Problem A: Calculator Messages

Given the mappings between some digits and letters, we have to find which word we get by replacing each digit of a number by the corresponding letter and by reading the result string in reverse order. This problem can be solved with a dictionary (`std::map` in C++, `Map` in Java and `dict` in Python), a `switch` instruction or a several `if` instructions.

Problem B: Winner

Given a sequence of numbers consisting solely of 1s and 2s, we have to find the number which occurs the most. We can use two counters, one for each number and compare them to get the answer.

Problem C: Easy Contest

This problem can be solved with a greedy algorithm. First, we will generate all possible problems, i.e. all combinations of ideas and statements with associated difficulty and store them in an array. Then we will sort the problems in increasing order of difficulty. Finally we will loop over the sorted array and select the N easiest problems. We will need to keep track of already selected ideas and statements since an idea or a statement can be used only once.

Problem D: Largest Number

To get the largest number that can be formed with the given digits, sort the digits in decreasing order. It is safer to read the input numbers as strings, since we don't know the length of the numbers.

Problem E: Matrix

Given a square grid G, we have to find k, the number of square sub-grids of length x which remain the same after been rotated 90° clockwise.

We use matrix notation in the following paragraphs. $g_{r,c}$ represents the element at row r and column c in the grid. And $s_{r,c}$ represents the element at row r and column c in a sub-grid s.

To find k, we need to consider each sub-grid and compare it to its rotation. We can do this in place using a simple trick. Let $g_{i,j}$ be the top-left element of a sub-grid s. $g_{i,j}$ is mapped to $s_{0,0}$ and any other element $g_{p,q}$ that belongs to the sub-grid is mapped to $s_{p-i,q-j}$. Note that it is always possible to get the original coordinates of a sub-grid element $s_{c,d}$ in G since $(p, q) = (c + i, d + i)$.

When a sub-grid is rotated, $s_{i,j}$ moves to the position of $s_{j,x-i-1}$. So for each element $s_{i,j}$ of the sub-grid, we have to find the element $s_{a,b}$ that will be moved to the position of $s_{i,j}$ after the rotation and check if they are equal. It is easy to find a and b since $(i, j) = (b, x - a - 1)$. Once the values of a and b are computed, we just have to compute the original coordinates of $s_{a,b}$ and $s_{i,j}$ in G and compare the values.

Problem F: Sequence

Given a sequence of integers, we have to find which of the following categories it belongs to: **increasing**, **decreasing**, **increasing then decreasing** or **decreasing then increasing**. The simplest way to find the answer is to loop over the sequence and check the relative order of consecutive elements.

Note: This problem can be solved easily in C++ by using STL functions `is_sorted` and `is_sorted_until`.

Problem G: Painting

The judges were very kind and gave the formula to get the neighbors of a cell. The only task left was how to flip the value v of cell when needed. This can be done in several ways.

1. Increase v by 1 each time the cell state changes, and output v modulo 2 at the end.
2. Each time the state of a cell has to change, compute $(v + 1)$ modulo 2.
3. Flip the value of a cell by computing v *xor* 1.

Problem H: Conferences Day

This problem can be modeled as a graph and solved using Kahn's algorithm for topological sort. The challenge left is how to select the next speech so that it will always be the longest that can start. This can be achieved by using a heap instead of a simple FIFO queue to get the next available speech.

Problem I: Bridges and Trees

Another problem from graph theory. We have to find all the bridges in a graph and then compute for each bridge the minimum and maximal spanning trees of the connected components obtained after removing the bridge from the graph. Since bridges are always part of the spanning tree, once we find the min and max spanning trees of the graph, we will have to remove each bridge from the spanning trees and compute the cost of spanning trees of the connected components remaining.

Problem J: Minimize Me

The minimum value of S is reached when K is the arithmetic mean of the collection $T_i - i$. In order words,

$$K = \frac{1}{n}\sum_{i=1}^{n} T_i - i$$

Problem K: Mango

Given a sequence of letters, we have to count how many time each letter appears and finally sort the sequence. This problem is a little mix of problems B and D.

Problem L: A Little Trick

Given two numbers X and Y, we have to find the maximum of $X + Y$, $X - Y$, $X * Y$ and $Y - X$.

Problem M: New Year and Problems Robbers

We have a grid consisting of several type of cells. Each type of cell has an associated blocking cost and some cells can't be blocked. We have to find the cheapest set of cells to block so that given a starting cell, it will be impossible to leave the grid.

To find the solution, we can simulate the strategy of a robber and try to find a way to leave the grid by testing all possible paths. As soon as a cell is reached, we check if the cell can be blocked or not. If the cell can be blocked, we block it and we don't visit it neighbors. If the cell can't be blocked, we have to explore its neighborhood. We continue this process until we reach a cell which is on an edge that can't block or there is no more cell to explore. In the first case, we report that it is not possible to stop the robbers. In the second case, we can stop them and the total cost is the sum of the cost of the blocked cells.