Introduction to Python
Assignment 3

1. WAP that will accept two sets of numbers and calculate the Pearson's correlation coefficient between them. (Use necessary exception handling). *(Optional if you do not know what it is)*

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,][\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

2. Write a program to print all prime numbers from 1 to 300.

3. Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form.
   29 * 1 = 29
   29 *2 = 58
   …

3. **WAP which repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.** ***(Use of list is not allowed in this program)***

   Enter a number: 4
   Enter a number: 5
   Enter a number: nine
   Invalid input
   Enter a number: 7
   Enter a number: done
   16 3 5.33333333333

4. Write another program that prompts for a list of numbers as above and at the end prints both the maximum and minimum of the numbers instead of the average. *(Use of list is not allowed in this program)*

5. Guess a number is a game that prompts a player to guess a number between 0 and 9, which is randomly generated by the system. When the input given by the user matches the number generated by the system then the user wins. The game should go as follows:

Guess the number: 5
Sorry, try again
Guess the number: 3
Sorry, try again
Guess the number: 8
You got it right! Congo!

6. Make an improvement to the Guess a number game. Guide the user where they are standing and limit the number of attempts to 3. For example, the game should go like this:

   Guess the number: 4
   Too low
   Guess the number: 9
   Too high
   Guess the number: 5
   Sorry you Lost!

7. **Let us make the above game a little more interesting by converting it into a gambling problem. Suppose that a player starts with Rs. 1,000. If a player can guess the number in his first chance, then he will be given a prize of Rs. 5,000, if he requires 2 attempts then he will get a prize of Rs.1,000. If he loses then he will lose Rs. 500. For example the game should go like this:**

   You have a cash of Rs. 1,000 with you...
   Guess the number: 8
   Too high
   Guess the number: 3
   You have just won Rs. 1,000
   Your balance: Rs. 2000

Extra Credit Problem

8. Recall the problem when you kept prompting the user to withdraw money from the wallet to start the game. Integrate this program with the one you have written for problem 7.
   a. For this problem, consider that the minimum balance to start the game is Rs. 500.
   b. After one game is over add/deduct money from the account.
   c. Prompt the user if he/she would like to play one more time.
      i. If the user says “Yes” then check if the user has enough balance to play the game again. If not prompt the user to withdraw money from his wallet

before the program starts the game. When the balance will be greater than Rs. 500 the program will automatically start the game.

ii. If the user says “No”, then stop the game with a message “Have a wonderful day!”