

Given Date: Feb 21, 2019

Submission Date: Feb 25, 2019 – Latest 10:00 o'clock

In this lab you will study on:

- . Input/output
- . Expressions
- . If conditions
- . For loops/while loops
- . Functions
- . Randint/time

Please write your answers in PYTHON, execute them and attach the output of your execution and submit all of them to Moodle.

Question 1:

- a. Write a function that takes an integer, *n*, as a parameter and displays a *n*-by-*n* matrix. Each element in the matrix is 0 or 1, which is generated randomly.

Hint: You can generate a random integer number between *x* and *y* (inclusive) by using the function **random.randint(x, y)**.

import random library for this function as the first statement in your program.

- b. Using your function write a program that prompts the user to enter a positive integer *n* and displays an *n*-by-*n* matrix.

Sample run:

```
Enter a positive integer: 6
0 0 1 1 1 1
1 1 1 0 1 1
0 1 1 0 1 0
0 1 1 0 1 1
1 1 1 0 1 0
1 1 0 1 1 0
```

Question 2:

Write a program that inputs a two dimensional list and a one dimensional list from the user. The program should display an appropriate message if the given one dimensional list exists in the given two dimensional list or not. The program should continue to enter one dimensional list until the user enters an empty list.

Sample Run:

```
Number of rows for two dimensional list please: 3
Enter 3 rows as a list of ints please:
1 2 3
4 5 6
7 8 9
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Another list of ints please:
4 5 6
This list exists in the two dimensional list
Another list of ints please:
1 3 5
This list does not exist in the two dimensional list
Another list of ints please:
```

Question 3:

- Write a function that takes two positive integer and returns GCD which is found by **Consecutive Integer Checking Algorithm**.
- Using your function write a program that prompts the user to enter two positive integer and displays the GCD of them.

Sample Run:

```
Non-negative first integer >=2 please: 64
Non-negative first integer >=2 please: 12
greatest common divisor: 4
```

Question 4:

- Write a function, **Sieve of Eratosthenes** as defined in chapter 1 (CH01 in lecture notes) by using python. Do not forget that mathematicians do not consider 1 to be a prime number. So output of this function should not include 1. Also, in order to able to use the output array of this function in the **prime factorization** function defined in the 2nd part of the question, you have to eliminate all zero (0) values from output array.

Hint: you may **import math** library and you may use **math.sqrt(x)** and **math.floor(x)** functions in python.

- Write a function, **prime factorization** which will find and return all the prime factors of a given positive integer number. This function should call and use the output of **Sieve of Eratosthenes** function while finding prime factorization of the given number.
- Write the main program which finds and prints two different positive integers (let's say m & n) **greatest common divisor** by using middle school procedure as below..

Middle-school procedure

Step 1 Find the **prime factorization** of **m**

Step 2 Find the **prime factorization** of **n**

Step 3 Find all the **common prime factors** of this two numbers and **print** them.

Step 4 Compute the **product of all the common prime factors** and print it as **gcd(m,n)**

NOTES:

Do not copy functions from the internet. You have to write them according to above instructions. If you submit same or very similar codes, I will accept them as cheat too.

Sample Run:

```
Non-negative integer >=2 please: 64
Another Non-negative integer >=2 please: 12

Common prime factors are: [2, 2]
Greatest common divisor: 4
```

Question 5:

By using **time** function of the python, compare the time efficiency of the question 3 and question 4 by giving the same two numbers to find the GCD of them. Please give very big numbers to be able to find the time which is suitable to compare them.

Pease send the copy of compared timing outputs and write a conclusion in one or two sentence.