

**Data Structure and Algorithm (MCA 271)**

**Lab Practical –**

***BY***

**Himanshu Heda (24225013)**

**SUBMITTED TO**

**Prof. Vandna Kansal**

**SCHOOL OF SCIENCES**

**2024-2025**

**Program Description:**

**Code of the program**

**Output**: - Paste the o/p of the program.

1. Linear Search : --

#include <stdio.h>

int linear\_search(int arr[], int size, int target) {

    for (int i = 0; i < size; i++) {

        if (arr[i] == target) {

            return i; // Return the index of the target

        }

    }

    return -1; // Return -1 if the target is not found

}

int main() {

    int arr[] = {5, 3, 8, 4, 2};

    int size = sizeof(arr) / sizeof(arr[0]);

    int target = 4;

    int result = linear\_search(arr, size, target);

    if (result != -1) {

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

    } else {

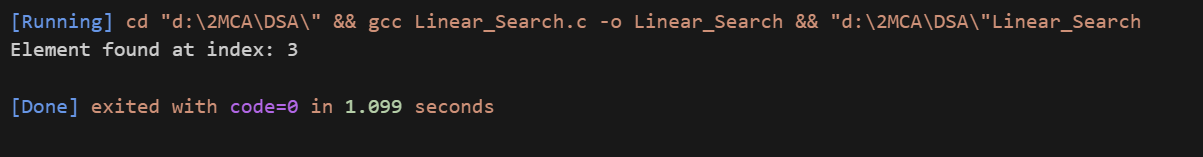
        printf("Element not found\n");

    }

    return 0;

}

Output : --



1. Binary Search : --

#include<stdio.h>

// An iterative binary search function

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

    while (low <= high)

    {

        int mid = low + (high-low)/2;

        // Check if x is present at mid

        if(arr[mid] == x)

            return mid;

        // If x greater, ignore left half

        if(arr[mid] < x)

            low = mid + 1;

        // If x smaller, ignore right half

        else

            high = mid - 1;

    }

    // If we reach here, then element was not present

    return -1;

}

int main(void){

    int arr[] = {2,3,4,10,40};

    int n = sizeof(arr)/sizeof(arr[0]);

    int x = 10;

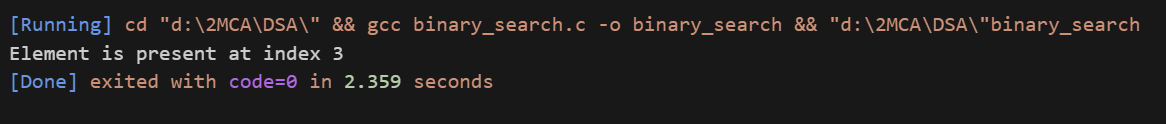
    int result = binarySearch(arr,0,n - 1,x);

    if(result == -1) printf("Element is not present in array");

    else printf("Element is present at index %d",result);

}

Output : --

****