NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
ISLAMABAD CAMPUS
INTRODUCTION TO COMPUTING (CS101) - FALL 2017
ASSIGNMENT-2

Due Date: October 15, 2017 (09:00pm)
Instructions:

1. *Write Python program for all the problems.*
2. *Solution to all the problems should be written in a separate (.py) file.*
3. *Submit the source code (i.e. python code in .py file) via slate. Submissions via email will not be accepted.*
4. *Moreover, you must submit your notebook file (.ipynb) as well with all the codes and their outputs in different cells.*
5. *Use proper naming convention to name the file containing source code.*
   For example, the file containing the source code for first question of the first assignment should be named as i17xxxx_assignment_2_q1.py, replace i17xxxx with your student number.

1. Write code to find the minimum point of following function. *[Hint: Evaluate the function by writing nested while loops for a large domain of x and y values and find the values where minimum value occur]*.

$$f(x,y)=(1-x)^2+100(y-x^2)^2$$

2. Write a program that produces the following output. Use nested while loops to capture the structure of the figure.

```
!!!!!!!!!!!!!!!!!!!!!!
\\!!!!!!!!!!!!!!!!!!//
\\\\!!!!!!!!!!!!!!////
\\\\\\!!!!!!!!!!//////
\\\\\\\\!!!!!!////////
\\\\\\\\\\!!//////////
```

3. Guess the Number Game: Now write a program that will ask the user to guess a secret number selected randomly by your program in 3 trials. After each trial, you should inform the user if she didn't guess correctly, the number trials remaining, and inform the user that she lost or won the game (when she guessed correctly.)
When game is over, ask the user to play again: If the user types "yes", have the game start again. If the user types "**no**", then stop playing the game. [*Hint: You can use random.random() function to get a random number in the range of [0,1], you can multiply the random number to another number X to get an integer random number in the range [0,X], i.e.*

```
import random
x=round ( random.random() *100 ) # now x will have a random
number
# in the range [0,100], multiplying with any other number
will produce the number in that range.
```

4. Write while loops to produce the following output:

```
##
# #
#  #
#   #
#    #
#     #
```

5. Write while loops to produce the following output:

```
    1
   22
  333
 4444
55555
```

6. Write while loops to produce the following output, with each line 40 characters wide:

```
----------------------------------------
_-^-_-^-_-^-_-^-_-^-_-^-_-^-_-^-_-^-_-^-
1122334455667788990011223344556677889900
----------------------------------------
```

7. It's common to print a rotating, increasing list of single-digit numbers at the start of a program's output as a visual guide to number the columns of the output to follow. With this in mind, write nested while loops to produce the following output, with each line 60 characters wide:

```
         |         |         |         |         |         |
123456789012345678901234567890123456789012345678901234567890
```

8. Write a program that produces the following output (with loops):

```
****** //////////// ******
*****  //////////\\  *****
****   ////////\\\\   ****
***    //////\\\\\\    ***
**     ////\\\\\\\\     **
*      //\\\\\\\\\\      *
       \\\\\\\\\\\\
```

9. Write a program that produces the following output (with loops, **Remember Divide and Conquer**):

```
+------+
|  ^^  |
| ^  ^ |
|^    ^|
|  ^^  |
| ^  ^ |
|^    ^|
+------+
|v    v|
| v  v |
|  vv  |
|v    v|
| v  v |
|  vv  |
+------+
```

10. Write a program that produces the following output (with loops):

```
+---------+
|    *    |
|   /*\   |
|  //*\\  |
| ///*\\\ |
| \\\*/// |
|  \\*//  |
|   \*/   |
|    *    |
+---------+
| \\\*/// |
|  \\*//  |
|   \*/   |
|    *    |
|    *    |
|   /*\   |
|  //*\\  |
| ///*\\\ |
+---------+
```

11. Write a program that produces the following output (with loops):

```
        *               *
          *           *
             *      *
                *
            *      *
        *              *
    *                      *
*                              *
    *                      *
         *             *
             *     *
                *
           *         *
       *                 *
  *                          *
*                              *
    *                      *
         *             *
             *     *
                *
           *         *
       *                 *
  *                          *
```

12. Write a program that produces the following output (with loops):

```
                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8
    1 2 3 4 5 6 7
      1 2 3 4 5 6
        1 2 3 4 5
          1 2 3 4
            1 2 3
              1 2
                1
```

13. Write a program that allows a user to convert positive numbers from binary (base 2) to decimal (Base 10). Use simple and accurate algorithm to produce the correct result. Here is an example of an algorithm illustrating how to convert from binary (base 2) to

Decimal:

(Base 10):

The binary or base-2 number 1 0 0 1 0 1 0 1 can be converted to a decimal number as follows:

$$1*2^7+0*2^6+0*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$$

= 128 + 16 + 4 + 1

= 149

The binary number 1 0 0 1 0 1 0 1 is written as 149 in the decimal system.

14. Write a program that accepts two integer input “lines” and “cheers” and prints a series of "cheer" lines at increasing levels of indentation. The first parameter represents the number of lines of output to print, and the second represents the number of "cheers" per line. For example, if lines=2 and cheers =4 than you should print 2 lines of output, each containing 4 "cheers." A "cheer" is an occurrence of the word "Go" in the output.

Neighboring cheers are separated by the word "Buddy" (of course you can put name of your favorite political person here as well :)), so 1 cheer is printed as "Go", 2 cheers as "Go Buddy Go", 3 cheers are printed as "Go Buddy Go Buddy Go", and so on.

The lines you print should be displayed at increasing levels of indentation. The first line displayed should have no indentation, but each following line should be intended by 3 spaces more than the one before it. In other words, the 2nd line of output should be indented by 3 spaces, the 3rd line by 6 spaces, and so on. You may assume that both parameters passed your function will have values of at least 1.

INPUT:

Lines=2	Lines=4	Lines=2
Cheers=1	Cheers=3	Cheers=4

OUTPUT

```
Go
   Go
```

```
Go Buddy Go Buddy Go
   Go Buddy Go Buddy Go
      Go Buddy Go Buddy Go
         Go Buddy Go Buddy Go
```

```
Go Buddy Go Buddy Go Buddy Go
   Go Buddy Go Buddy Go Buddy Go
```

15. Write a while loop that produces the following output:

$$\frac{1}{2},\frac{3}{4},\frac{7}{8},\frac{15}{16},\frac{31}{32}\ldots\ldots\ldots\ldots\ldots.$$

16. Print the first 10 numbers of the series:

4, 14, 49,117,256,478,841

17. PakAsia is a country that has currency notes of 6, 9 and 20 PakAsia Rupees (PAR) only. Thus, it is possible, for example, to exchange exactly 15 Pakistani Rupees (PKR) (with one note of 6 Rs and a second note of 9 Rs will add up to make 15, since 1PKR == 1PAR), but it is not possible to exchange exactly 16 PKR, since no non-negative integer combination of 6's, 9's and 20's notes of PAR add up to make 16 PKR. To determine if it is possible to exchange exactly **n** PKR to PAR, one has to find non-negative integer (can be 0) values of a, b, and c such that:

$$6a+9b+20c=n$$

Now write a Python code that takes, ‘n’ PKR from user as input, and prints if it is possible to exchange it with a combination of 6, 9 and 20 PAR such that the total sum of PakAsia Rupees equals **n**, and otherwise prints it is not possible.