**Lab Task -3**

1. Implement linear Search:

#include <stdio.h>

int main() {

int n, x;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d elements of the array:\n", n);

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

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

}

printf("Enter the element to search: ");

scanf("%d", &x);

int found = 0;

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

if (arr[i] == x) {

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

found = 1;

break;

}

}

if (!found) {

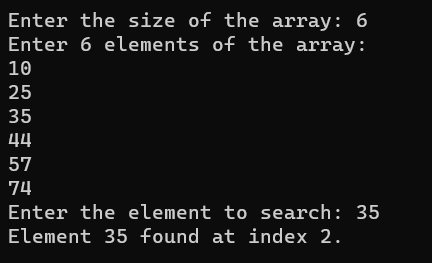
printf("Element %d not found in array.\n", x);

}

return 0;

}

**Out put:**



1. Implement binary search:

#include <stdio.h>

int binary\_search(int arr[], int n, int target) {

int left = 0;

int right = n - 1;

while (left <= right) {

int mid = (left + right) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

int main() {

int n;

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d sorted elements: ", n);

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

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

}

int target;

printf("Enter the target element to search for: ");

scanf("%d", &target);

int index = binary\_search(arr, n, target);

if (index != -1) {

printf("Target element %d found at index %d.\n", target, index);

} else {

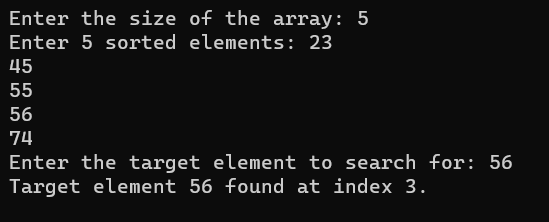
printf("Target element %d not found in the array.\n", target);

}

return 0;

}

**Out put:**

****