B. Sc. in EEE
Summer semester

Date: July 2025

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Lab Quiz - **02 (Set-B)**

Summer Semester - 2025

Course Number: EEE 4416

Full Marks: 20

Course Title: Simulation Lab

Time: 35 minutes

Question – 01

- Create a row vector vC using the colon operator that starts from 2, ends at 35, with increment = 3.
- How many elements are there in the array vC?
- Create the following new vectors by assigning elements of vC to the new vectors:
-

a) A vector (name it vC_odd) that contains all the elements with **odd indices** of vC.

vC_odd = [2 8 14 ... 32]

b) A vector (name it vC_even) that contains all the elements with **even indices** of vC.

vC_even = [5 11 17 ... 35]

- Join/Stack the two vectors vertically to create a 2D matrix.

vC_mat = [2, 8, 14, …;
5, 11, 17, …]

- Count the number of elements that are greater than 20.

Question – 02

Write a function '**extractEmails'** that takes a character vector, searches each element for substrings matching the pattern of an email address, and returns a cell array of email addresses in the order they first appear. For simplicity, assume that any word that contains @ - is an email address.

Test cases:

Test case 01:

- Input: 'Please contact john.doe@example.com for more info.'
- Output: {'john.doe@example.com'}

Test case 02:

- Input: 'List: alice@example.com, bob.smith@company.co.uk; carol@school.edu'
- Output: {'alice@example.com', 'bob.smith@company.co.uk', 'carol@school.edu'}

Test case 03:

- Input: 'The email address of the student is maxpayne44@iut-dhaka.edu'
- Output: {'maxpayne44@ iut-dhaka.edu'}

Question – 03

Create a function named **'sumDigits'** that returns the sum of the even digits of an integer given as input.

Test Case 1:

- Input: 1024
- Output: 6

Test Case 2:

- Input: 911001
- Output: 0

Test Case 3:

- Input: 464646111
- Output: 30

Question – 4

Write a function called '**longest_inc_con_seq**' that takes an array as input and returns the **length** of the longest increasing subsequence, given that all elements of the subsequence are strictly **increasing and are consecutive**.

For example, arr = [10, 9, 2, 5, 3, 7, 101, 18]

Here, there can be many increasing subsequences. Such as:

- [10, 101, 18] – increasing but not consecutive (x)
- [2, 5]
- [3, 7, 101]
- [7, 101]

You need to find the longest possible subsequence under the condition that the values are increasing and contiguous.

Test case – 01

- Input: [2, 5, 10, 3, 1]
- Output: 3

Test case – 02

- Input: [10, 20, 30, 40, 50]
- Output: 5

Test case – 03

- Input: [9, 6, 4, 5, 2, 1, 4, 4, 9, 11, 2, 5]
- Output: 3

Test case – 04

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

Test case – 05

- Input: [0, 8, 4, 12, 2, 10, 6, 14, 1, 9]
- Output: 2

Test case – 06

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