

## Searching

### 1. Linear Search:-

---

```
void linearSearch(int *arr,int noe)
{
    int number;
    cout<<"Enter number to search:
    "<<endl;
    cin>>number;
    bool found=false;
    for(int i=0;i<noe;i++)
    {
        if(number==arr[i])
        {
            found=true;
            break;
        }
    }
    if(found)
    {
        cout<<"Element found"<<endl;
    }
    else
    {
        cout<<"Element not found"<<endl;
    }
}
```

---

### 2. Binary Search:-

---

```
int binarySearch(int *arr,int size)
{
    int low,mid,high;
    int stack[2][2],top;
    int number;
    cout<<"Enter number to search: ";
    cin>>number;
    top=2;
    top--;
    stack[0][top]=0;
    stack[1][top]=size-1;
    while(top!=2)
    {
        low=stack[0][top];
        high=stack[1][top];
```

```
        top++;
        mid=(low+high)/2;
        if(number==arr[mid])
        {
            return 1;
        }
        if(number>arr[mid])
        {
            top--;
            stack[0][top]=mid+1;
            stack[1][top]=high;
        }
        if(number<arr[mid])
        {
            top--;
            stack[0][top]=low;
            stack[1][top]=mid;
        }
        if(mid==low || mid==high)
        {
            break;
        }
    }
    return -1;
}
```

---

### 3. Jump Search :-

---

```
void jumpSearch(int *arr,int size)
{
    int number;
    cout<<"Enter number to search for: ";
    cin>>number;
    int steps=sqrt(number);
    steps--;
    bool found=false;
    int got=0;
    for(int i=0;i<size;i=i+steps)
    {
        if(number==arr[i])
        {
            found=true;
            break;
        }
    }
```

## Searching

```
}
if(number<arr[i])
{
got=linearSearch(arr,i,size,number);
break;
}
}
if(found)
{
cout<<"Number found";
return;
}
if(got>0) cout<<"Number found";
else cout<<"Number not found";
}
int linearSearch(int *arr,int index,int
size,int number)
{
for(int i=index;i>0;i--)
{
if(arr[i]==number)
{
return 1;
}
}
return 0;
}
```

---