Introduction to Computing
Practice Assignment # 2

1. Write a function named **isIn** that accepts two strings as arguments and returns **True** if either string occurs anywhere in the other as a substring, and **False** otherwise. **Hint:** you might want to use the built-in **str** operation **in.**

2. Write a function **primeFactor** that takes one parameter as its argument and obtain the prime factors of the number. **Hint:** Find all the prime number that multiply together to make the original number. e.g. prime factors of 15 are 5 and 3.

3. Take an integer number as an input from user**,** write a function that checks weather a given number is Armstrong or not. **Hint:** If sum of the cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)

4. Write a function to find the octal equivalent of the given number. For example: the octal equivalent of 50 is 62.

5. Write your own **Range** function that take three parameters (start, stop and step) as an input and return the list of integer numbers.

6. Write a function **binary2Decimal** that take binary number as input and return decimal equivalent of the given number.

7. Write a function **Multiple** that considers a pair of integers and determines whether the second integer is the multiple of the first or not. The function should take two integers as arguments and return **True** if the second is a multiple of the first, and **False** otherwise.

8. Write a function that displays a solid square of a character whose side and the character to be printed are specified in integer and character as the function parameters. For example, if side is 5 and character is '#', the function should display:

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

9. Implement the following integer functions:
   i. A Function **Celsius** that returns the Celsius equivalent of a Fahrenheit temperature.
   ii. A Function **Fahrenheit** returns the Fahrenheit equivalent of a Celsius temperature.
   iii. Use these functions to write a program that prints charts showing the Fahrenheit equivalents of all Celsius temperatures from 0 to 100 degrees, and the Celsius equivalents of all Fahrenheit temperatures from 32 to 212 degrees. Print the outputs in a tabular format that minimizes the number of lines of output while remaining readable.

10. An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sums up to the number. For example, 6 is a perfect number because 6 = 1 + 2 + 3. Write a function **perfect** that determines whether parameter number is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and 1000.

11. Write a program that simulates coin tossing. For each toss of the coin, the program should print Heads or Tails. Let the program toss the coin 100 times, and count the number of times each side of the coin appears and then print the results. The program should call a separate function **flip** that takes no arguments and returns 0 for tails and 1 for heads. Note: To get random 1 or 0 you can import a **random** module and then use **random.randint(0,1)**

12. Declare a list of 10 random integers and then find mean, median and mode of integers.

13. Declare a **list** of 20 integers that has duplicate numbers, write a program to delete duplicate numbers from list.

14. Write a function that takes a tuple of 10 number as a parameter and return the minimum of all the numbers.

15. Write a program that take 10-time, name (string) and marks (integer) as an input from the user. Populate a list such that all even indices have name and odd indices with the corresponding marks. e.g. [‘Ahmad’, 29, ‘Asad’, 15, ‘Zainab’, 20]. At the end print the list. Hint: You can use append function of list data type.

16. Implement a program that starts by asking the user to enter a login id (i.e., a string). The program then checks whether the id entered by the user is in the list ['Ahmad', 'Zainab', 'Hina', 'Ali'] of valid users. Depending on the outcome, an appropriate message should be printed. Regardless of the outcome, your function should print “Done” before terminating. Here is an example of a successful login:

    >>>

    Login: Ali

    You are in!

    Done.

17. Take a name (i.e., a string) as an input from the user and insert it into a list, then pass this list to a function **swap** that exchange the first and last values of list. After swapping print the resultant list.

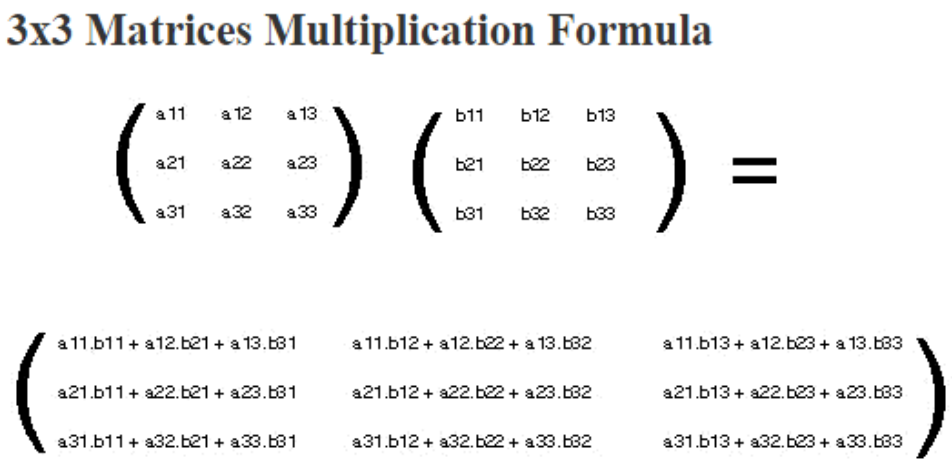
18. Write a for loop that iterates over a list of strings **myList** and prints the first three characters of every word. e.g. If **myList** is the list ['January', 'February', 'March'] thenthe following should be printed:

    Jan

    Feb

    Mar

19. Write a **twoDlist** function that perform 3×3 matrix multiplication. e.g.



20. Write a program that requests an integer **n** from the user and append the squares of all numbers from **0** up to, but not including, **n**. At the end print the list.

    >>>

    Enter n: 4

    [0, 1, 4, 9]

21. Implement a function **ion2e()** that takes a string as input and returns a copy of the string with the following change: if the entered word (input string) ends in **'ion'**, then **'ion'** is replaced with **'e'**, otherwise it returns the same word.

    >>> ion2e(“congratulation”)

    “congratulate”

    >>> ion2e(“marathon”)

    “marathon”