#include<stdio.h>

void main(){

int a=0,i=0,searchvalue=0,flag=0;

printf("Linear Search\n====================================\nEnter The Length Of Array: ");

scanf("%d",&a);

int b[a];

for(i=0;i<a;i++){

printf("Enter The %dth element: ",i+1);

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

}

printf("Array Entered : ");

for(i=0;i<a;i++){

printf("%d\t",b[i]);

}

printf("\nEnter The Search Value: ");

scanf("%d",&searchvalue);

for(i=0;i<a;i++){

if(b[i]==searchvalue){

flag=1;

printf("Element Found At %dth position\n",i+1);

}

}

if(flag==0){

printf("Element Not Found\n");

}

}

Output:

Linear Search

====================================

Enter The Length Of Array: 4

Enter The 1th element: 10

Enter The 2th element: 5

Enter The 3th element: 9

Enter The 4th element: 21

Array Entered : 10 5 9 21

Enter The Search Value: 21

Element Found At 4th position

#include<stdio.h>

#define Size 100

void main(){

int flag=0,FIRST,LAST,mid,Array[Size],maxSize,searchvalue,i;

printf("Binary Search : \nEnter the size of array : ");

scanf("%d",&maxSize);

printf("!!!!! Element Should be ordered !!!!!\n");

for(i=0;i<maxSize;i++){

printf("Enter Element : ");

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

}

FIRST=0;

LAST=maxSize-1;

printf("Enter the value to be searched : ");

scanf("%d",&searchvalue);

while (FIRST<=LAST){

mid=((FIRST+LAST)/2);

if (Array[mid]==searchvalue){

printf("Element Found At %d\n",mid+1);

flag=1;

break;

}

else{

if (Array[mid]<searchvalue){

FIRST=mid+1;

}

else{

LAST=mid-1;

}

}

}

if (flag==0){

printf("Element doesn't exsist in the array\n");

}

}

Output:

Binary Search :

Enter the size of array : 5

!!!!! Element Should be ordered !!!!!

Enter Element : 1

Enter Element : 3

Enter Element : 5

Enter Element : 7

Enter Element : 9

Enter the value to be searched : 5

Element Found At 3

Binary Search :

Enter the size of array : 5

!!!!! Element Should be ordered !!!!!

Enter Element : 1

Enter Element : 3

Enter Element : 5

Enter Element : 7

Enter Element : 9

Enter the value to be searched : 2

Element doesn't exsist in the array

#include<stdio.h>

#include<stdlib.h>

struct node{

int data;

struct node \*next;

};

struct node \*head =NULL;

void insertBegin(){

int item;

struct node \* new;

printf("Choice Made Is Insertion Through The Begining\nEnter Data To Be Added : ");

scanf("%d",&item);

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

new->data=item;

new->next=NULL;

if (head==NULL){

head=new;

}

else{

new->next=head;

head=new;

}

}

void insertEnd(){

int item;

struct node \* new;

struct node \* temp;

printf("Choice Made Is Insertion Through The End\nEnter Data To Be Added : ");

scanf("%d",&item);

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

new->data=item;

new->next=NULL;

if (head==NULL){

head=new;

}

else{

temp=head;

while(temp->next!=NULL){

temp=temp->next;

}

temp->next=new;

}

}

void insertBtw(){

int item,x;

struct node \* new;

struct node \* temp;

printf("Choice Made Is Insertion Between Elements\nEnter Data To Be Added : ");

scanf("%d",&item);

printf("After which element ? : ");

scanf("%d",&x);

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

new->data=item;

new->next=NULL;

if (head==NULL){

head=new;

}

else{

temp=head;

while(temp->data!=x){

temp=temp->next;

}

new->next=temp->next;

temp->next=new;

}

}

void deleteBegin(){

printf("Choice Made Is Deletion Through The Begining\n");

if (head==NULL){

printf("Singly Linked List Empty\n");

}

else{

head=head->next;

}

}

void deleteEnd(){

printf("Choice Made Is Deletion Through The End\n");

struct node \* temp;

struct node \* prev;

if (head==NULL){

printf("Singly Linked List Empty\n");

}

else{

temp=head;

while (temp->next!=NULL){

prev=temp;

temp=temp->next;

}

prev->next=NULL;

}

}

void deleteBtw(){

printf("Choice Made Is Deletion Between\n");

struct node \* temp;

struct node \* prev;

if (head==NULL){

printf("Singly Linked List Empty\n");

}

else if (head->next==NULL){

head=NULL;

}

else{

int x;

printf("Value To Be Deleted : ");

scanf("%d",&x);

temp=head;

while (temp->data!=x){

prev=temp;

temp=temp->next;

}

prev->next=temp->next;

}

}

void display(){

printf("Choice Made Is Display\n");

struct node \* temp;

if (head==NULL){

printf("Singly Linked List Empty\n");

}

else{

printf("Elements Present Are :\t");

temp=head;

while (temp->next!=NULL){

int data=(temp->data);

printf("%d\t",data);

temp=temp->next;

}

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

}

}

void main(){

int choice=0;

do{

printf("====================\n Singly Linked List : \n ==================== \n 1.Insert Beginning \n 2.Insert End \n 3.Insert After Particular Element\n 4.Delete Beginning \n 5.Delete End \n 6.Delete Particular Element \n 7.Display \n 8.Exit \n\t Enter Choice : ");

scanf("%d",&choice);

switch (choice){

case 1:

insertBegin();

break;

case 2:

insertEnd();

break;

case 3:

insertBtw();

break;

case 4:

deleteBegin();

break;

case 5:

deleteEnd();

break;

case 6:

deleteBtw();

break;

case 7:

display();

break;

case 8:

exit(0);

default:

printf("!!Invaild Choice Retry!!\n");

break;

}

}while(1);

}