Domain Winter Camp DAY-5

Name = Diksha Sharma

Section/Group:-901-Kpit-B

UID = 22BCS12218

Problem 1

```cpp
#include <iostream>
#include <vector>
using namespace std;

// Function to find the first occurrence of k in the array
int findFirstOccurrence(int k, const vector<int>& arr) {
    for (size_t i = 0; i < arr.size(); ++i) {
        if (arr[i] == k) {
            return i + 1; // 1-based indexing
        }
    }
    return -1; // Element not found
}

int main() {
    int k;
    vector<int> arr;
    int n;

    // Input size of array
    cout << "Enter the number of elements in the array: ";
    cin >> n;

    arr.resize(n);
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }
```

input

```
Enter the number of elements in the array: 5
Enter the elements of the array: 1 2 3 4 5
Enter the value to search for: 3
Output: 3
```

Problem 2

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

// Function to search for k in a sorted array using binary search
bool binarySearch(const vector<int>& arr, int k) {
    int left = 0, right = arr.size() - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (arr[mid] == k) {
            return true; // Element found
        } else if (arr[mid] < k) {
            left = mid + 1; // Search in the right half
        } else {
            right = mid - 1; // Search in the left half
        }
    }

    return false; // Element not found
}

int main() {
    int n, k;
    vector<int> arr;
```

input

```
Enter the number of elements in the array: 3
Enter the elements of the array (sorted in ascending order): 1 4 7
Enter the value to search for: 7
Output: true
```

Problem 3

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

// Function to find target indices after sorting
vector<int> targetIndices(vector<int>& nums, int target) {
    vector<int> result;

    // Step 1: Sort the array
    sort(nums.begin(), nums.end());

    // Step 2: Find the target indices
    for (int i = 0; i < nums.size(); ++i) {
        if (nums[i] == target) {
            result.push_back(i);
        }
    }

    return result;
}

int main() {
    int n, target;
    vector<int> nums;

    // Input size of array
    cout << "Enter the number of elements in the array: ";
```

input

```
Enter the number of elements in the array: 5
Enter the elements of the array: 1 2 3 4 5
Enter the target value: 4
Output: [3]
```

Problem 4

```cpp
#include <iostream>
#include <vector>
using namespace std;

// Function to find the insert position or the index of the target
int searchInsert(vector<int>& nums, int target) {
    int left = 0, right = nums.size() - 1;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (nums[mid] == target) {
            return mid; // Target found
        } else if (nums[mid] < target) {
            left = mid + 1; // Search in the right half
        } else {
            right = mid - 1; // Search in the left half
        }
    }

    // If target is not found, left is the insert position
    return left;
}

int main() {
    int n, target;
    vector<int> nums;
```

input

```
Enter the number of elements in the array: 6
Enter the elements of the array (sorted in ascending order): 1 2 3 4 5 6
Enter the target value: 5
Output: 4
```

Problem 5

```cpp
    // Assign rank based on arr2's order
    for (int i = 0; i < arr2.size(); ++i) {
        rank[arr2[i]] = i;
    }

    // Custom comparator for sorting
    auto comparator = [&rank](int a, int b) {
        if (rank.count(a) && rank.count(b)) {
            return rank[a] < rank[b]; // Both in arr2, sort by rank
        } else if (rank.count(a)) {
            return true; // a is in arr2 but b is not
        } else if (rank.count(b)) {
            return false; // b is in arr2 but a is not
        } else {
            return a < b; // Neither in arr2, sort ascending
        }
    };

    sort(arr1.begin(), arr1.end(), comparator);

    return arr1;
}

int main() {
    int n1, n2;
    vector<int> arr1, arr2;

    // Input size of arr1
    cout << "Enter the number of elements in arr1: ";
```

input

```
Enter the number of elements in arr1: 5
Enter the elements of arr1: 1 2 3 4 5
Enter the number of elements in arr2: 5
Enter the elements of arr2: 9 8 7 6 5
Output: [5, 1, 2, 3, 4]
```