These problems are to practice using common containers in C++17 such as `vector`, `queue`, `set`, `string`, etc., as well as algorithms from `<algorithm>`.

1. Write a C++ program that ...
   - Takes in strings and saves them in a `vector` until an empty string is given (using `std::cin`)
   - Prints "ALL" if every string is at least four characters long, "SOME" if some strings are at least four characters long, and "NONE" if no strings are four characters long (using `std::all_of`, `std::any_of`, `std::none_of`)
   - Prints the last character of every string (using `std::for_each`)
   - Prints the number of strings that contain a digit (using `std::count_if`)
   - Converts every string to lower case (using `std::transform` and `std::tolower`)
   - Creates a new `vector` of `pair<int, int>` where element i's first value is equal to the sum of the letters of string i (where $a = 0, b = 1, \ldots, z = 26$) and the second value is i (using `std::transform`)
   - Sorts this `vector` of `pair<int, int>` by the first value (using `std::sort`)
   - Finds and prints the string with minimum value but with value over 75, or "NONE", if no such string is found, in $\mathcal{O}(\log n)$ time (using `std::lower_bound`, which is a binary search as it turns out)
2. Write a C++ program that ...
   - Takes in four numbers x_1, x_2, x_3, x_4 (using `std::cin`)
   - Prints the $\min_i x_i$ and $\max_i x_i$ (using `std::min` and `std::max`)
   - Takes in a fifth number x_5 and clamps it between $\min_i x_i$ and $\max_i x_i$ (using `std::min` and `std::max`. Hint: This should be one line)
3. Write a C++ program that takes in an $n \times n$ matrix $A = (a_{ij})$ and computes
$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i}$$
   Where S_n is the set of permutations of the set $\{1, \ldots, n\}$, σ is a permutation, and σ_i is the ith element of the permutation σ. Also, $\operatorname{sign}(\sigma)$ is the signature of the permutation. The exact meaning is not important, but you should just flip between 1 and -1 every iteration, starting with 1 (using `std::next_permutation`).
4. Write a C++ program that ...
   - Generates a `vector` of length 1000 filled with values $1, 2, 3, \ldots, 1000$ (using `std::generate`)
   - Shuffles this vector (using `std::shuffle`. Note: Don't use `random_shuffle` as it was deprecated in C++14 and entirely removed in C++17)
   - Reverses the subarray consisting of elements at even indices (e.g. the list 1234 becomes 3214) (using `std::stack` and `std::queue`)
   - Adds a different random number between -500 and 500 to each element of the vector (using `std::rand`)
   - Counts the number of distinct elements in the vector (using `std::set`)
5. Write a C++ program that reads lines and does the following (using `std::getline` and `std::map`):
   - If the line contains the string ":q", exits the program.
   - If the line contains a string and a number separated by a single space, the program internally associates the string with that number.
   - If the line contains only a string, the program prints the number associated with the string, or "STRING NOT FOUND" if this string has not been saved before.