**Programs – 15IT322E – Tutorial**

**Lists:**

1. Get a list of items from user till he types ‘quit’ and print the list.

2. Intersection of lists. Take all those elements which are common to both of the initial lists and store them into another list.

3. Write a Python program to remove duplicates from a list.

4. Write a Python program to generate a 3\*4\*6 3D array whose each element is \*.

5. Sort a list of tuples in descending order according to the last element of the tuple.

Sample list: [(10, 20, **40**), (40, 50, **60**), (70, 80, **90**)]  
Expected Output: [(70, 80, 90), (40, 50, 60), (10, 20, 40)]

6. Get a list of numbers from the user. Separate the list into two lists of odd and even numbers.

**Strings:**

1. Get a string from the user. Replace all occurrences of its first character (except the first character itself) with $. For example if the input string in “google” your program output should be “goo$le”.

2. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

Suppose the following input is supplied to the program:

without,hello,bag,world

Then, the output should be:

bag,hello,without,world

3. Write a program that accepts a sentence and calculate the number of letters and digits.

Suppose the following input is supplied to the program:

hello world! 123

Then, the output should be:

LETTERS 10

DIGITS 3

3. Find word frequency from a sentence which is given input as a string to your program. Use dictionaries

4. Your program should print “Valid” if both halves of a string have the same characters(inclusive of frequency checks as well. If the input string cannot be divided into halves, then display a message accordingly.

Example:

aabbaa ia a valid string

aabbba is invalid string

5. Your program should get a name in the format First Middle Last and print in the format F.M.Last.

Example:

If input is Lal Bahadhur Shastri

Output is L.B.Shastri

**Functions:**

1. A valid card number passes a digit-sum test known as the Luhn checksum algorithm. Luhn's algorithm states that if you sum the digits of the number in a certain way, the total sum must be a multiple of 10 for a valid number.

Systems that accept credit cards perform a Luhn test before contacting the credit card company for final verification.

The algorithm for summing the digits is the following.

Consider each digit of the credit card to have a zero-based index: the first (starting from right) is at index 0, and the last is at index 15. Start from the rightmost digit and process each digit one at a time. For digits at even-numbered indexes (the 14th digit, 12th digit, etc.), simply add that digit to the cumulative sum. For digits at odd-numbered indexes (the 15th, 13th, etc), double the digit's value, then if that doubled value is less than 10, add it to the sum. If the doubled number is 10 or greater, add each of its digits separately into the sum.

The following pseudocode describes the Luhn algorithm to sum the digits:

4408041254369873 is an example credit card number that passes the Luhn algorithm. The following figure shows the algorithm summing the latter number in detail. Notice how digits at even indexes are doubled and potentially split into two digits if they exceed 10 when doubled. For example, the number 7 at index 8 which is doubled to 14 which split to make 1+4.

An example checksum using the Luhn algorithm.

CC # 4408 0412 7436 9853

4 4 0 8 0 4 1 2 7 4 3 6 9 8 5 3

Scale \*2 \*2 \*2 \*2 \*2 \*2 \*2 \*2

--------------------------------------------------------------------

8 4 0 8 0 4 2 2 14 4 6 6 18 8 10 3

Sum = 8 + 4 + 0 + 8 + 0 + 4 + 2 + 2 + 1+4 + 4 + 6 + 6 + 1+8 + 8 + 1+0 + 3

= 70

70 is divisible by 10, therefore this card number is valid.

Write a program where the user can type in a credit card number and receive a message stating whether the number was valid. The program should have a function validate\_number() that takes the credit card number as argument and prints the message “Valid Credit Card Number” or “Invalid Credit Card Number” accordingly. The program should print an error message and exit if the length of the credit card number is not equal to 16.

2. In general, if any one side length is greater than or equal to the sum of the other two then the lengths cannot be used to form a triangle. Otherwise they can form a triangle. Write a function that determines whether or not three lengths can form a triangle. The function will take 3 parameters and return a Boolean result. In addition, write a program that reads 3 lengths from the user and demonstrates the behaviour of this function.

**Dictionaries:**

1. Find the most popular friend

Individual Student Friend Rating Score Dictionaries, (Construct a min of 5 student dictionaries)

Score 1 is lowest and Score of 5 is highest. Every student rates all the other students. So each dictionary has 9 elements.

Example with 3 students,

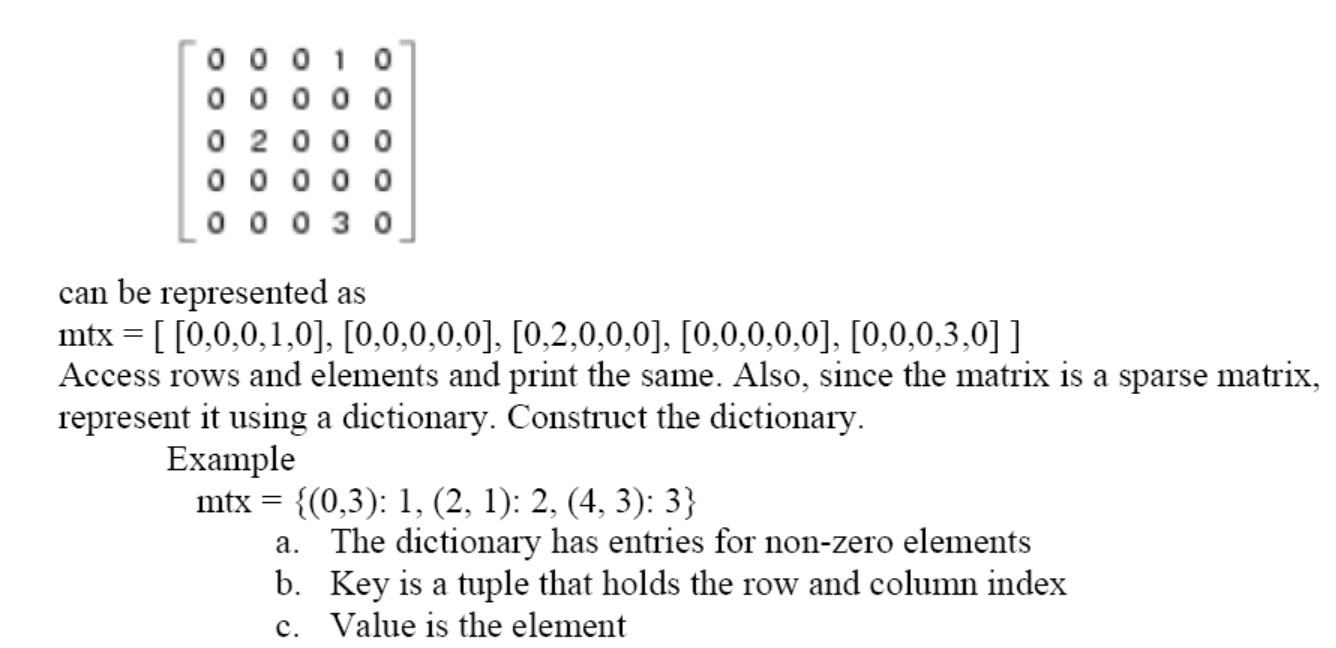
sam\_friends = {“Bob”:3, “Raj”:5}

bob\_friends = {“Sam”:5,”Raj”:1}

raj\_friends = {“Sam”:3,”Bob”:5}

Find who has the highest score and that student is the most popular friend. Print the most popular friend and his/her score.

2.



3. In cryptography, a ceaser cipher is a substitution cipher and a very simple encryption technique in which each letter is replaced by a letter some fixed number of positions down the alphabet. The ROT 13 is the widely used ceaser cipher with a shift of 13 (rotate by 13 places). For example,'a' is replaced with 'n', 'b' with 'o' and So on tlll'z' with m. The key for ROT 13 can be represented using a dictionary which could be hardcoded. Assume plain text is only in small case and only contains alphabets. Write a program with function to encrypt plain text to cipher text.

4. Write a program that uses a dictionary that contains usernames and passwords. The program should ask the user to enter their usemame and password. If the usemame is not in the dictionary, the program should indicate that the person is not a valid user of the system. If the usemame in the dictionary, but the user does not enter right password, the program to program should say that passlr,ord is invalid. If the password is correc! the program should tell "Loggedln". Write the validation in a function.

5. Create a function that takes an integer between 0 and 999 as its only parameter, and returns a string containing the English words for that number. For example, if the parameter to the function is 142 then your function should return “one hundred forty two”. Use one or more dictionaries to implement your solution rather than large if/elif/else constructs.