Q.1 WAP to insert a new element at end as well as at given position in an array.

```
#include<iostream>

using namespace std;

void display(int arr[],int size){

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

    cout << arr[i]<< " ";

 }}

int main(){

 int arr[100];

 int size,value,position;

 cout << "enter size of array : " ;

 cin >> size;          //taking size of array

 cout << "enter "<< size<< " elements in array" <<endl;

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

  cout << "enter element for index  "<<i << " :";

  cin >> arr[i];            //adding elements to the array

 }

 cout << "entered elements are : ";

 display(arr,size); //displaying added elements

 cout << endl;

 cout << "enter element to add at end of an array : ";

 cin >>value;

 arr[size++]=value;     // adding element at last of array and increasing size

  cout << "elemets after adding in end: ";

 display(arr,size);

 cout <<endl;

 cout << "enter position to enter in form of index : ";

 cin >> position;
```

```
 cout << "enter value to enter at positon " <<position <<" :";
 cin >>value;
 size++;
 for(int i=size-1;i>position-1;i--){ //shifting elements from postition to pos+1
  cout << i << "at this instance is "<< i<< endl;
  arr[i]=arr[i-1];
 }
 arr[position-1]=value;
 cout << "elemets after adding at postion  "<< position<< " :" ;
 display(arr,size);
  return 0;
}
```

OUTPUT :-

```
enter size of array : 5
enter 5 elements in array
enter element for index  0 :1
enter element for index  1 :2
enter element for index  2 :3
enter element for index  3 :4
enter element for index  4 :5
entered elements are : 1 2 3 4 5
enter element to add at end of an array : 6
elemets after adding in end: 1 2 3 4 5 6
enter position to enter : 1
enter value to enter at positon 1 :0
elemets after adding at postion  1 :0 1 2 3 4 5 6
```

Q.2 WAP to delete an element from a given whose value is given or whose position is given.

```
#include<iostream>
using namespace std;
void display(int arr[],int size){
  for(int i=0;i<size;i++){
    cout << arr[i]<< " ";
 }}
int main(){
 int arr[100];
 int size,value,position;
 bool found=false;
 cout << "enter size of array : " ;
 cin >> size;          //taking size of array
 cout << "enter "<< size<< " elements in array" <<endl;
 for(int i=0;i<size;i++){
  cout << "enter element for index  "<<i << " : ";
  cin >> arr[i];       //adding elements to the array
 }
 cout << "entered elements are : ";
 display(arr,size);     //displaying added elements
 cout << endl;
 cout << "enter element to be Deleted : ";
 cin >>value;
 cout << "deleting , NOTE :- only first matching value is deleted"<< endl;
 for (int i=0;i<size;i++){
   if(arr[i]==value){
     position=i;
     found=true;
```

```
    break;
   }

}
if(found){

 for(int i=position ; i<size;i++){
   arr[i]=arr[i+1];
 }
 size--;
  cout << "array after deleting  "<<value<< " : ";
display(arr,size);
cout <<endl;

}else{
 cout << "no such element in the array given "<<endl;
}
cout << "enter position to delete : ";
cin >> position;
if(position>size||position <1){
 cout << "index out of bound no such positon " <<endl;
}else{
  for(int i=position-1 ; i<size;i++){
   arr[i]=arr[i+1];
 }
 size --;
  cout << "array after deleting  value at position : "<<position<< " : ";
display(arr,size);
}
```

```
  return 0;
}
```

OUTPUT :-

```
enter size of array : 5
enter 5 elements in array
enter element for index  0 : 1
enter element for index  1 : 2
enter element for index  2 : 3
enter element for index  3 : 4
enter element for index  4 : 5
entered elements are : 1 2 3 4 5
enter element to be Deleted : 5
deleting , NOTE :- only first matching value is deleted
array after deleting  5 : 1 2 3 4
enter position to delete : 2
array after deleting  value at position : 2 : 1 3 4
```

Q.3 WAP to find the location of a given element using Linear Search.

```
#include<iostream>
using namespace std;
void display(int arr[],int size){
  for(int i=0;i<size;i++){
    cout << arr[i]<< " ";
}}
int main(){
 int arr[100];
 int size,target;
 bool found=false;
 cout << "enter size of array : " ;
 cin >> size;   //taking size of array
 cout << "enter "<< size<< " elements in array" <<endl;
 for(int i=0;i<size;i++){
  cout << "enter element for index  "<<i << " :";
  cin >> arr[i];   //adding elements to the array
 }
 cout << "entered elements are : ";
 display(arr,size);   //displaying added elements
 cout << endl;
 cout << "enter target to search : ";
 cin >>target;
 for(int i=0;i<size;i++){
  if(arr[i]==target){
    found=true;
    cout << "target found at index :"<<i <<" or at position : "<<i+1<<endl;
    break;
```

```
  }}
 if(!found){
  cout << "target not found"<<endl;
 }
  return 0;
}
```

OUTPUT :-

```
enter size of array : 10
enter 10 elements in array
enter element for index  0 :1
enter element for index  1 :2
enter element for index  2 :2
enter element for index  3 :4
enter element for index  4 :84
enter element for index  5 :54
enter element for index  6 :54
enter element for index  7 :65
enter element for index  8 :47
enter element for index  9 :24
entered elements are : 1 2 2 4 84 54 54 65 47 24
enter target to search : 47
target found at index :8 or at position : 9
```

Q.4 WAP to find the location of a given element using Binary Search.

```
#include<iostream>
using namespace std;
void display(int arr[],int size){
  for(int i=0;i<size;i++){
    cout << arr[i]<< " ";
  }}
int main(){
 int arr[100];
 int size,target,lower,upper,mid;
 bool found=false;
 cout << "enter size of array : " ;
 cin >> size; //taking size of array
 cout << "enter "<< size<< " elements in array" <<endl;
 for(int i=0;i<size;i++){
  cout << "enter element for index  "<<i << " :";
  cin >> arr[i]; //adding elements to the array
 }
 cout << "entered elements are : ";
 display(arr,size); //displaying added elements
 cout << endl;
 cout << "enter target to search : ";
 cin >>target;
 lower=0;
 upper=size-1;
 while(upper>=lower){
  mid=lower+(upper-lower)/2;
if(arr[mid]==target){
  cout <<" target found at index : "<<mid << " positon : "<<mid+1 <<endl;
```

```
  found=true;
  break;
}
if(arr[mid]>target){
  upper=mid-1;
}else if (arr[mid]<target){
  lower=mid+1;
} }
 if(!found){
  cout << "target not found"<<endl;
 }
  return 0;
}
```

OUTPUT :-

```
enter size of array : 5
enter 5 elements in array
enter element for index  0 :1
enter element for index  1 :2
enter element for index  2 :3
enter element for index  3 :4
enter element for index  4 :5
entered elements are : 1 2 3 4 5
enter target to search : 4
target found at index : 3 positon : 4
```

Q.5 WAP to implement push and pop operation on a stack using linear array.

```
#include <iostream>
using namespace std;
class stack{
  private :
  int  max =3;
  int arr[3];
  int top;
   public :
  stack(){
   top=0;
  }
  void push(int item){
   if(top==max){
     cout << "stack overflow exception "<< endl;
     return ;  }
   cout << item << " added to stack"<<endl;
   arr[top]=item;
   top++; }
  void pop(){
   if(top-1<0){
     cout << "stack underflow exception "<<endl;
     return ;}
      cout <<  "top popped from stack"<<endl;
   top--;}
  void display(){
   for(int i=0;i<top;i++){
     cout << arr[i]<< " ";}}};
```

```
int main(){
  stack st1;
  st1.push(4);
  st1.push(5);
  st1.push(6);
  st1.push(9);
  cout << "elements in stack : "<<endl;
  st1.display();
  cout <<endl;
  st1.pop();
  st1.pop();
  st1.pop();
   st1.pop();
   cout << "elements in stack : "<<endl;
  st1.display();
  return 0;
}
```

OUTPUT :-

```
4 added to stack
5 added to stack
6 added to stack
stack overflow exception
elements in stack :
4 5 6
top popped from stack
top popped from stack
top popped from stack
stack underflow exception
elements in stack :
```