**PROGRAM- 1**

**AIM:** Write a program for implementing the Linear Search.

1. Using Array

**Code:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

void main()

{

clrscr();

int element,i,cnt=0,s,arr[20];

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

cin>>s;

cout<<"Enter the elements:";

for(i=0;i<s;i++)

{

cin>>arr[i];

}

cout<<"Enter the element to search:";

cin>>element;

for(i=0;i<s;i++)

{

if(arr[i]==element)

{

cnt++;

}

}

cout<<"Given element is Found"<<" "<<cnt<<" times in the array";

if (cnt==0)

{

cout<<"Element is not found in the list";

}

getch(); }

**Output:**

****

1. Using Linked List

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<alloc.h>

struct node{

int value;

struct node \*next;

}; node \*head;

void insertion(){

int s;

node \*nd,\*pointer;

pointer=head;

printf("\n Enter the value to be inserted:");

scanf("%d",&s);

nd=(node\*)malloc(sizeof(struct node));

nd->value=s;

if(head==NULL)

{

nd->next=NULL;

head=nd;}

else{

while(pointer->next!=NULL)

{

pointer=pointer->next;

}

pointer->next=nd;

nd->next=NULL;

}

}

void traversal(){

struct node \*pointer;

pointer=head;

if(head==NULL)

{

printf("\n List is empty");

getch();

}

else{

while(pointer!=NULL) {

printf(" %d->\t",pointer->value);

pointer=pointer->next;

}

getch();

}

}

void search\_ele(){

int cnt=0, element,idx=0;

struct node\*pointer;

pointer=head;

printf("\n Enter the number to be searched:");

scanf("%d",&element);

while(pointer!=NULL||cnt==0){

if(pointer->value==element)

{

cnt=1;

break;

}

else{

idx++;

pointer=pointer->next;

}

}

if(cnt==1)

{

printf("\n %d is found at %d",element,idx+1);

}

else{

printf("\n no is not present");

getch();

}

}

void main(){

int c;

P: clrscr();

printf("1. Insertion\n");

printf("2. Traversal\n");

printf("3. Searching\n");

printf("4. Exit\n");

printf("Enter your choice");

scanf("%d",&c);

switch(c){

case 1: insertion();

goto P;

case 2: traversal();

goto P;

case 3: search\_ele();

goto P;

case 4: exit(0);

default:

printf("This Choice is not acceptable.");

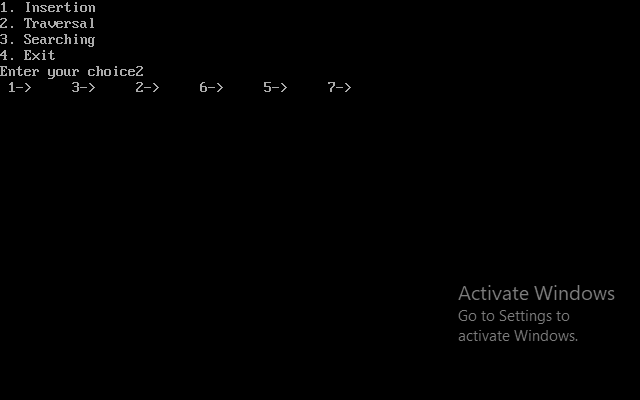
getch();

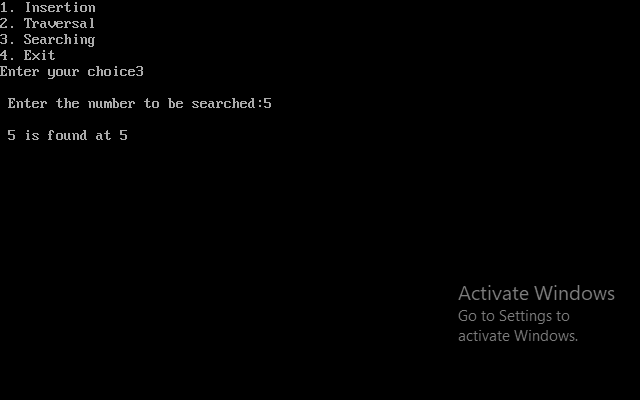
goto P;

}

}

**Output:**





**Complexity:**

Best Case : O(1)

Average Case : O(n)

Worst Case : O(n)