**EXPERIMENT-1**

**AIM:**

Following operations on menu driven array:

1. Add elements to array.
2. Delete element by searching or by position.
3. Separate Odd Evens.
4. Sum of elements.
5. Average.
6. Max & Min.
7. Reverse of the array.

**Code:**

#include <iostream>

using namespace std;

void display\_elements\_arr(int x, int \*p);

void ToDo()

{

    cout << "1.Insert Element.\n2.Delete Element.\n3.Seperate Odd Even.\n4.Sum of all elements.\n"

         << "5.Average of all elements.\n6.Display Max & Min.\n7.Reverse of array.\n8.Display arr.\n9.Exit.\n";

}

int \*Insert\_Element(int size, int arr[])

{

    int element, position;

    cout << "Enter the value to be added: ";

    cin >> element;

    cout << "Enter the position: ";

    cin >> position;

    for (int i = size - 1; i > position; i--)

    {

        arr[i] = arr[i - 1];

    }

    arr[position] = element;

    return arr;

}

int \*Delete\_Element(int size, int arr[])

{

    int position, number\_to\_delete, choice;

    cout << "What to do: \n1. Delete element at a given position. \n2. Search and delete a number.";

    cin >> choice;

    if (choice == 1)

    {

        cout << "Enter position to delete: ";

        cin >> position;

        for (int i = position; i < size - 1; i++)

        {

            arr[i] = arr[i + 1];

            cout << i << endl;

        }

    }

    else if (choice == 2)

    {

        cout << "Enter number to search and delete: ";

        cin >> number\_to\_delete;

        for (int i = 0; i <= size; i++)

        {

            if (arr[i] == number\_to\_delete)

            {

                for (int j = i; j <= size; j++)

                {

                    arr[j] = arr[j + 1];

                }

            }

        }

    }

    return arr;

}

void \*Seperate\_Odd\_Even(int arr[], int size)

{

    int No\_Of\_Evens = 0, No\_Of\_Odds = 0;

    for (int i = 0; i < size; i++)

    {

        if (arr[i] % 2 == 0)

        {

            No\_Of\_Evens++;

        }

        else

        {

            No\_Of\_Odds++;

        }

    }

    int even\_arr[No\_Of\_Evens], odd\_arr[No\_Of\_Odds], j = 0, k = 0;

    for (int i = 0; i < size; i++)

    {

        if (arr[i] % 2 == 0)

        {

            even\_arr[j] = arr[i];

            j++;

        }

        else

        {

            odd\_arr[k] = arr[i];

            k++;

        }

    }

    cout << "Even:\n";

    display\_elements\_arr(j, even\_arr);

    cout << "\n\nOdd:\n";

    display\_elements\_arr(k, odd\_arr);

    cout << endl;

}

void Sum(int arr[], int size)

{

    int sum = 0;

    for (int i = 0; i < size; i++)

    {

        sum += arr[i];

    }

    cout << "\n Sum = " << sum << endl;

}

void Average(int arr[], int size)

{

    int sum = 0;

    for (int i = 0; i < size; i++)

    {

        sum += arr[i];

    }

    cout << "\n Average = " << sum / size << endl;

}

void MaxMin(int arr[], int size)

{

    int max = arr[1], min = arr[0];

    for (int i = 0; i < size; i++)

    {

        if (arr[i] > max)

            max = arr[i];

        if (arr[i] < min)

            min = arr[i];

    }

    cout << "Maximum = " << max;

    cout << "\nMinimum = " << min << endl;

}

void Reverse(int arr[], int size)

{

    cout << "\nArray in reverse order:\n";

    cout << "| ";

    for (int i = size - 1; i >= 0; i--)

    {

        cout << arr[i] << " | ";

    }

    cout << endl;

}

void display\_elements\_arr(int x, int \*p)

{

    cout << "| ";

    for (int i = 0; i < x; i++)

    {

        cout << \*p << " | ";

        \*p++;

    }

    cout << endl;

}

void RandomNumFill(int arr[], int size)

{

    for (int i = 0; i < size; i++)

    {

        arr[i] = 10 + (rand() % 90);

    }

}

int main()

{

    int n, choice;

    cout << "Enter the size of an array : ";

    cin >> n;

    int \*arr = new int(n);

    RandomNumFill(arr, n);

    display\_elements\_arr(n, arr);

    ToDo();

    while (choice != 8)

    {

        cout << "\nEnter your choice : ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            Insert\_Element(n, arr);

            break;

        case 2:

            Delete\_Element(n, arr);

            break;

        case 3:

            Seperate\_Odd\_Even(arr, n);

            break;

        case 4:

            Sum(arr, n);

            break;

        case 5:

            Average(arr, n);

            break;

        case 6:

            MaxMin(arr, n);

            break;

        case 7:

            Reverse(arr, n);

            break;

        case 8:

            display\_elements\_arr(n, arr);

            exit(0);

            break;

        case 9:

            exit(0);

            break;

        default:

            cout << "INVALID CHOICE!!";

        }

    }

    return 0;

}