**Assignment :- 01**

**Name:-** Abhishek Gaikwad **TY A1**

**Roll no:-** 371017  **Prn**:-22210975

**Problem Statement**: Write a program to perform binary search on an unsorted random list of at least 5000 elements. The key element should be user input. Use the Divide & Conquer method to implement this program.

**Code:-**

#include <iostream>

#include <cstdlib>  // For rand() and srand()

#include <ctime>    // For time()

#include <vector>   // For std::vector

using namespace std;

int main()

{

    // Seed the random number generator

    srand(time(0));

    // Declare variables

    int num, first, last, middle;

    int size, temp;

    // Ask user for the size of the array

    cout << "Enter the size of the array: ";

    cin >> size;

    // Create an array using a vector

    vector<int> arr(size);

    // Populate the array with random numbers between 1 and 10000

    for (int i = 0; i < size; i++)

        arr[i] = 1 + rand() % 10000; // Generate number between 1 to 10000

    // Display the generated array

    cout << "The generated array: ";

    for (int i = 0; i < size; i++)

        cout << arr[i] << " ";

    cout << endl;

    // Sort the array in ascending order using Bubble Sort

    for (int x = 0; x < size - 1; x++) // x < size - 1 to ensure bounds

    {

        for (int y = 0; y < size - x - 1; y++) // Corrected loop

        {

            if (arr[y] > arr[y + 1]) // Compare adjacent elements

            {

                // Swap elements if they are in the wrong order

                temp = arr[y];

                arr[y] = arr[y + 1];

                arr[y + 1] = temp;

            }

        }

    }

    // Output the sorted array

    cout << "\nElements sorted in ascending order: ";

    for (int x = 0; x < size; x++)

    {

        cout << arr[x] << " ";

    }

    cout << endl;

    // Ask user for the element to search

    cout << "\nEnter Element to be Searched: ";

    cin >> num;

    // Initialize search boundaries

    first = 0;

    last = size - 1;

    // Perform binary search

    while (first <= last)

    {

        middle = first + (last - first) / 2; // Avoids potential overflow

        if (arr[middle] < num) // If middle element is less than the target, search the right half

            first = middle + 1;

        else if (arr[middle] == num) // If the element is found

        {

            cout << "\nThe number, " << num << " found at position " << middle;

            break;

        }

        else // If middle element is greater than the target, search the left half

            last = middle - 1;

    }

    // If the element is not found in the array

    if (first > last)

        cout << "\nThe number, " << num << " is not found in the given array";

    cout << endl;

    return 0;

}

Output:-

