

Course Name- AP-I

Course Code- 21CSP-314

### **Experiment-1.4**

**Aim-** Sorting and Searching: Implement the concept of Searching and Sorting techniques

**Objectives-** The objectives of this program is to understand the concept of different types of searching and sorting.

Problem1: <https://www.hackerrank.com/challenges/fraudulent-activity-notifications/problem?isFullScreen=true>

Problem2: <https://www.hackerrank.com/challenges/missing-numbers/problem?isFullScreen=true>

### **Description-**

Searching is the process of fetching a specific element in a collection of elements. The collection can be an array or a linked list. If you find the element in the list, the process is considered successful, and it returns the location of that element.

There are two types of searching:

1. Linear Search
2. Binary Search

A Sorting Algorithm is used to rearrange a given array or list of elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of elements in the respective data structure.

There are different types of sorting techniques:

1. Bubble sort
2. Quick Sort
3. Merge Sort
4. Selection Sort
5. Insertion Sort
6. Counting Sort
7. Bucket Sort
8. Radix Sort
9. Heap sort

Course Name- AP-I

Course Code- 21CSP-314

**Problem 1:** Given the number of trailing days  $d$  and a client's total daily expenditures for a period of  $n$  days, determine the number of times the client will receive a notification over all  $n$  days.

**Problem 2:** Given two arrays of integers, find which elements in the second array are missing from the first array.

**Code:**

1.

```
Change Theme Language C++14
1  #include <iostream>
2  #include <vector>
3  #include <map>
4  #include <set>
5  #include <algorithm>
6
7  using namespace std;
8
9  #define MAXE 210
10
11  int A[200010];
12  int F[MAXE];
13
14  int median2(int D) {
15      int p = 0;
16      for (int i = 0; i < MAXE; i++) {
17          p += F[i];
18          if (p * 2 > D) {
19              return 2 * i;
20          } else if (p * 2 == D) {
21              for (int j = i + 1; ; j++) {
22                  if (F[j]) {
23                      return i + j;
24                  }
25              }
26          }
27      }
28      return -1;
29  }
30
```

Course Name- AP-I

Course Code- 21CSP-314

```
30
31 int main() {
32     int N, D;
33     cin >> N >> D;
34     for (int i = 0; i < N; i++) {
35         cin >> A[i];
36     }
37     int result = 0;
38     for (int i = 0; i < N; i++) {
39         if (i >= D) {
40             if (A[i] >= median2(D)) {
41                 ++result;
42             }
43             F[A[i] - D]--;
44         }
45         F[A[i]]++;
46     }
47     cout << result << endl;
48     return 0;
49 }
50
```

Line: 50 Col: 1

2.

```
Change Theme Language C++14
1  #include <bits/stdc++.h>
2  using namespace std;
3
4
5  int main() {
6
7      long long n,m,temp;
8      cin>>n;
9      vector<int> a;
10     for(long long i=0;i<n;i++) {
11         cin >> temp;
12         a.push_back(temp);
13     }
14     cin>>m;
15     vector<int> b;
16     for(long long i=0;i<m;i++){
17         cin >> temp;
18         b.push_back(temp);
19     }
20     sort(a.begin(),a.end());
21     sort(b.begin(),b.end());
22     long long i=0,j=0;
23     while(i<n && j<m){
24         if(a[i]==b[j]) {
25             i++;j++;
26             b[j-1]=0;
27         }
28         else if(a[i]>b[j])j++;
29         else i++;
30     }

```

Course Name- AP-I

Course Code- 21CSP-314

```
30. }  
31.     set<int> st;  
32.     for(i=0;i<m;i++) {  
33.         if(b[i]!=0) st.insert(b[i]);  
34.     }  
35.     for(set<int>::iterator it = st.begin();it!=st.end();it++){  
36.         cout<<*it<<" ";  
37.     }  
38.     cout<<endl;  
39.     return 0;  
40. }  
41. }
```

Line: 41 Co.

## Outcome-

### Problem 1 outcome

## Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

#### ✓ Sample Test case 0

#### ✓ Sample Test case 1

#### ✓ Sample Test case 2

Input (stdin)

1	9 5
2	2 3 4 2 3 6 8 4 5

Your Output (stdout)

1	2
---	---

Expected Output

1	2
---	---

Course Name- AP-I

Course Code- 21CSP-314

## Problem 2 Outcome

### Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

#### ✓ Sample Test case 0

#### ✓ Sample Test case 1

Input (stdin)

1	10
2	203 204 205 206 207 208 203 204 205 206
3	13
4	203 204 204 205 206 207 205 208 203 206 205 206 204

Your Output (stdout)

1	204 205 206
---	-------------

Expected Output

1	204 205 206
---	-------------

## Learning Outcomes-

1. Learnt different searching techniques.
2. Learnt different types of sorting techniques.
3. Learnt to find the element from an array using linear search.