1. WAP to perform linear search and Binary search

Linear Search

#include < iostream >

using namespace std;

void linearSearch(int a[], int n) {

int temp = -1;

for (int i = 0; i < 5; i++) {

if (a[i] == n) {

cout << "Element found at position: " << i + 1 << endl;

temp = 0;

break;

}

}

if (temp == -1) {

cout << "No Element Found" << endl;

}

}

int main() {

int arr[5];

cout << "Please enter 5 elements of the Array" << endl;

for (int i = 0; i < 5; i++) {

cin >> arr[i];

}

cout << "Please enter an element to search" << endl;

int num;

cin >> num;

linearSearch(arr, num);

return 0;

}

Binary Search

#include <iostream>

#include <conio.h>

**using** **namespace** std;

**int** main ()

{

    // declaration of the variables and array

**int** arr[100], st, mid, end, i, num, tgt;

    cout << " Define the size of the array: " << endl;

    cin >> num; // get size

    // enter only sorted array

        cout << " Enter the values in sorted array either ascending or descending order: " << endl;

    // use for loop to iterate values

**for** (i = 0; i < num; i++)

    {

        cout << " arr [" << i << "] = ";

        cin >> arr[i];

    }

    // initialize the starting and ending variable's values

    st = 0;

    end = num - 1; // size of array (num) - 1

    // define the item or value to be search

    cout << " Define a value to be searched from sorted array: " << endl;

    cin >> tgt;

    // use while loop to check 'st', should be less than equal to 'end'.

**while** ( st <= end)

    {

        // get middle value by splitting into half

        mid = ( st + end ) / 2;

        /\* if we get the target value at mid index, print the position and exit from the program. \*/

**if** (arr[mid] == tgt)          {

            cout << " Element is found at index " << (mid + 1);

            exit (0); // use for exit program the program

        }

        // check the value of target element is greater than the mid element' value

**else** **if** ( tgt > arr[mid])

        {

            st = mid + 1; // set the new value for st variable

        }

        // check the value of target element is less than the mid element' value

**else** **if** ( tgt < arr[mid])

        {

            end = mid - 1; // set the new value for end variable

        }

    }

    cout << " Number is not found. " << endl;

**return** 0;  }