Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Fall, Year : 2024), B.Sc. in CSE (Day)

Lab Report NO 01
Course Title: Data Structures Lab
Course Code: CSE 206 Section: 232 D10

<u>Student Details</u>

	Name	ID
1.	Md.Rabby Khan	213902037

Lab Date	**: 10-09-2024**
Submission Date	**: 14-09-2024**
Course Teacher’s Name	**: Fatema Akter**

<u>Lab Report Status</u>
Marks: ………………………………… **Signature:.....................**
Comments:... **Date:..............................**

1. TITLE OF THE LAB REPORT EXPERIMENT

Write a C/C++ program to implement an array using user input, with array insertion at any specific position and deletion from any specific position.

2. OBJECTIVES/AIM

- Implement functions to insert and delete elements in an array.
- Test these functions to ensure they work correctly.
- Learn how these operations affect the array's data and structure.
- Gain a deeper understanding of array manipulation in C++ programming.

3. PROCEDURE / ANALYSIS / DESIGN

PSEUDOCODE:

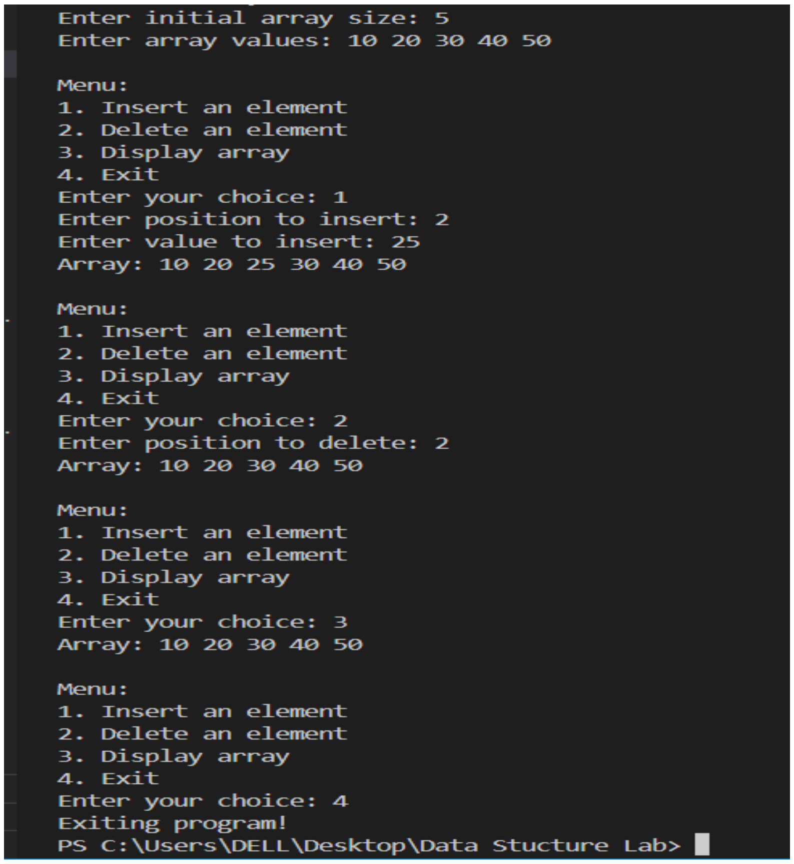
```
1. Start
2. Input size of array
3. Input array elements
4. Display Menu: Insert, Delete, Display, Exit
5. For Insertion:
6. Check if position is valid
7. Shift elements to the right
8. Insert value
9. Increase size
10. For Deletion:
11. Check if position is valid
12. Shift elements to the left
13. Decrease size
14. Display array contents
15. Repeat until Exit
16. End
```

.

4. IMPLEMENTATION

```
#include<iostream>
using namespace std;

// Function to display the contents of the array
void displayArray(const int arr[], int size)
{
    cout << "Array: ";
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
// Function to insert an element at a specific position in the array
void insertAtPosition(int arr[], int &size, int value, int pos)
{
    if (pos < 0 || pos > size)
    {
        cout << "Invalid position!" << endl;
        return;
    }

    for (int i = size; i > pos; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[pos] = value;
    size++;
}
// Function to delete an element from a specific position in the array
void deleteFromPosition(int arr[], int &size, int pos)
{
    if (pos < 0 || pos >= size)
    {
        cout << "Invalid position!" << endl;
        return;
    }
    for (int i = pos; i < size - 1; i++)
    {
        arr[i] = arr[i + 1];
    }
    size--;
}

void performAction(int arr[], int &size, int choice)
{
    int pos, value;

    switch (choice)
    {
    case 1:
        cout << "Enter position to insert: ";
        cin >> pos;
        cout << "Enter value to insert: ";
        cin >> value;
        insertAtPosition(arr, size, value, pos);
        displayArray(arr, size);
        break;

    case 2:
        cout << "Enter position to delete: ";
        cin >> pos;
        deleteFromPosition(arr, size, pos);
        displayArray(arr, size);
        break;
    case 3:
        displayArray(arr, size);
        break;
    default:
        cout << "Invalid choice!" << endl;
        break;
    }
}
int main()
{
    int size, choice;
    int arr[100];  // Array with a maximum size of 100

    // Prompt user for initial array size and values
    cout << "Enter initial array size: ";
    cin >> size;

    cout << "Enter array values: ";
    for (int i = 0; i < size; i++)
    {
        cin >> arr[i];
    }
    do
    {
        cout << "\nMenu:\n";
        cout << "1. Insert an element\n";
        cout << "2. Delete an element\n";
        cout << "3. Display array\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        if (choice != 4)
        {
            performAction(arr, size, choice);
        }
    }
    while (choice != 4);

    cout << "Exiting program!" << endl;
    return 0;
}
```

5. TEST RESULT / OUTPUT



```
Enter initial array size: 5
Enter array values: 10 20 30 40 50

Menu:
1. Insert an element
2. Delete an element
3. Display array
4. Exit
Enter your choice: 1
Enter position to insert: 2
Enter value to insert: 25
Array: 10 20 25 30 40 50

Menu:
1. Insert an element
2. Delete an element
3. Display array
4. Exit
Enter your choice: 2
Enter position to delete: 2
Array: 10 20 30 40 50

Menu:
1. Insert an element
2. Delete an element
3. Display array
4. Exit
Enter your choice: 3
Array: 10 20 30 40 50

Menu:
1. Insert an element
2. Delete an element
3. Display array
4. Exit
Enter your choice: 4
Exiting program!
PS C:\Users\DELL\Desktop\Data Stucture Lab>
```

6. ANALYSIS AND DISCUSSION

The C++ program successfully achieved the lab objectives by allowing insertion and deletion of elements in an array at specific positions. It manually shifts elements to maintain array structure, highlighting how static arrays function. Both insertion and deletion operations performed as expected, with appropriate validation for edge cases such as invalid positions.

The program's simplicity makes it effective for understanding basic array manipulation, though improvements could be made by incorporating dynamic arrays for better memory management. Overall, the experiment demonstrated efficient array operations and reinforced the importance of memory handling in static arrays.

7. SUMMARY

This experiment demonstrated how to manipulate arrays using manual insertion and deletion techniques. It solidified my understanding of basic array operations and showcased the importance of careful memory management when working with static arrays in C++.