## Write a C program to implement the Binary Search and Linear Search algorithm with array and linked list.

**Binary Search Through Array**

#include <stdio.h> int main()

{

int i, low, high, mid, n, key, array[100]; scanf("%d",&n);

for(i = 0; i < n; i++) scanf("%d",&array[i]); scanf("%d", &key);

low = 0; high = n - 1;

mid = (low+high)/2; while (low <= high) { if(array[mid] < key) low = mid + 1;

else if (array[mid] == key) {

printf("%d found at index %d", key, mid+1); break;

}

else

high = mid - 1;

mid = (low + high)/2;

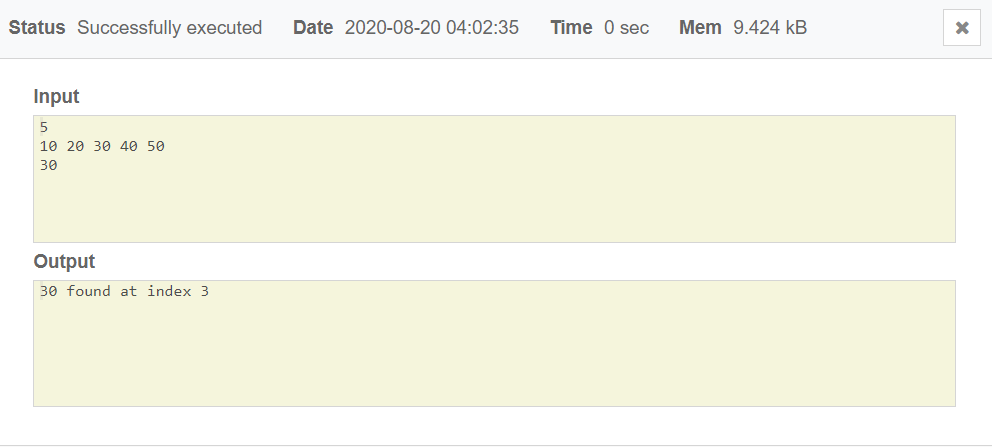
}

if(low > high)

printf("Not found! %d isn't present in the list", key); return 0;

}

**Output:**



## **Linear Search Through Array**

#include<stdio.h> int main()

{

int a[20],i,x,n;

scanf("%d",&n);

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

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

scanf("%d",&x);

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

if(a[i]==x) break;

if(i<n)

printf("Element found at index %d \n",i);

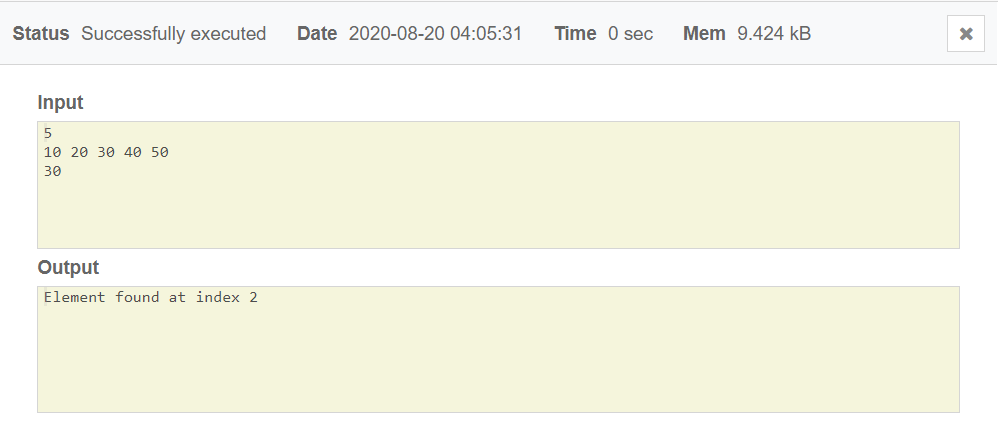
else

printf("Element not found \n");

return 0;

}

**Output:**



## **Linear Search Through Linked List**

#include<stdio.h> #include<malloc.h> struct node

{

int data;

struct node \*next;

}

first, \*nw;

int search(int item)

{

int count=1; nw=&first;

while(nw->next!=NULL)

{

if(nw->data==item) break;

else

count++; nw=nw->next;

}

return count;

}

int main()

{

int no,i,item,pos; first.next=NULL; nw=&first;

scanf("%d",&no);

printf("\n"); for(i=0;i< no;i++)

{

nw->next=(struct node \*)malloc(sizeof(struct node)); scanf("%d",&nw->data);

nw=nw->next;

}

nw->next=NULL;

printf("Elements in linked list ");

nw=&first;

while(nw->next!=NULL)

{

printf("%d ",nw->data); nw=nw->next;

}

printf("\n");

scanf("%d",&item);

pos=search(item); if(pos<=no)

printf("%d is found at node %d",item,pos); else

printf("Sorry! '%d' is not in linked list.",item); return 0;

}

**Output:**

****

**Binary Search Through Linked List**

#include<stdio.h> #include<stdlib.h>

struct Node{ int data;

struct Node\* next;

};

Node \*newNode(int x){

struct Node\* temp = new Node; temp->data = x;

temp->next = NULL; return temp;

}

struct Node\* mid\_node(Node\* start, Node\* last){ if (start == NULL)

return NULL;

struct Node\* slow = start;

struct Node\* fast = start -> next; while (fast != last){

fast = fast -> next; if (fast != last){

slow = slow -> next; fast = fast -> next;

}

}

return slow;

}

struct Node\* binarySearch(Node \*head, int value){ struct Node\* start = head;

struct Node\* last = NULL; do{

Node\* mid = mid\_node(start, last);

if (mid == NULL) return NULL;

if (mid -> data == value) return mid;

else if (mid -> data < value) start = mid -> next;

else

last = mid;

}

while (last == NULL || last != start); return NULL;

}

int main(){

int value;

Node \*head = newNode(5); head->next = newNode(12);

head->next->next = newNode(18);

head->next->next->next = newNode(23);

head->next->next->next->next = newNode(52);

head->next->next->next->next->next = newNode(76); printf("Enter a value: ");

scanf("%d",&value);

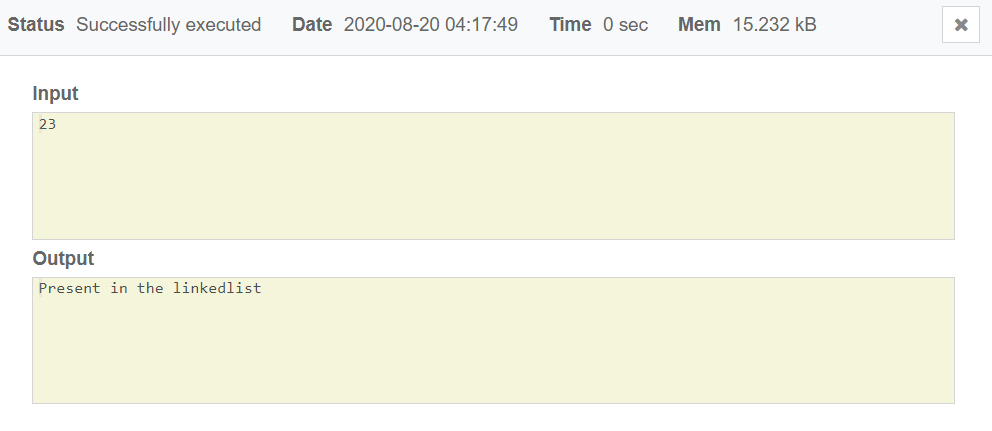
if (binarySearch(head, value) == NULL) printf("Value is not present in linked list\n");

else

printf("The value is present in linked list\n"); return 0;

}

Output:



## Write a program to sort values in Ascending / Increasing order using Bubble Sort technique in linear array using recursion and without recursion.

**Bubble Sort With Recursion**

#include <stdio.h> #include <stdlib.h>

void bubbleSort(int \*data, int n) { int i, temp;

if (n > 0) {

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

if (data[i - 1] > data[i]) { temp = data[i]; data[i] = data[i - 1]; data[i - 1] = temp;

}

}

bubbleSort(data, n - 1);

}

return;

}

int main() { int i, n, \*data;

printf("Enter the number of inputs: "); scanf("%d", &n);

data = (int \*) malloc(sizeof(int) \* n); for (i = 0; i < n; i++) {

printf("data[%d]: ", i);

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

}

bubbleSort(data, n); printf("Sorted array "); for (i = 0; i < n; i++) {

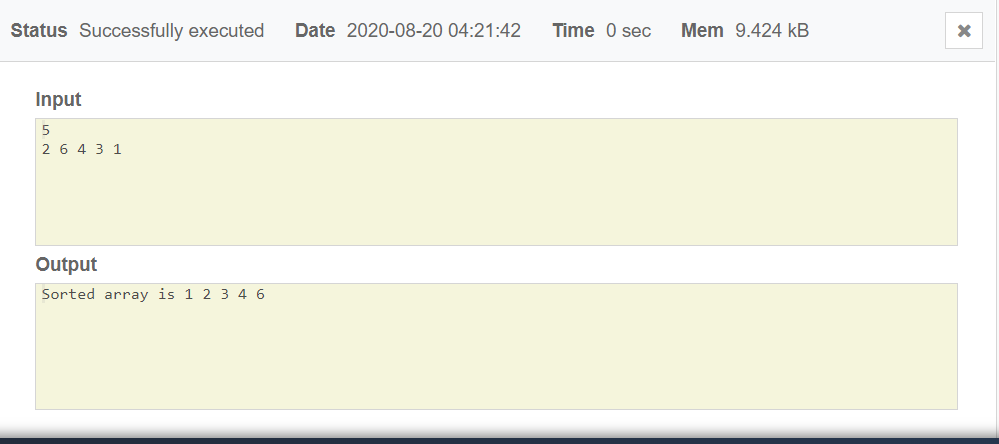
printf("%d ", data[i]);

}

printf("\n"); return 0;

}

Output:



## **Bubble Sort Without Recursion In Ascending And Descending Order**

#include <stdio.h> #include<conio.h> int main()

{

int a[100],n,i,j; printf("Array size: "); scanf("%d",&n); printf("Elements: "); for(i=0;i<n;i++)

{

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

}

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

{

for (int j = 0; j < n; j++)

{

if (a[j] > a[i])

{

int tmp = a[i]; a[i] = a[j]; a[j] = tmp;

}

}

}

printf("Ascending ");

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

{

printf("%d ", a[i]);

}

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

{

for (int j = 0; j < n; j++)

{

if (a[j] < a[i])

{

int tmp = a[i]; a[i] = a[j]; a[j] = tmp;

}

}

}

printf("\nDescending ");

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

{

printf("%d ", a[i]);

}

return 0;

}

**Output:**

