Islamic University of Technology (IUT)
Organization of Islamic Cooperation (OIC)
Department of Electrical and Electronic Engineering (EEE)

EEE 4416: Simulation Lab
Lab – 06 Assignment

Exercise - 01

Problem statement: Draw Z

Write a function named 'draw_Z' that takes an integer 'n' (n>2) as the input and draws the following matrix of size 'n' with 0 and 1. Your program should work for any values of n.

For example,

- Input: n=5
- Output: [1 1 1 1 1;

```
          0 0 0 1 0;
          0 0 1 0 0;
          0 1 0 0 0;
          1 1 1 1 1]
```

Test case – 02

- Input: 3
- Output: [1 1 1

```
            0 1 0
            1 1 1]
```

Part-02:

- Draw a z-shape matrix of size 100. Interchange the positions of 0 and 1 [You may use the *tilde (~) operator*].
- Display the binary image (use the 'imshow' function).

Asif Newaz
Lecturer, Department of EEE, IUT

Exercise – 02

Problem statement: Repeating elements

Write a function named 'repeat_elem' that takes an integer 'n' as input and provides the following output. Here, each element is repeated by its number of times, e.g., 4 is repeated 4 times.

Test Case – 01:

- Input: n = 3
- Output: [1, 2, 2, 3, 3, 3]

Test Case – 02:

- Input: n = 5
- Output: [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]

Test Case – 03:

- Input: n = 1
- Output: 1

Exercise – 03

Problem statement: Goldbach Conjecture

The Goldbach conjecture states that every **even** integer **greater than 2** is the sum of two primes.

For example, 8 = 5 + 3

20 = 17 + 3 = 13 + 7

As you can see, there can be multiple ways an even integer satisfies the Goldbach conjecture.

Write a function named 'Goldbach' that takes an integer (even or odd) as input and provides **any of the perfect solutions.** The returned array should be sorted.

Test Case – 01:

- Input: n = 10
- Output: [3,7] or [5,5]

Test Case – 02:

- Input: n = 101
- Output: "Odd numbers don't satisfy the necessary criteria"

Test Case – 03:

- Input: n = 500000
- Output: [859, 499141] or …

- An important part of writing a program is that it should satisfy all possible scenarios, not just the ones that are shown in test cases.
- If you look at the problem above, there is an important **corner case** – n=2. Although I didn't write a test case for that, you should make sure that your program handles that case as well.
- Generating test cases by yourself is another important part of writing code.

Asif Newaz
Lecturer, Department of EEE, IUT

Exercise – 04

Problem statement:

Say, I want to perform different mathematical operations between two inputs a and b. The operations are given as the 3rd parameter of your function. But sometimes, 3rd parameter may not be given.

Operations –

i. "add" -- stands for addition (a+b)
ii. "sub" -- stands for subtraction (a-b)
iii. "mul" -- stands for multiplication (a*b)
iv. "div" -- stands for division (a/b)
v. If not given, then perform modulo operation [mod(a,b)]

Write a function named 'arith_op' that takes two or three arguments as the input and performs the aforementioned tasks.

Test Case – 01:

- Input: a=20, b=10, 'sub'
- Output: 10

Test Case – 02:

- Input: a=5, b=10, 'div'
- Output: 0.5

Test Case – 03:

- Input: 22, 10
- Output: 2

Test Case – 04:

- Input: 22
- Output: 'Not enough input arguments'

Exercise - 05

Problem Statement: Diagonally Dominant Matrix

A matrix is said to be diagonally dominant if, for every row of the matrix, the **magnitude** of the diagonal entry in a row is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row.

Given a matrix, find out whether it is diagonally dominant or not.

Write a function 'diag_dominant' that takes one or two inputs – the matrix **and/or a clause**. The clause states 'weak' or 'strict' indicating *weak diagonal dominance or strict diagonal dominance.*

- Weak diagonal dominance refers to weak inequality (>=)
- strict diagonal dominance refers to strict inequality (>).
- The default value for the 2nd input is 'weak'.

The function should return a logical true or false.

Test Case – 01:
- Input: [5, 0; 1, 5]
- Output: True

Test Case – 02:
- Input: [5, 0, 0, 10; 1, 5, 5, 10; 2, 4, 4, 5; 3, 2, 2, 1]
- Output: False

Test Case – 03:
- Input: [-2, 2, 1; 1, 3, 2; 1, -2, 0]
- Output: False

Test Case – 04:
- Input-1: [-4, 2, 1; 1, 6, 2; 1, -2, 5]
- Input-2: 'strong'
- Output: True

Test Case – 05:
- Input-1: [5, 0, 0; 1, 5, 2; 2, 4, 6]
- Input-2: 'strong'
- Output: false

Test Case – 06:
- Input-1: [5, 0, 0; 1, 5, 2; 2, 4, 6]
- Input-2: 'weak'
- Output: true

Asif Newaz
Lecturer, Department of EEE, IUT

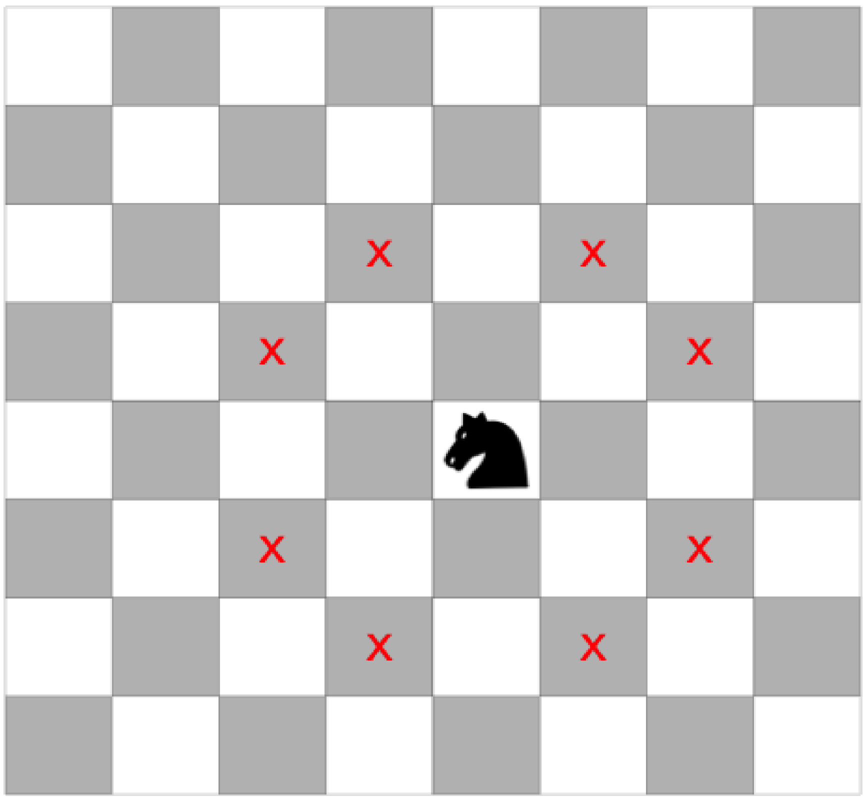
Exercise - 06

Problem Statement: Night's Watch

A chess board has 8 rows and 8 columns. You can think of it as an 8-by-8 matrix. Now your board contains only one piece, a knight. It is placed in the (x, y) position. In the picture below, the knight is at (5, 5).

A knight has 8 possible valid moves that are marked as cross in the picture. For example, a knight can move – 'two steps right and one step up'; which brings it to position (4, 7).

Write a function that takes the original position of the knight as input and returns all the valid positions to move as output in an 8 by 2 matrix.



Output: (4,7), (6,7), (4,3), (6,3), (3,4), (3,6), (7,4), (7,6). Use an 8 by 2 matrix to represent the answer.

Out = [4, 7;

6, 7;

4, 3;

… …]

- ✓ You can place the output positions in any order.
- ✓ You also need to check if the position is valid or not. For instance, (9, 2) would not be a valid position.

Asif Newaz
Lecturer, Department of EEE, IUT

Test Case – 02:

- Input: [1, 1]
- Output: [3, 2; 2, 3]

Test Case – 03:

- Input: [8, 8]
- Output: [7, 6; 6, 7]

Test Case – 04:

- Input: [8, 9]
- Output: ‘Invalid position’

Exercise - 07

Problem statement: Repeating elements again

Repeat exercise - 02 with a slight modification. The function now may take one additional input argument – the starting position. If not given, the function will work just like before, starting from 1.

Write a function called ‘‘repeat_elem_v2’ that satisfies the following test cases.

Test Case – 01:

- Input: 3
- Output: [1, 2, 2, 3, 3, 3]

Test Case – 02:

- Input: 5, 4
- Output: [4, 4, 4, 4, 5, 5, 5, 5, 5]

Test Case – 03:

- Input: 1
- Output: 1

Test Case – 04:

- Input: 1, 8
- Output: [ ]

Test Case – 05:

- Input: 6, 6
- Output: [6, 6, 6, 6, 6, 6]

Asif Newaz
Lecturer, Department of EEE, IUT