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

(1). Using array

**Code:**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void main()

{

int i,a[10],s,found=0,x;

clrscr();

printf("\nEnter size of an array from(1 to 10)");

scanf("%d",&s);

printf("\nEnter elements of an array");

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

scanf("%d",&a[i]);

printf("\n Enter a number to search");

scanf("%d",&x);

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

{

if(a[i]==x)

{

printf("\n%d is found at location %d",a[i], i+1);

found=1;

}

else

{

i=i+1;

}

}

if(found==0)

{

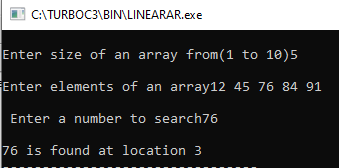
printf("\nSearch unsuccessful");

}

getch();

}

**Output:**



(2). Using Linked list

**Source code:**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<alloc.h>

struct NODE{

int data;

struct NODE \*nxt;

}; NODE \*head;

void insertion(){

int num;

NODE \*m,\*ptr;

ptr=head;

printf("\n Enter the number that you want to insert:");

scanf("%d",&num);

m=(NODE\*)malloc(sizeof(struct NODE));

m->data=num;

if(head==NULL)

{

m->nxt=NULL;

head=m;}

else{

while(ptr->nxt!=NULL)

{

ptr=ptr->nxt;

}

ptr->nxt=m;

m->nxt=NULL;

} }

void traverse\_lst(){

struct NODE \*ptr;

ptr=head;

if(head==NULL)

{

printf("\n The list is completely empty");

getch();

}

else{

while(ptr!=NULL) {

printf("%d->\t",ptr->data);

ptr=ptr->nxt;

} }

getch();

}

void search\_element(){

int ct=0,elmt,i=0;

struct NODE\*ptr;

ptr=head;

printf("\n enter the number you want to search");

scanf("%d",&elmt);

while(ptr!=NULL||ct==1){

if(ptr->data==elmt)

{

ct=1;

break;

}

else{

i++;

ptr=ptr->nxt;

}

}

if(ct==1)

{

printf("\n %d is found at %d",elmt,i+1);

}

else{

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

}

getch();

}

void main(){

int choice;

R: clrscr();

printf("1. Insertion\n");

printf("2. Traverse\n");

printf("3. Search\n");

printf("4. Exit\n");

printf("Enter your choice");

scanf("%d",&choice);

switch(choice){

case 1: insertion();

goto R;

case 2: traverse\_lst();

goto R;

case 3: search\_element();

goto R;

case 4: exit(0);

default:

printf("Incorrect choiceis made.");

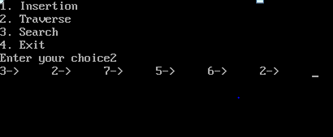
getch();

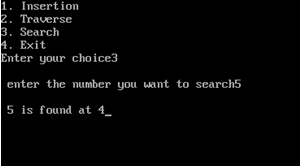
goto R;

}

}

**Output:**





**Complexity:**

Best Case : O(1)

Average Case: O(n)

Worst Case : O(n)