1.Binary search (Iterative)

#include<stdio.h>

#include<stdlib.h>

void binary\_search(int \*, int,int);

void bubbleSort(int \*a, int n) ;

int main() {

int \*a,n,i,x;

printf("\nEnter the total number of elements in the array : ");

scanf("%d",&n);

a=(int\*)malloc(n\*sizeof(int));

printf("\nEnter elements of the array : \n");

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

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

}

bubbleSort(a,n);

printf("\nEnter element to be searched : ");

scanf("%d",&x);

binary\_search(a,n,x);

return 0;

}

void bubbleSort(int \*a, int n) {

int i,j,temp;

for ( i= 0; i < n - 1; i++) {

for (j = 0; j < n-i- 1;j++) {

if (a[j] > a[j + 1]) {

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

}

void binary\_search(int \*a,int n, int x) {

int low=0,high=n-1,mid;

while(low<=high) {

mid=(low+high)/2;

if( x > a[mid] )

low=mid+1;

if( x < a[mid] )

high=mid-1;

if( x == a[mid] ) {

printf("\nElement Present in the array ");

break;

}

}

if(low>high) {

printf("\nElement not found !!");

}

}

2.Binary search (Recursive)

#include <stdio.h>

#include<stdlib.h>

int binarySearch(int arr[], int low, int high, int x) {

if (low <= high) {

int mid = (low + high)/2;

if (arr[mid] == x) {

printf("Element is present in the array ");

return mid;

}

if (arr[mid] > x)

return binarySearch(arr, low, mid-1, x);

return binarySearch(arr, mid+1, high, x);

}

printf("Element is not present in array");

return -1;

}

void bubbleSort(int \*a, int n) {

int i,j,temp;

for ( i= 0; i < n - 1; i++) {

for (j = 0; j < n-i- 1;j++) {

if (a[j] > a[j + 1]) {

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

}

}

int main() {

int \*a,n,i,x;

printf("\nEnter the total number of elements in the array : ");

scanf("%d",&n);

a=(int\*)malloc(n\*sizeof(int));

printf("\nEnter elements of the array : \n");

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

{

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

}

bubbleSort(a,n);

printf("\nEnter element to be searched : ");

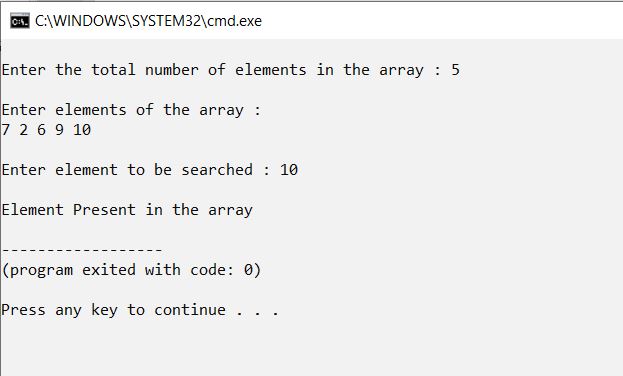
scanf("%d",&x);

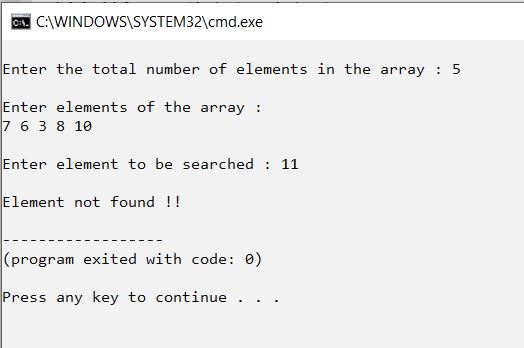
binarySearch(a, 0, n-1, x);

return 0;

}

(Iterative)





(Recursive)

