





Code (Linear Search)

#include <iostream>

#include <bits/stdc++.h>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main()

{

    int key, n, flag = 0;

    cout << "Enter Number of elements: ";

    cin >> n;

    int arr[n];

    cout << "Enter the Elements: ";

    for (int i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

    cout << "Enter Key: ";

    cin >> key;

    auto start = steady\_clock::now();

    for (int i = 0; i < n; i++)

    {

        if (key == arr[i])

        {

            cout << "Key found at index: " << i << endl;

            flag = 1;

            break;

        }

    }

    if (flag == 0)

    {

        cout << "key not found" << endl;

    }

    auto stop = steady\_clock::now();

    auto duration = duration\_cast<nanoseconds>(stop - start);

    cout << "\nTime taken by function: " << duration.count() << " nanoseconds" << endl;

}

Output (Linear Search)



Code (Binary Search)

#include <bits/stdc++.h>

#include <chrono>

using namespace std;

using namespace std::chrono;

int binary\_search(int arr[], int first, int last, int key)

{

    if (last >= first)

    {

        int mid = first + (last - first) / 2;

        if (arr[mid] == key)

            return mid;

        if (arr[mid] > key)

            return binary\_search(arr, first, mid - 1, key);

        return binary\_search(arr, mid + 1, last, key);

    }

    return -1;

}

int main()

{

    int n, mid, last, first = 0, key, flag = 0, result;

    cout << "Enter Number of Elements: ";

    cin >> last;

    int arr[last];

    cout << "Enter elements in Sorted order: ";

    for (int i = 0; i < last; i++)

    {

        cin >> arr[i];

    }

    cout << "Enter Key: ";

    cin >> key;

    auto start = steady\_clock::now();

    result = binary\_search(arr, first, last, key);

    if (result != -1)

    {

        cout << "Key Found at index: " << result << endl;

    }

    else

    {

        cout << "Key not Found" << endl;

    }

    auto stop = steady\_clock::now();

    auto duration = duration\_cast<nanoseconds>(stop - start);

    cout << "\nTime taken by function: " << duration.count() << " nanoseconds" << endl;

}

Output (Binary Search)

