NYU, Tandon School of Engineering

CS-UY 1114 Introduction to Programming and Problem Solving — Fall 2015

Homework # 7

Due _Nov. 2nd 11:55pm__

Submission instructions:

1. You should submit your homework in the NYU Classes system.
2. Put your code in 3 '.py' files, each containing a script for each question. Name your files 'hw7q1.py', 'hw7q2.py', 'hw7q3.py', etc.

   Zip your3 '.py' files to one file in the following format: '.rar' or '.zip'. Name your zipped file 'Your_name_NetID_hw7.rar' or 'Your_name_NetID_hw7.zip'.

   Make sure your zipped file is not corrupt before submitting to NYU Classes. **Corrupt file will not be graded.**

Key points in this homework:

- 'for' and 'while' loops.
- string and its methods.
- Random number generator.

Note: do **NOT** use syntax that was not covered in class.

Question 1:

In this problem you need to implement Caesar cipher. Ask the user to enter a right shift and a string *s*, your program should display the encrypted string.

You program should use the shift to encrypt each capital letter as another capital letter, each lowercase letter as another lowercase letter, and leave all other character, ie., '$', '#','3', etc. without shifting.

For example, an execution of your answer would look like:

Enter a right shift: 23

Enter a string with at least one capital letter: Lazy Dog

Encrypted string is: Ixwv Ald

Question 2: Power table

Print out power table with 5 rows and 10 columns. Value of the power table at row i, column j is j^i.

The columns should be spaced by a tab.

Expected output:

```
>>> ================================ RESTART ================================
>>>
1       2       3       4       5       6       7       8       9       10
1       4       9       16      25      36      49      64      81      100
1       8       27      64      125     216     343     512     729     1000
1       16      81      256     625     1296    2401    4096    6561    10000
1       32      243     1024    3125    7776    16807   32768   59049   100000
>>>
```

Question 3:

Random number guessing game: write a program to generate a random integer between 1 and 100 (inclusive) and have the user guess the number. **User can guess at most 5 times.**

In more detail, your program should start by generating a random number between 1 and 100.

Then you should input an integer and the program would print out a **new range** if your guess is wrong. The program would also print out **how many guesses you have left**.

If your guess is correct, the program should announce in how many guesses you have made it.

Example:

I thought of a number between 1 and 100! Try to guess it.

Try to guess what it is: 15

Wrong guess! Guess in range 16 to 100. You have 4 more chances.

Try to guess what it is: 34

Wrong guess! Guess in range 16 to 33. You have 3 more chances.

Try to guess what it is: 23

Congrats! You guessed my number in 3 guesses.