ASSISMENT 1

Student Name: Bhaskar Kumar **UID: 23BCS14337**
Branch:BE-CSE **Section/Group: KRG-3B**
Semester: 5 **Date of Performance:22-07-25**
Subject Name: DAA
Subject Code:23CSH-301

1. **Aim: WAP for array insertion and deletion.**

2. **Objective: To develop a program showing insertion and deletion in an array (using C++) and the algorithm analysis of**

   Insertion at begin, end or any given position.

   Deletion at begin, end or any given position.

3. **CODE:**

```
#include<bits/stdc++.h>
using namespace std;

//Insertions in  an Array
void insertionatbegin(vector<int>& a)
{
   int n = a.size();

   int num;
   cout<<"Enter the number to insert at beging "<<endl;
   cin>>num;

   a.resize(n+1);

   for(int i = n; i > 0; i--)
   {
      a[i] = a[i-1];
   }
   a[0] = num;
}

void insertionatend(vector<int>& a)
```

```
{
   int n = a.size();

   int num;
   cout<<"Enter the number to insert at end "<<endl;
   cin>>num;

   a.resize(n+1);

   a[n] = num;
}


void insertionatpos(vector<int>& a)
{
   int n = a.size();

   int num;
   cout<<"Enter the number to insert at given postion "<<endl;
   cin>>num;

   a.resize(n+1);
   int pos;
   cout<<"Enter the postition "<<endl;
   cin>>pos;

   for(int i = n; i>=pos; i--)
   {
      a[i] = a[i-1];
   }
   a[pos-1] = num;

}


//Deletion from an array
void deleteformbegin(vector<int>& a)
{
   int n = a.size();
   int temp;
   temp = a[0];

   for(int i = 1; i<n; i++)
   {
      a[i-1] = a[i];
   }
   a.resize(n-1);
}
```

```
void deletefromend(vector<int>& a)
{
   int n = a.size();
   int temp;
   temp = a[n-1];

   a.resize(n-1);
}

void deletefrompos(vector<int>& a)
{
   int n = a.size();
   int temp;

   int pos;
   cout<<"enter the position "<<endl;
   cin>>pos;

   temp = a[pos-1];

   for(int i = pos-1; i<n-1; i++)
   {
      a[i] = a[i+1];
   }
   a.resize(n-1);
}

//Driver Code
int main()
{
   int n;
   cout<<"Enter the number of elements "<<endl;
   cin>>n;
   vector<int> arr(n);

   cout<<"Enter the elements "<<endl;
   for(int i = 0; i<n; i++)
   {
      cin>>arr[i];
   }

   for(int x: arr)
      cout<<x<<" ";

   cout<<endl;
```

```
 //Insertion function calls
    insertionatbegin(arr);
    for(int x: arr)
        cout<<x<<" ";

    cout<<endl;
    insertionatend(arr);
    for(int x : arr)
        cout<<x<<" ";

    cout<<endl;
    insertionatpos(arr);
    for(int x: arr)
        cout<<x<<" ";


 //delete function calls
    deleteformbegin(arr);
    cout<<"Delete from Begin"<<endl;
    for(int x:arr)
        cout<<x<<" ";

    cout<<endl;

    deletefromend(arr);
    cout<<"Delete from end"<<endl;
    for(int x:arr)
        cout<<x<<" ";

    cout<<endl;

    deletefrompos(arr);
    cout<<"Delete from given position "<<endl;
    for(int x:arr)
        cout<<x<<" ";
}
```

OUTPUT:

Insertion

```
Output
Enter the number of elements
5
Enter the elements
1
2
3
4
5
1 2 3 4 5
Enter the number to insert at beging
9

9 1 2 3 4 5
Enter the number to insert at end
7
9 1 2 3 4 5 7
Enter the number to insert at given postion
8
Enter the postition
5
9 1 2 3 8 4 5 7
```

Deletion

```
Output
Enter the number of elements
6
Enter the elements
1
2
3
4
5
6
1 2 3 4 5 6
Delete from Begin
2 3 4 5 6
Delete from end
2 3 4 5
enter the position
3
Delete from given position
2 3 5
```

4.Learning Outcomes:

- Inserting at a **specific index** requires shifting all elements **rightward** from that index.
- Inserting at the **end** is direct if there's space: `arr[size] = value; size++;`.
- In **static arrays**, size is fixed — you may need to create a **new array** if it's full.
- To delete from a **specific index**, shift all elements **leftward** after that index.
- Deleting the **last element** is easy: just reduce the size: `size--;`.
- Array size doesn't shrink automatically — memory stays allocated.

5. Algorithm Analysis: