# DATA STRUCTURES

# FALL 2023

**LAB 03**

**Learning Outcomes**

In this lab you are expected to learn the following:

* Arrays

# Lab Tasks

## Task 01 (30 minutes)

Declare an array (size of array must be greater than 10) and initialize it by the user. Create two functions inside your header that take the array as an argument.

“int min(int \*a, int size)” and “int max(int \*a, int size)”.

One function must return the Maximum element having the highest value and the other function must return the Minimum element having the lowest value.

## Task 02 (20 minutes)

Write a program that searches for a given character inside a string and print the string from the point of match. Consider the following output for example.

Input: Enter a string: programming

Enter a character to search for: o

Output: ogramming

## Task 03(30 minutes)

Write a program to demonstrate the use of pointer to pointer. Make an array of characters (a word) by char \*word, another array of words (a sentence) using char \*\*sentence. Return the sentence using a double-pointer.

Function prototype: char\*\* words\_to\_sentence(int row, int col)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Word 1 | H | e | l | l | o |
| Word 2 | w | o | r | l | d |
| Word 3 | h | o | w |  |  |
| Word 4 | a | r | e |  |  |
| Word 5 | y | o | u |  |  |

## Task 04(60 minutes)

Design a custom <Template>C++ class called MyVector to create a dynamic array-like data structure. Your MyVector class should provide the following functionalities:

addElement: Implement a method to add elements to the end of the vector. Ensure that the vector dynamically resizes itself when it reaches capacity.

getElement: Create a method that allows users to access elements by their index. Ensure that it handles both valid and out-of-bounds index requests gracefully.

changeElement: Include a method that enables users to modify the value of an element at a specific index in the vector. Ensure it handles out-of-bounds indices and prevents memory violations.

removeElement: Implement a method to remove an element at a specified index, shifting the remaining elements accordingly and updating the vector's size.

Provide the complete code for your MyVector class and demonstrate its usage by adding elements of different data types, accessing elements at various indices, changing element values, and removing elements from an instance of MyVector.