**Aim:** Program to implement Binary search.

(1). Using array

**Source code:**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

void main()

{

int mid,item,i,a[20],size,f=0,upper,lower=1;

clrscr();

cout<<"Enter size of array";

cin>>size;

upper=size;

cout<<"Enter elements of an array in ascending order:";

for(i=1;i<=size;i++)

{

cin>>a[i];

}

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

cin>>item;

while((f==0)&&(lower<upper))

{

mid=(lower+upper)/2;

if(item==a[mid])

{

cout<<"element is found at "<<mid;

f=1;

}

else

{

if(item<a[mid])

{

upper=mid-1;

}

else

{

lower=mid+1;

}

}

if (f==0)

{

cout<<"Element not found!";

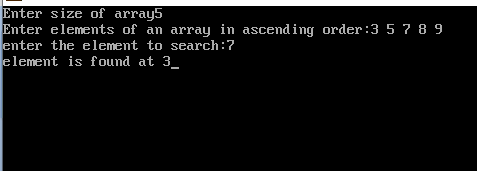
}

}

getch();

}

**Output:**



(2). Using linked list

**Source code:**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct node{

int data;

struct node \*next;

};

struct node \*head,\*n;

void insertion(){

int p;

struct node \*pntr;

pntr=head;

printf("Enter the value of the node (in ascending order:) ");

scanf("%d",&p);

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

n->data=p;

if(head==NULL){

n->next=NULL;

head=n;

}else{

while(pntr->next!=NULL){

pntr=pntr->next;

}

n->next=NULL;

pntr->next=n;

}

}

void traversal(){

struct node \*pntr;

pntr=head;

if(head==NULL){

printf("The given List is empty");

getch();

}

else{

while(pntr!=NULL){

printf("%d-> ",pntr->data);

pntr=pntr->next;

}

getch();

}

}

void search(){

struct node \*pntr;

pntr=head;

int j,lower=1,upper=0,s,cnt=0,key\_value;

if(head==NULL){

printf(" The given List is empty");

}

else{

printf("Enter the key value: ");

scanf("%d",&key\_value);

while(pntr!=NULL){

pntr=pntr->next;

upper++;

}

do{

s=0;

pntr=head;

j=(lower+upper)/2;

while(s!=j-1){

pntr=pntr->next;

s++;

}

if(pntr->data==key\_value){

printf("%d found at position %d",key\_value,j);

cnt=1;

}

else if(key\_value>pntr->data){

lower=j+1;

}

else{

upper=j-1;

}

}while(lower<=upper && cnt==0);

if(cnt==0){

printf(" The Value is not found in the list.");

}

}

getch();

}

void main(){

int ch;

M:system("cls");

printf("1. Insertion\n");

printf("2. Traversal\n");

printf("3. Search\n");

printf("4.Exit\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch(ch){

case 1:

insertion();

goto M;

case 2:

traversal();

goto M;

case 3:

search();

goto M;

case 4:

exit(0);

default:

printf("Invalid choice");

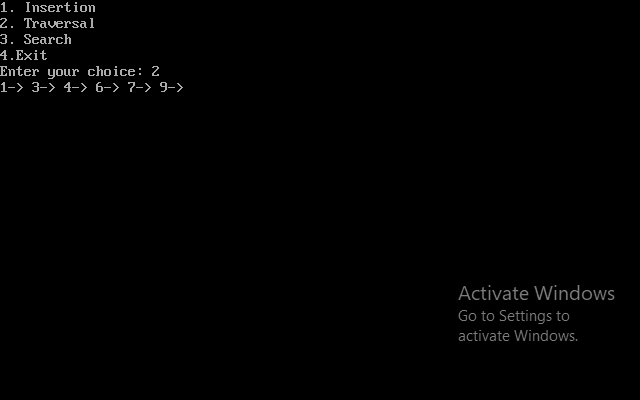
getch();

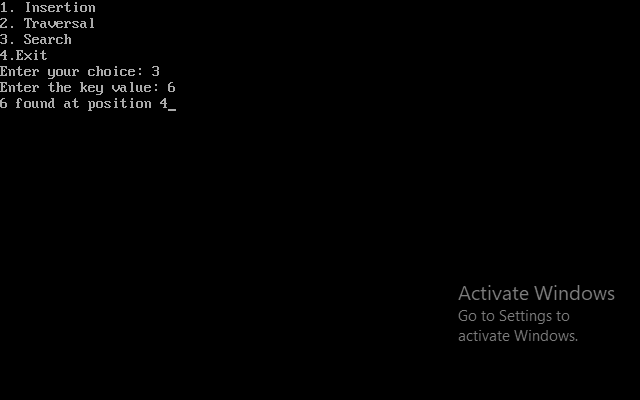
goto M;

}

}

**Output:**

****

****

**Analysis:**

1. Best case complexity = O(1)

2. Worst case complexity = O(log n)

3. Average case complexity = O(log n)