**1. Linear Search in C**

**Time Complexity**: O(n)

**Code:**

#include <stdio.h>

int linear Search(int arr[], int size, int key) {

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

if(arr[i] == key)

return i; // Return index

}

return -1;

}

int main () {

int arr[100], n, key;

printf("Enter number of elements: ");

scanf("%d", &n);

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

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

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

printf("Enter element to search: ");

scanf("%d", &key);

int result = linear Search(arr, n, key);

if (result != -1)

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

else

printf("Element not found\n");

return 0;

}

**Sample Input/Output:**

Enter number of elements: 5

Enter 5 elements:

10 30 50 20 40

Enter element to search: 20

Element found at index 3

**2. Binary Search in C**  
**Time Complexity**: O(log n)

**Code:**

#include <stdio.h>

int binary Search(int arr[], int size, int key) {

int low = 0, high = size - 1;

while (low <= high) {

int mid = (low + high) / 2;

if(arr[mid] == key)

return mid;

else if (key < arr[mid])

high = mid - 1;

else

low = mid + 1;

}

return -1;

}

int main () {

int arr[100], n, key;

printf("Enter number of elements: ");

scanf("%d", &n);

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

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

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

printf("Enter element to search: ");

scanf("%d", &key);

int result = binary Search(arr, n, key);

if (result != -1)

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

else

printf("Element not found\n");

return 0;

}

Sample Input/Output:

Enter number of elements: 5

Enter 5 sorted elements:

10 20 30 40 50

Enter element to search: 30

Element found at index 2