DESIGN AND ANALYSIS OF ALGORITHM

LAB EXAMINATION

DAA 14.351

PROGRAM: LINEAR SEARCH

AIM: TO FIND AN ELEMENT X IN THE ARRAY USING LINEAR SEARCH.

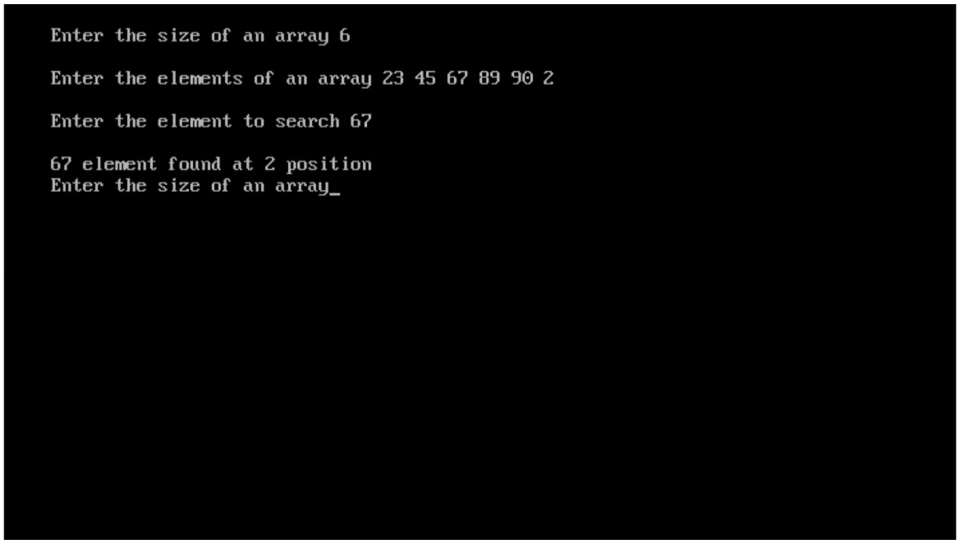
INPUT: ARRAY AND ELEMENT TO SEARCH.

OUTPUT: IF ELEMENT FOUND THE INDEX OF THE ELEMENT OTHERWISE NOT FOUND .

SOURCE CODE

```
//linear search
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
# define MAX 50
void main(){
 int i,x,t,n,found=0,a[MAX];
 int io;
 clrscr();
 printf("\nEnter the size of an array");
 io=scanf("%d",&n);
 if(io!=1||n<0){
  printf("\nInvalid input");
  getch();
  exit(0);
  }
 else{
  printf("\nEnter the elements of an array");
   for(i=0;i<n;i++){
    scanf("%d",&a[i]);
  }
 do{
 i=0;
 printf("\nEnter the element to search");
 scanf("%d",&x);
 while(i<=n){
  if(a[i]==x){
    found=1;
    t=i;
    break;
  }
  else{
   i=i+1;
  }
 }
 if(found==0){
  printf("\nSearch unsucessful");
  exit(0);
 }
 else{
  printf("\n%d element found at %d position",x,t);
  exit(0);
 }
 }
 while(1);
 }
 }
```

OUTPUT:



```
Enter the size of an array 6

Enter the elements of an array 23 45 67 89 90 2

Enter the element to search 67

67 element found at 2 position
Enter the size of an array_
```

AIM: TO SEARCH AN ELEMENT IN LINKED LIST

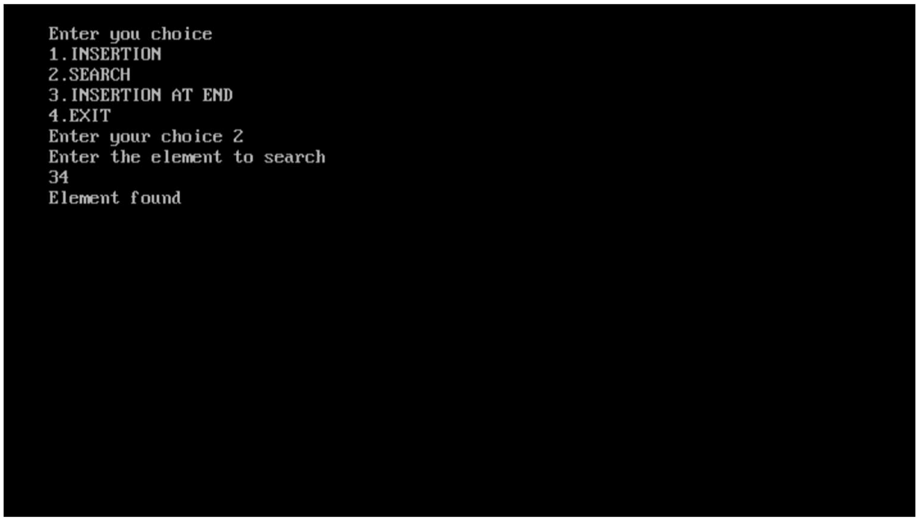
SOURCE CODE

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<alloc.h>

struct NODE
{
 int data;
 struct NODE *link;
};

NODE *Head;
void insertend()
{
 int x;
 NODE *N;
 NODE *ptr;
 printf("\nEnter the data x for the last node");
 scanf("%d",&x);
 if(Head==NULL)
 {
  N=(NODE *)malloc(sizeof(struct NODE));
  N->data=x;
  N->link=NULL;
  Head=N;
 }
 else
 {
  ptr=Head;
  while(ptr->link!=NULL)
  {
   ptr=ptr->link;
  }
  N=(NODE *)malloc(sizeof(struct NODE));
  N->data=x;
  N->link=NULL;
  ptr->link=N;
 }
}
void insertfront()
{
 int x;
 NODE *N;
 printf("\nEnter data x for new node");
 scanf("%d",&x);
 if(Head==NULL)
 {
  N=(NODE *)malloc(sizeof(struct NODE));
  N->data=x;
  N->link=NULL;
  Head=N;
  }
 else
 {
 N=(NODE *)malloc(sizeof(struct NODE));
 N->data=x;
 N->link=Head;
 Head=N;
 }
}
void search()
{
 NODE *ptr;
 ptr=Head;
 int x;
 printf("Enter the element to search \n");
 scanf("%d",&x);
 int c=0;
 while(ptr!=NULL)
 {
  if(x==ptr->data){
     printf("Element found\n");
     c=1;
  }
  ptr=ptr->link;
 }
 if(c==0){
  printf("Element not found\n");
 }
 getch();
}
void main()
{
 int c;
 L:clrscr();
 printf("\nEnter you choice");
 printf("\n1.INSERTION");
 printf("\n2.SEARCH");
 printf("\n3.INSERTION AT END");
 printf("\n4.EXIT");
 printf("\nEnter your choice");
 scanf("%d",&c);
 switch(c)
 {
 case 1:insertfront();
            goto L;
 case 2:search();
            goto L;
 case 3:insertend();
            goto L;
 case 4:exit(0);
 }
 getch();
}
```

OUTPUT



```
Enter you choice
1.INSERTION
2.SEARCH
3.INSERTION AT END
4.EXIT
Enter your choice 2
Enter the element to search
34
Element found
```