



## CET 214 – Data Structures & Algorithms

### Experiment # 3

#### Experiment Title

Operation on Arrays

#### Assessment of CLO(s): IV

Performed on 04-10-2024

Student Name			
Roll No.		Group	
Semester		Session	

Total (Max)	Criteria 1 (2.5)	Criteria 2 (2.5)	Criteria 3 (2.5)	Criteria 4 (2.5)	Total (10)
Marks Obtained					
Remarks (if any)					

#### Experiment evaluated by

Instructor's Name	Engr. Muhammad Asad Husain		
Date		Signature	

**Department of Engineering Technology**  
(UIT University)

**Course Code: CET214      Course Title: Data Structures & Algorithms      Course Credits: 2+1      Session: Fall 2024**

**Rubric for assessment criteria to perform experiment number 3.**

<b>Level Criteria</b>	<b>UNSATISFACTORY 1</b>	<b>COMPETENT 2</b>	<b>PROFICIENT 3</b>	<b>DISTINGUISHED 4</b>
<b>Capability of writing algorithm/ Procedure</b>	None of the steps are implemented of an algorithm.	Few steps are implemented correctly of an algorithm.	Most of the steps are implemented correctly of an algorithm.	All the steps are implemented correctly of an algorithm.
<b>Capability of writing Program</b>	Programs not completed.	Completeness of code, consistent variable naming and unformatted.	Completeness of code, inconsistent variable naming and well formatted.	Completeness of code, consistent variable naming and well formatted.
<b>Completion of target in Lab</b>	25% target has been completed	50% target has been completed	75% target has been completed	100% target has been completed
<b>Output</b>	None of the outputs are correct.	Few outputs have been found correctly.	Some of the outputs are correct and well formatted.	Most of the outputs are correct and well formatted.

## Practical Objective(s):

1. Learn various operation on arrays.
2. Implementation of array operations in C++.

## Theory

### Basic Operations in Arrays

The basic operations in the Arrays are insertion, deletion, searching, display, traverse, and update. These operations are usually performed to either modify the data in the array or to report the status of the array.

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.
- **Display** – Displays the contents of the array.

### Algorithm # 1: Insertion Operation

Following is an algorithm to insert elements into a Linear Array until we reach the end of the array

- |                |  |
|----------------|--|
| <b>Step 1.</b> | <b>Start</b>   |
| <b>Step 2.</b> | <b>Create an Array of a desired datatype and size.</b>         |
| <b>Step 3.</b> | <b>Initialize a variable 'i' as 0.</b>                         |
| <b>Step 4.</b> | <b>Enter the element at i<sup>th</sup> index of the array.</b> |
| <b>Step 5.</b> | <b>Increment i by 1.</b>                                       |
| <b>Step 6.</b> | <b>Repeat Steps 4 &amp; 5 until the end of the array.</b>      |
| <b>Step 7.</b> | <b>Stop</b>  |

### Algorithm # 2: Deletion Operation

Consider LA is a linear array with N elements and K is a positive integer such that  $K \leq N$ . Following is the algorithm to delete an element available at the Kth position of LA.

- |                |  |
|----------------|--|
| <b>Step 1.</b> | <b>Start</b>                               |
| <b>Step 2.</b> | <b>Set J = K</b>                           |
| <b>Step 3.</b> | <b>Repeat steps 4 and 5 while J &lt; N</b> |
| <b>Step 4.</b> | <b>Set LA[J] = LA[J + 1]</b>               |
| <b>Step 5.</b> | <b>Set J = J+1</b>                         |
| <b>Step 6.</b> | <b>Set N = N-1</b>                         |
| <b>Step 7.</b> | <b>Stop</b>                                |

### Algorithm # 3: Search Operation

Consider LA is a linear array with N elements and K is a positive integer such that  $K \leq N$ . Following is the algorithm to find an element with a value of ITEM using sequential search.

- Step 1.**      **Start**
- Step 2.**      **Set  $J = 0$**
- Step 3.**      **Repeat steps 4 and 5 while  $J < N$**
- Step 4.**      **IF  $LA[J]$  is equal ITEM THEN GOTO STEP 6**
- Step 5.**      **Set  $J = J + 1$**
- Step 6.**      **PRINT J, ITEM**
- Step 7.**      **Stop**

#### **Algorithm # 4: Traversal Operation**

Following is the algorithm to traverse through all the elements present in a Linear Array

- Step 1.**      **Start**
- Step 2.**      **Initialize an Array of certain size and datatype.**
- Step 3.**      **Initialize another variable 'i' with 0.**
- Step 4.**      **Print the ith value in the array and increment i.**
- Step 5.**      **Repeat Step 4 until the end of the array is reached.**
- Step 6.**      **End**

#### **Algorithm # 5: Update Operation**

Consider LA is a linear array with N elements and K is a positive integer such that  $K \leq N$ . Following is the algorithm to update an element available at the Kth position of LA.

- Step 1.**      **Start**
- Step 2.**      **Set  $LA[K-1] = \text{ITEM}$**
- Step 3.**      **Stop**

#### **Algorithm # 6: Display Operation**

Consider LA is a linear array with N elements. Following is the algorithm to display an array element.

- Step 1.**      **Start**
- Step 2.**      **Print all the elements in the Array**
- Step 3.**      **Stop**

### **Do It Yourself:**

1. Implement Algorithm 1 to 6 in C++.