

LAB Manual

PART A

(PART A : TO BE REFERRED BY STUDENTS)

Experiment No.01

A.1 Aim:

Introduction to Data Structures and implementation of Arrays

A.2 Prerequisite:

Prior knowledge of introduction to data structure.

A.3 Outcome:

After successful completion of this experiment students will be able to:

After successful completion of this experiment students will be able to:

1. Understand the concept of data structures
2. Know the implementation and use of arrays

A.4 Theory:

Data structure is logical or mathematical organization of data; it describes how to store the data and access data from memory. Actually in our programming data stored in main memory (RAM) and To develop efficient software or firmware we need to care about memory. To efficiently manage we required data structure.

There are two different types of data structure:

Linear Data Structure: In linear data structure data elements stored in sequential manner. Stack, Queue and Linked List are the types of linear data structure.

Non Linear Data Structure: In Non-Linear data structure data elements are not stored in the sequence manner. Tree and Graph are the type of non-linear data structure.

Array is a container which can hold a fix number of items and these items should be of the same type. Most of the data structures make use of arrays to implement their algorithms. Following are the important terms to understand the concept of Array.

- Element – Each item stored in an array is called an element.
- Index – Each location of an element in an array has a numerical index, which is used to identify the element.

Array Representation

Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.



Basic Operations

Following are the basic operations supported by an array.

Traverse – print all the array elements one by one.

Insertion – Adds an element at the given index.

Deletion – Deletes an element at the given index.

Search – Searches an element using the given index or by the value.

Update – Updates an element at the given index.

A.5 Procedure/Algorithm:

A.5.1 TASK 1:

Q1. Define the array ADT.

Q2. Explain what is meant by {row major order,column major order}.

Q3. Give a generic formula for the space complexity of an N dimensional array.

A.5.1 TASK 2:

Q4. Write a C/C++ program of array to perform following **(1D Array)**

- i. Find the sum and Average of all the elements of a 1-D array.
- ii. Find the sum and Average of all the elements of a 2-D array.
- iii. Find highest and lowest element in an array.
- iv. Insert and delete an element in an array.

Q5. Declare an array of size 20. Take the elements (integer values) of the array from the user.
The output should have below mentioned three results:

- a. Print the array in reverse order.
- b. Print the mean, mode and median of the array.
- c. Move all "zeroes" present in the array to the end of the array.

A.5.1 TASK 2:

Save and close the file and name it as **DS_Experiment01_YourName**.

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll No.	Name:
Class :	Batch :
Date of Experiment:	Date of Submission
Grade :	

B.1 Answers of Task to be written by student:

B.2 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

B.3 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.2)
