Problem:

Given an array of n integers where each integer in array is in the range [1...n] inclusive.

Goal:

Return all integers between [1...n] that do not exist in input array.

Approach #1

1.) Sort the array

2.) Have a counter from 1...n
For every number not found in input, emplace counter value to output.

3.) Return array.

Approach #2

1.) Create array of size n, populate each item 1...n
2.) Enumerate over all values in input array.
Each value - 1 is the index of array in step 1.
Set the array[value - 1] to -1
3.) Erase all values that equal -1

4.) Values left over are all numbers that do not exist in input array.

```
std::vector<int> findDisappearedNumbers(const std::vector<int>& input) {
        std::vector<int> result(input.size());
        std::iota(result.begin(), result.end(), 1);

        for (auto num : input) {
            result[num - 1] = -1;  // assuming ALL values [1...n]
        }

        result.erase(std::remove(result.begin(), result.end, -1), result.end());
        return result;

}
```