Python Programming Lab (R20)

1) Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.
2) Write a program that asks the user to enter three numbers (use three separate input statements). Create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.
3) Write a program that uses a *for* loop to print the numbers 8, 11, 14, 17, 20, . . . , 83, 86, 89.
4) Write a program that asks the user for their name and how many times to print it. The program should print out the user's name the specified number of times.
5) Use a *for* loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

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

6) Generate a random number between 1 and 10. Ask the user to guess the number and print a message based on whether they get it right or not.
7) Write a program that asks the user to enter a word and prints out whether that word contains any vowels.
8) Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters *abcde* and *ABCDE* the program should print out *AaBbCcDdEe*.
9) Write a program that generates a list of 20 random numbers between 1 and 100.
   (a) Print the list.
   (b) Print the average of the elements in the list.
   (c) Print the largest and smallest values in the list.
   (d) Print the second largest and second smallest entries in the list
   (e) Print how many even numbers are in the list.

10) Write a program that asks the user for an integer and creates a list that consists of the factors of that integer.
11) Write a program that generates 100 random integers that are either 0 or 1. Then find the longest run of zeros, the largest number of zeros in a row. For instance, the longest run of zeros in [1,0,1,1,0,0,0,0,1,0,0] is 4.
12) Write a program that removes any repeated items from a list so that each item appears at most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

13) Write a program that asks the user to enter a length in feet. The program should then give the user the option to convert from feet into inches, yards, miles, millimeters, centimeters, meters, or kilometers. Say if the user enters a 1, then the program converts to inches, if they enter a 2, then the program converts to yards, etc. While this can be done with if statements, it is much shorter with lists and it is also easier to add new conversions if you use lists.
1 foot=12 inches, 1 foot=0.333333 yards, 1foot=0.000189394 miles
1 foot=304.8 millimeters, 1 foot=30.48 centimeters, 1 foot=0.3048meters,
1 foot=0.0003048kilometers

14) Write a function called *sum_digits* that is given an integer num and returns the sum of the digits of num.

15) Write a function called *number_of_factors* that takes an integer and returns how many factors the number has.

16) Write a function called primes that is given a number n and returns a list of the first n primes. Let the default value of n be 100.

17) Write a program that reads a file consisting of email addresses, each on its own line. Your program should print out a string consisting of those email addresses separated by semicolons.

18) Write a program that reads a list of temperatures from a file called *temps.txt*, converts those temperatures to Fahrenheit, and writes the results to a file called ftemps.txt.

19) Write a class called Time whose only field is a time in seconds. It should have a method called *convert_to_minutes* that returns a string of minutes and seconds formatted as in the following example: if seconds is 230, the method should return '3:50'. It should also have a method called *convert_to_hours* that returns a string of hours, minutes, and seconds formatted analogously to the previous method.

20) Write a program to illustrate operator overloading in python.

21) Write a program to demonstrate Try, except, else and finally.

22) Write a program to demonstrate user defined exceptions.

Additional Programs

1) Write a program to check whether given number is Armstrong number or not
2) Write a program to read your name and print it in reverse
3) Write a recursive function called factorial that is given an integer num and returns the factorial of the num.