Problem C. Invocations

Go through all the tests, starting from the first one, maintain a variable max (initially set to 99) and a vector of test numbers where the maximum is reached (initially empty).

If the solution runs slower on some test than half the time limit, mark the corresponding cell as yellow or red (depending on whether the solution fits into a whole TL), set the value of max to plus infinity, and clear the vector of maximums. Otherwise, mark the corresponding cell as green.

If the solution runs faster on some test than half the time limit, but slower than the current value of max, modify the value of max, clear the vector of maximums, and write the test number there. If the solution runs for the current value of max on some test, add the test number to the vector of maximums. After going through all the tests, go through all the tests from the vector of maximums, mark the corresponding cells as blue, and then the coloring is ready — output the colors of each cell.

Problem I. Exceptional Set

A string of length n cannot be split into more than n non-empty strings (otherwise their total length will be at least $n + 1$). Split into n strings of one character each (if there is more than one character in any string, then the total length will again be at least $n + 1$). A string of one character is a palindrome, thus the constructed set is exceptional. It is obvious from the construction that such a set is unique.

Therefore, the solution should read the string and output it character by character, outputting each character on a new line.

Problem K. Diversity Of Strings

Let's consider the blocks consisting of consecutive asterisks and bounded on the right and left by a letter (or the beginning/end of the string for the first and last such block). If there is only one such block (i.e., on the left — the beginning of the string, on the right — the end), then the maximum number of uniform strings is 26 — all asterisks in the block can be replaced by any letter (but in such a way that all letters are the same).

Let's calculate the answer in the case where there is at least one letter in the string. If the block has a letter on the left and the end of the string on the right (or a letter on the right and the beginning of the string on the left), then if at least one letter in the block does not match the letter on the edge, the diversity will increase compared to the case when all letters match the letter on the edge, i.e., in this block, the letter is chosen in a unique way. Similarly, regarding the blocks where the first and last letters match — you can get diversity 1 in a unique way by filling the block with this letter. Thus, it remains to count the number of cases for each block where the first and last letters are different. Inside this block (including the boundaries), the minimum diversity cannot be less than two (because the beginning and end are different). To make the minimum diversity equal to two, there should be only one "transition" between the letters equal to the first and the letters equal to the last. The number of ways to do this is equal to the number of asterisks in the block plus one (let the first letter be `a`, the last letter be `b`, and there are l asterisks between them, then all strings are suitable in which the first k letters ($0 \leq k \leq l$) are `a`, the rest are `b`, and there are $l + 1$ such strings). Since the deciphering of asterisks inside the blocks is independent, the answer will be the product of the values $l_i + 1$ for all blocks where the first and last letters are different (here l_i is the number of asterisks in the block).

Problem L. Champoins League

If a match ends in a draw, the total points scored by the teams increases by 2 (1+1), otherwise the total increases by 3 (0+3 or 3+0). In total, there are $N(N-1)$ matches played. Thus, if the number is between $2N(N-1)$ and $3N(N-1)$, the situation is possible, and the number of drawn matches is equal to $3N(N-1) - K$ (as each drawn match reduces the total by 1).