Homework.

Part 1.
Implement the following algorithms with templates.

(You can find description of each of the algorithm on the https://en.cppreference.com/w/)

- **`search_`** analogue of **`std::search`**
  Based on your implementation of **`search_`** write a templated function **`contains`** which takes two sequences and returns true if the second sequence appears in the first one, otherwise return false (actually this is almost the same thing as **`search_`**, but returning a bool instead of iterator, so basically you just need to call **`search_`** inside **`contains_`** and return true/false based on the iterator returned by **`search_`**).
- **`rotate_`** - an analogue of **`std::rotate`**
- **`merge_`** an analogue of **`std::merge`**
- **`square_sort`** - the small problem we talked about last time
- **`next_permutation_`** - an analogue of **`std::next_perutation`**
- Given an array of distinct integers, return *all the possible permutations*. You can return the answer in any order. You should write two solutions: **`all_permutations_recursive`** (a recursive solution without std algorithms and templates) and **`all_permutations_with_next_permutation`** (an iterative solution based on the returned value of **`next_permutation_`** from previous point).