Goal

- Today we will be taking a further look at data structures in C++.
- Data structures for today:
  1. Stack
  2. Queue
  3. Priority Queue

Resources

Stack

- https://cplusplus.com/reference/stack/stack/
  - read up on what the following functions do:
    - push, pop
    - top
    - size, empty

Queue

- https://cplusplus.com/reference/queue/queue/
  - read up on what the following functions do:
    - push, pop
    - front
    - size, empty

Priority Queue

- https://cplusplus.com/reference/queue/priority_queue/
  - read up on what the following function do:
    - push, pop
    - top
    - size, empty
  - for custom ordering in set, maps and priority you can place a comparator in side a struct and call name it `operator`

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

Questions

Note: Solve the questions in order.

Stack

1. Given an integer n followed by n integers.
    - Push the integers into a stack s_1 in the order of input.
    - Create a new stack s_2.
    - While the s_1 is not empty, pop the top element and push it into s_2 only if s_2 is either empty or the top element in s_2 is smaller than the current element.
        - Print the current element from s_1 regardless of whether or not it gets pushed into s_2.
    - While s_2 is not empty, pop the elements out and print them.
    - sample_input

    ```
    10
    1 5 3 2 4 8 9 12 11 3
    ```

    - sample_output

    ```
    3 11 12 9 8 4 2 3 5 1
    12 11 3
    ```

Queue

1. https://cses.fi/problemset/task/1084
    - Solve this question like before.
    - Except after sorting the vectors, transfer all the elements in order into queues.
    - After this use only the queues to assign the apartments.

Priority Queue

1. https://cses.fi/problemset/task/1084
    - Solve this question without using vectors.
    - Directly put the elements into the priority queue.
    - By default the priority queue will return the highest elements first.
    - Using this order, solve the question.
    - Basically, you will be assigning the largest first.
        - Overall concept remains the same.
2. https://cses.fi/problemset/task/1084
    - Same task as the one above, Except make the priority queue return the elements in ascending order.