Goal

- Today we will be looking at the basic data structures in C++.
- This is to test your proficiency with the following data structures:
    1. Vectors
    2. Sets
    3. Maps

Resources

Vectors

- https://cplusplus.com/reference/vector/vector/
    - read up on what the following functions do:
        - begin, end
        - rbegin, rend
        - size, empty
        - push_back, pop_back
        - erase, clear
- read up on how to use the following functions with vectors:
    - https://cplusplus.com/reference/algorithm/sort/
        - read the examples to see how to define a custom sorting function

Sets

- https://cplusplus.com/reference/set/set/
    - read up on what the following functions do:
        - begin, end
        - rbegin, rend
        - size, empty
        - insert
        - erase, clear
        - count

Maps

- https://cplusplus.com/reference/map/map/
    - read up on what the following functions do:
        - begin, end
        - rbegin, rend
        - size, empty
        - insert
        - erase, clear
        - count

Questions

Vectors

1. Given an integer n and n pairs of integers.
    - Store the pairs in a vector of pairs.
    - Sort the vector in ascending order using the `sort` function.
    - Print out the pairs in order.
    - Sort the vector in descending order using the `sort` function along with the `rbegin` and `rend` iterators.
    - Print out the pairs in order.
    - Use the `sort` function along with a custom comparator to sort the vector in descending order based on the first value of the pairs.
        - If the first value of two pairs are equal, they should be arranged in ascending order based on the second value.
    - Print out the pairs in order.
    - sample_input

```
5
1 7
3 5
2 6
1 6
2 7
```

- sample_output

```
1 6
1 7
2 6
2 7
3 5

3 5
2 7
2 6
1 7
1 6

3 5
2 6
2 7
1 6
1 7
```

- Solution

```
#include <bits/stdc++.h>
using namespace std;
```

```
bool cmp (pair<int, int> a, pair<int, int> b) {
    if (a.first == b.first)
        return a.second < b.second;
    return a.first > b.first;
}

int main() {
    int n; cin >> n;
    vector<pair<int, int>> v(n);
    for (int i = 0; i < n; i++)
        cin >> v[i].first >> v[i].second;

    sort(v.begin(), v.end());
    for (int i = 0; i < n; i++)
        cout << v[i].first << " " << v[i].second << "\n";

    sort(v.rbegin(), v.rend());
    for (int i = 0; i < n; i++)
        cout << v[i].first << " " << v[i].second << "\n";

    sort(v.begin(), v.end(), cmp);
    for (int i = 0; i < n; i++)
        cout << v[i].first << " " << v[i].second << "\n";
    return 0;
}
```

2. https://cses.fi/problemset/task/1084 (Medium)

- Solution

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, m, k;
    cin >> n >> m >> k;

    vector<int> a(n);
    for (int i = 0; i < n; i++)
        cin >> a[i];
    sort(a.begin(), a.end());

    vector<int> b(m);
    for (int i = 0; i < m; i++)
        cin >> b[i];
    sort(b.begin(), b.end());

    int next_person = 0, next_apt = 0;
    int ctr = 0;
    while (next_person != n && next_apt != m) {
        if (a[next_person] > b[next_apt] + k)
            next_apt++;
        else if (a[next_person] < b[next_apt] - k)
            next_person++;
        else
```

```
            next_person++, next_apt++, ctr++;
    }
    cout << ctr;
    return 0;
}
```

Sets

1. Given integers n and q.
   - You will be given a list of n integers, store them in a set.
   - Print the number of distinct elements in the set.
   - Using a range based `for` loop print the distinct elements in ascending order.
   - Now, using a reverse iterator and a `for` loop, print the distinct elements in descending order.
   - The next q lines of input will contain a single integer.
     - For each integer print "YES" if that integer belongs to the set otherwise print "NO".
   - sample_input

   ```
   10 6
   1 1 2 4 5 4 2 5 1 5
   1
   2
   3
   4
   5
   6
   ```

   - sample_output

   ```
   4
   1 2 4 5
   5 4 2 1
   YES
   YES
   NO
   YES
   YES
   NO
   ```

- Solution

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, q; cin >> n >> q;
```

```
    set<int> s;
    for (int i = 0; i < n; i++) {
        int x; cin >> x;
        s.insert(x);
    }

    cout << s.size() << "\n";

    for (int x : s)
        cout << x << " ";
    cout << "\n";

    for (auto it = s.rbegin(); it != s.rend(); it++)
        cout << *it << " ";
    cout << "\n";

    while (q--) {
        int x; cin >> x;
        cout << (s.count(x) ? "YES" : "NO") << "\n";
    }
    return 0;
}
```

2. https://cses.fi/problemset/task/1621

- Solution

```
#include <iostream>
#include <set>
using namespace std;

int main() {
    int n; cin >> n;
    set<int> s;
    while (n--) {
        int temp; cin >> temp;
        s.insert(temp);
    }
    cout << s.size();
    return 0;
}
```

Maps

1. Given integers n and q.
   - You will be given a list of n integers, store them in a map.
   - Each element should be mapped to its frequency.
   - Print the number of distinct elements in the map.
   - Using a range based `for` loop print the elements and their frequency in ascending order of the key.
   - Now, using a reverse iterator and a `for` loop, print the elements and their frequency in descending order of the key.
   - The next q lines of input will contain a single integer.

    - For each integer print it's frequency.
  - sample_input

```
10 6
1 1 2 4 5 4 2 5 1 5
1
2
3
4
5
6
```

  - sample_output

```
4
1->3
2->2
4->2
5->3
5->3
4->2
2->2
1->3
3
2
0
2
3
0
```

- Solution

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, q; cin >> n >> q;
    map<int, int> mp;
    for (int i = 0; i < n; i++) {
        int x; cin >> x;
        mp[x]++;
    }

    cout << mp.size() << "\n";

    for (pair<int, int> p : mp)
        cout << p.first << "->" << p.second << "\n";

    for (auto it = mp.rbegin(); it != mp.rend(); it++)
```

```
        cout << it->first << "->" << it->second << "\n";

    while (q--) {
        int x; cin >> x;
        cout << mp[x] << "\n";
    }
    return 0;
}
```