of your lab4.py file (like you did with the cisc106 file):

from random import *

Now you can generate random numbers between x and y using randrange(x,y)

For instance, if you wanted to generate a random number between 0 and 100 (not including 100) and store it in the variable *randvar*, you'd write the following:

randvar = randrange(0,100)

Problem: Write a function that takes as an input parameter an integer. The function then generates a random number between 0 and that integer. The function then uses the input command to ask the user to input a number between 0 and the input parameter. If the user guesses the number correctly, the function returns a string saying, "Congratulations! You guessed the number!" If the user doesn't guess the number correctly, it should return a string saying, "Your guess was too high" if the guess was above the random number, or "Your guess was too low" if the guess was below the random number.

Problem 2: (10 points)

Write a function that takes no input parameters and returns a Boolean value. It generates a random number 0 or 1 (so the range is 0,2). It then uses the input command to ask the user, "Heads or Tails?" If the user guesses "Heads" and the random number is 1, or if the user guessed "Tails" and the random number is 0, the function returns True. Otherwise, it returns False.

Then write another function that takes no input parameters and returns a string. It calls the previous function 5 times (you should use a loop for that, either while or for). Use a variable to keep track of how many times the user guesses correctly in this new function. Print on the screen whether the user won (he/she guessed correctly 3 or more times), or whether the computer won (the computer wins when the user loses).

Problem 3: (10 points)

Write a function that takes as an input parameter an integer and returns the total sum of all the numbers from 0 until the input number.

Problem 4: (10 points)

Write a function that takes as an input parameter an integer. It then generates that many random numbers between 0 and 100 and sums the random numbers. The function should print the generated numbers and the total sum of them. So, for instance, if the input parameter is 3 and the random numbers are 49, 8, and 22, the output should be to print

"Numbers: "

49

8

22

"Total sum: 79".

Problem 5: (10 points)

A teacher may want students to practice their multiplication tables. Each week it's a different number. So for instance, one week the students should practice their 5s multiplication tables, and the next week their 7s multiplication tables.

Write a function that takes as an input parameter an integer. The function then generates a random number between 0 and 11 (remember, using randrange, 11 won't be included. This is what we want). The function then uses the input command to ask the user, "What is randnum times x?" where

randnum is the random number generated, and x is the input parameter. So, for instance, if the input parameter was 5, and the random number generated was 7, the user would be asked, "what is 7 times 5?"

The function should then take the answer the user entered and check whether the user was correct or not, returning a Boolean value with True if the user was correct, and False if the user was wrong.

Problem 6: (10 points)

Write a function that takes a list of integers as an argument and, for each element on the list, calculates and prints its square.

Problem 7: (15 points)

Wacky BlackJack: Write a function that takes no input parameters. It then generates random numbers between 1 and 11 until the user enters "no" in response to the input command, "Do you wish to continue?".

The user wants to come as close to 21 as possible without going over. (The function should keep track of the total and print it out before asking if the user wants to continue). Once the user has decided not to continue (they entered, "no"), the function should generate another random number (outside the loop), this time between 11 and 31. This is the "dealer's hand". The function should print out the dealer's score. Then, if the Dealer's hand is over 21, the user automatically wins and the program should print that the user won. If the Dealer's hand is 21 or under, and the user's hand is over 21, the dealer wins and a message should be printed saying that the dealer won. If the dealer's hand is 21 or under and it is greater than the user's hand, a message should be printed saying that the dealer won. If the user's hand is 21 or under and greater than or equal to the dealer's hand (we're giving the user the advantage here), a message should be printed saying that the user won.

Problem 8: (15 points)

Write a function, **random_list**, that produces a list of random numbers between 0 and maxvalue (inclusive) with length of N. The following tests should work for your function:

```
assertEqual(len(random_list(100, 1000)), 1000)
assertEqual(min(random_list(100, 1000)) >= 0, True)
assertEqual(max(random_list(1, 1000)) <= 1, True)
```