



THE UNIVERSITY
OF LAHORE
**ISLAMABAD
CAMPUS**

Data Structure and Algorithm (CS09203)

Lab Report

Name: Muhammad Umer
Registration #: CSU-F16-104
Lab Report #: 01
Dated: 26-03-2018
Submitted To: Mr. Usman Ahmed

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 1

Introduction to Arrays and its operation

Objective

The objectives of this lab session are to understand the basic and various operations on arrays in C++.

Software Tool

1. Window 7 (32-bit)
2. Sublime Text Editor
3. Dev C++

Theory

We have already studied array in our computer programming course. We would be using the knowledge we learned there to implement different operation on arrays.

Traversing Linear Arrays:-

Let A be the collection of data elements stored in the memory of the computer. Suppose we want to print the contents of each element of A or suppose we want to count the number of elements of A with a given property. This can be accomplished by traversing A that is by accessing and Processing each element of A exactly once.

The following algorithm traverses a linear array. The simplicity of the algorithm comes from the fact that LA is a linear structure. Other linear structures such as linked list can also be easily traversed. On the other hand the traversal of non-linear structures such as trees and graphs is considerably more complicated.

Algorithm:-

(Traversing a Linear Array) Here LA is a linear Array with lower Bound LB and upper Bound UB. This algorithm traverses LA.

Applying an operation PROCESS to each element of LA.

[Initialize Counter] Set $X=LB$.

1. Repeat Step 3 and 4 **while** $K \leq UB$.

[Visit element] Apply PROCESS to $LA[X]$.

[Increase Counter] Set $X=X+1$.

[End of Step 2 Loop]

5. Exit.

Inserting and Deleting:-

Let A be a collection of data elements in the memory of computer. Inserting refers to the operation of adding another element to the collection A and deleting refers to the operation of removing one of the elements from A. Here we discuss the inserting and deleting when A is a linear array.

Inserting an element at the end of the linear array can be easily done provided the memory space allocated for the array is large enough to accommodate the additional element. On the other hand suppose we need to insert an element in the middle of the array. Then on average half of the elements must be moved downward to the new location to accommodate the new element and keep the order of other elements.

Similarly deleting the element at the end of an array presents no difficulties but deleting the element somewhere in the middle of the array would require that each subsequent element be moved one location upward in order to fill up the array.

Algorithm of Insertion operation:-

(Inserting into Linear Array) INSERT (LA, N, K, ITEM)

Here LA is a linear array with N elements and K is a positive integer such that $K \leq N$. This algorithm inserts an element ITEM into the Kth position in LA.

1. [Initialize Counter] Set $J = N$.
2. Repeat Step 3 and 4 while $J \geq K$.
3. [Move Jth element downward] Set $LA[J+1] = LA[J]$.
4. [Decrease Counter] Set $J = J - 1$.
- End of Step 2 Loop.
5. [Insert element] Set $LA[K] = \text{ITEM}$.
6. [Reset N] Set $N = N + 1$.
7. Exit.

Lab Task:-

Write a C++ program to implement all the above described algorithms and display the following menu and ask the user for the desired operation.

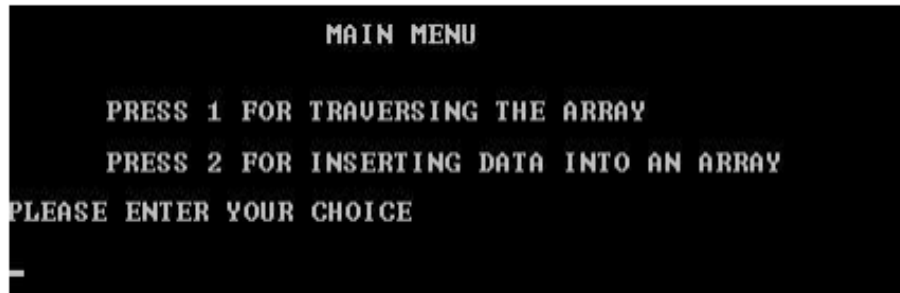


Figure 1: The output of the program

The program should have the option for reusing it after you have completed the desired task.

Solution:-

```
#include<iostream>
#include<conio.h>
using namespace std;
void travers_array(int array[] , int n){
    cout<<"\n\n";
    for(int x=0; x<n; x++){
        cout<<array[x]<<" ";
    }
    cout<<"\n\nPRESS_ANY_KEY_TO_CONTINUE..." ;
    getch();
}

void insert(int array[] ,int& n, int k, int item){
    for(int j=n; j>=k; j--){
        array[j+1] = array[j];
    }
    array[k] = item;
    n = n+1;
    cout<<"\nVALUE_ADDED_SUCCESSFULLY" ;
    cout<<"\n\nPRESS_ANY_KEY_TO_CONTINUE..." ;
    getch();
}

void delete_value(int array[] , int& n, int k){
```

```

        for(int j=k; j<=sizeof(array); j++){
            array[j] = array[j+1];
        }
        n = n-1;
        cout<<"\nVALUE_DELETED_SUCCESSFULLY";
        cout<<"\n\nPRESS_ANY_KEY_TO_CONTINUE...";
        getch();
    }
    int main()
    {
        int choice, n = 5, arr[n], location, value;
        cout<<"\t\t\tENTER_5_ELEMENTS\n\n";
        for(int i=0; i<5; i++)
        {
            cout<<"Enter_"<<i+1<<"_Value: ";
            cin>>arr[i];
        }
        up:
        system("cls");
        cout<<"\t\t\tMAIN_MENU\n\n";
        cout<<"\tPRESS_1_FOR_TRAVERSING_THE_ARRAY\n\n";
        cout<<"\tPRESS_2_FOR_INSERTING_DATA_INTO_ARRAY\n\n";
        cout<<"\tPRESS_3_FOR_DELETING_DATA_FROM_ARRAY\n\n";
        cout<<"\tPRESS_4_TO_EXIT\n\n";
        cout<<"PLEASE_ENTER_YOUR_CHOICE\n\n";
        cin>>choice;
        if(choice == 1){
            travers_array(arr, n);
            goto up;
        }
        else if(choice == 2){
            cout<<"\n\nENTER_LOCATION_TO_INSERT: ";
            cin>>location;
            cout<<"\n\nENTER_VALUE_TO_INSERT: ";
            cin>>value;
            insert(arr, n, location-1, value);
            goto up;
        }
        else if(choice == 3){
            cout<<"\n\nENTER_LOCATION_TO_DELETE: ";

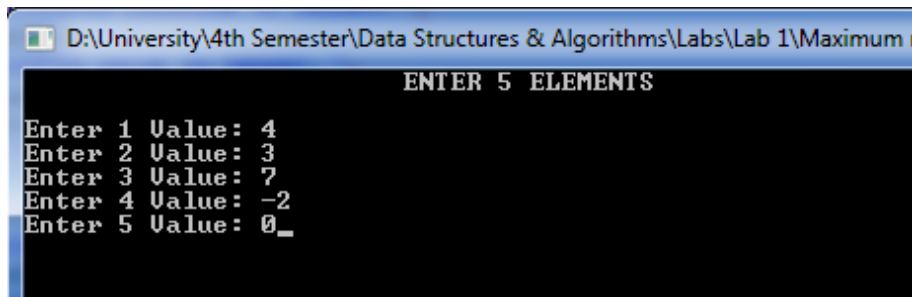
```

```

        cin>>location;
        delete_value(arr, n, location-1);
        goto up;
    }
    else if(choice == 4){
        exit(0);
    }
    else{
        cout<<"\n\nWRONG_CHOICE!";
        cout<<"\n\nPRESS_ANY_KEY_TO_CHOOSE_AGAIN...";
        getch();
        goto up;
    }
    return 0;
}

```

Output:-

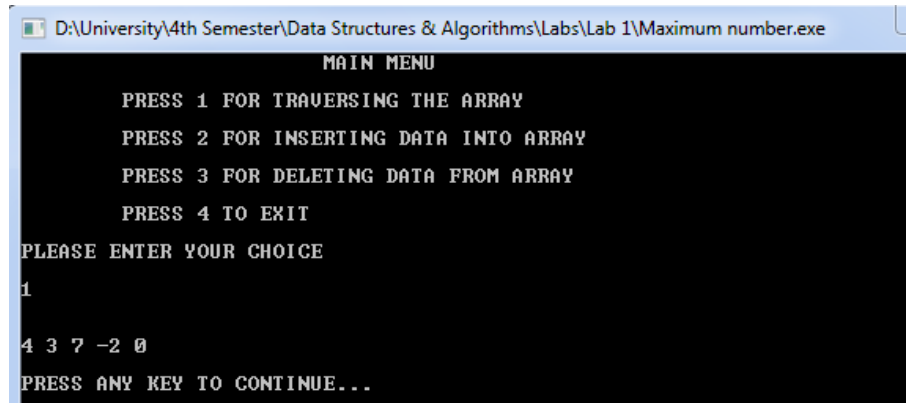


```

D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum
ENTER 5 ELEMENTS
Enter 1 Value: 4
Enter 2 Value: 3
Enter 3 Value: 7
Enter 4 Value: -2
Enter 5 Value: 0_

```

Figure 2: Getting input in the start



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe

MAIN MENU

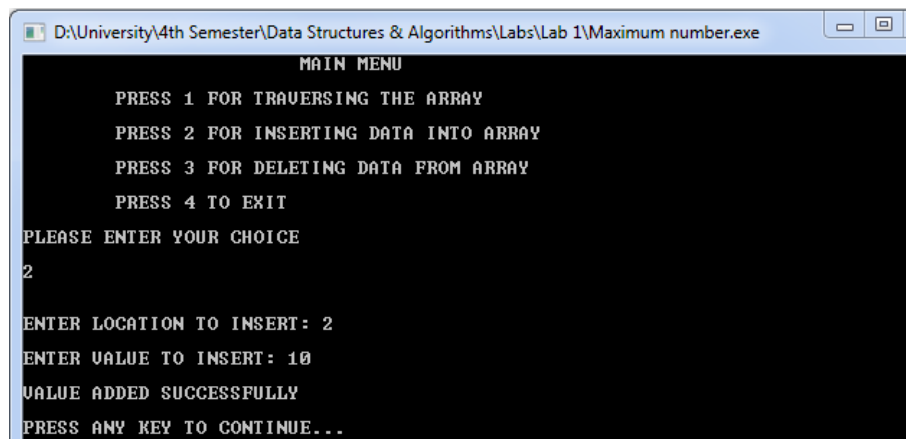
PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT

PLEASE ENTER YOUR CHOICE
1

4 3 7 -2 0

PRESS ANY KEY TO CONTINUE...
```

Figure 3: Main Menu and Traversing the array



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe

MAIN MENU

PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT

PLEASE ENTER YOUR CHOICE
2

ENTER LOCATION TO INSERT: 2
ENTER VALUE TO INSERT: 10
VALUE ADDED SUCCESSFULLY

PRESS ANY KEY TO CONTINUE...
```

Figure 4: Inserting item in the array

```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe

MAIN MENU

PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT
PLEASE ENTER YOUR CHOICE
1
4 10 3 7 -2 0
PRESS ANY KEY TO CONTINUE...
```

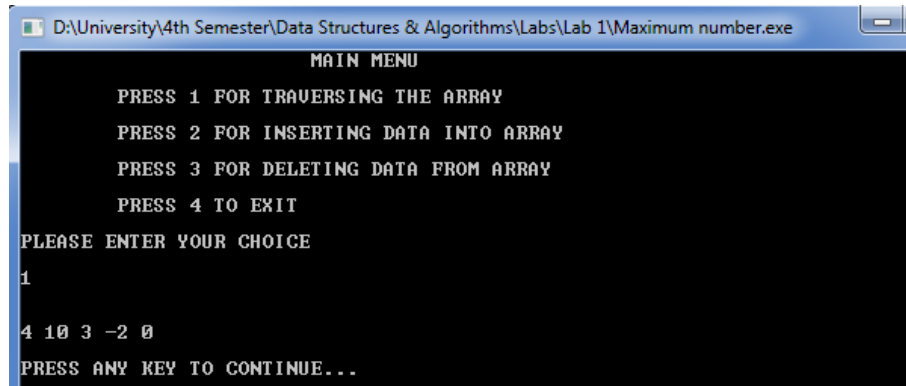
Figure 5: Traversing after inserting in the array

```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe

MAIN MENU

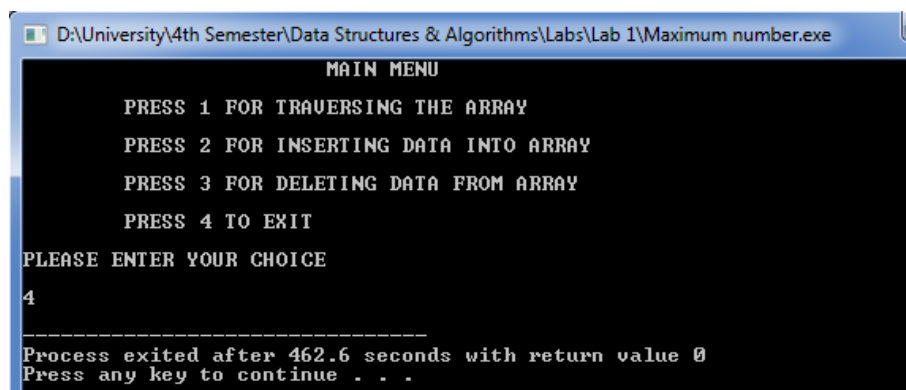
PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT
PLEASE ENTER YOUR CHOICE
3
ENTER LOCATION TO DELETE: 4
VALUE DELETED SUCCESSFULLY
PRESS ANY KEY TO CONTINUE...
```

Figure 6: Deleting item from the array



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe
MAIN MENU
PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT
PLEASE ENTER YOUR CHOICE
1
4 10 3 -2 0
PRESS ANY KEY TO CONTINUE...
```

Figure 7: Traversing after deleting from array



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 1\Maximum number.exe
MAIN MENU
PRESS 1 FOR TRAVERSING THE ARRAY
PRESS 2 FOR INSERTING DATA INTO ARRAY
PRESS 3 FOR DELETING DATA FROM ARRAY
PRESS 4 TO EXIT
PLEASE ENTER YOUR CHOICE
4
-----
Process exited after 462.6 seconds with return value 0
Press any key to continue . . .
```

Figure 8: Exit

Source Code:- <https://github.com/umerayan/Semester-4-Labs.git>

Conclusion:- Operations on arrays is a good practice to start learning Data Structures. We have already studied array in our computer programming course. In this lab we have performed different operations on arrays i.e. insertion, deletion and traversing. The above program can perform all these three operations on arrays. The program is coded in C++ and open source for everyone to use at my github account (The Link provided above)

(Concerned Teacher/Lab Engineer)