Level 1 Contest 1 | TLE

This is the Editorial for Contest 1 of Level 1 in the TLE Batch.
Each Problem has a detailed Approach or solution explanation and a Code link. Use this editorial to compare your solutions with the given approach. Remember, it's important to try solving the problems on your own before consulting the editorial for the best learning experience.

A. Sum

There are multiple ways to solve this problem.

Most interpreted languages have some function that takes the string, evaluates it as code, and then returns the result. One of the examples is the **eval** function in Python. If the language you use supports something like that, you can read the input as a string and use it as the argument of such a function.

Suppose you use a language where this is impossible. There are still many approaches to this problem. The most straightforward one is to take the first and the last characters of the input string, calculate their ASCII codes, and then subtract the ASCII code of the character 0 from them to get these digits as integers, not as characters. Then you can just add them up and print the result.

Code: https://pastecode.io/s/no03b08x

B. GCD Game

- Hint:

  Answer for any $n \geq 2$ is equal to $\lfloor \frac{n}{2} \rfloor$.

- Solution:
  Let the maximum gcd be equal to g. Since the two numbers in a pair are distinct, one of them must be g and both of them must be divisible by g. The smallest multiple of g, greater than g, is $2 \cdot g$. Since each number in the pair must be \le n, we must have $2 \cdot g \leq n$, or $g \leq \lfloor \frac{n}{2} \rfloor$. We can achieve $g = \lfloor \frac{n}{2} \rfloor$, by choosing $\lfloor \frac{n}{2} \rfloor$ and $2 \cdot \lfloor \frac{n}{2} \rfloor$.

Time Complexity: $O(1)$

Code: https://pastecode.io/s/vpes5q4b

C. Card Game

- **Hint:**
  Selecting the smallest two elements on the table is a good choice in the first move.

- Solution:
  Let b denote the sorted array a. Assume that b contains only distinct elements for convenience. We prove by induction on n that the maximum final score is $b_1 + b_3 + \ldots + b_{2n-1}$.

  For the base case $n = 1$, the final and only possible score that can be achieved is b_1.

  Now let $n > 1$.

  Claim: It is optimal to choose b_1 with b_2 for some move.
  - **Proof:**
    - Suppose that in some move, b_1 is chosen with b_i and b_2 is chosen with b_j, for some $2 < i, j < 2n$, $i \neq j$.
    - The contribution to the score according to these choices is $\min(b_1, b_i) + \min(b_2, b_j) = b_1 + b_2$.
    - However, if we had chosen b_1 and b_2 in one move, and b_i and b_j in the other move, the score according to these choices is $\min(b_1, b_2) + \min(b_i, b_j) = b_1 + \min(b_i, b_j)$.
    - As $i, j > 2, b_i > b_2$ and $b_j > b_2 \implies \min(b_i, b_j) > b_2$. Thus, we can achieve a strictly larger score by choosing b_1 with b_2 in some move.
    - The choice of selecting b_1 and b_2 contributes a value of b_1 to the score. The maximum score that can achieved for the remaining numbers $[b_3, b_4, \ldots, b_{2n}]$ on the whiteboard in the remaining moves is $b_3 + b_5 + b_7 + \ldots b_{2n-1}$ by the induction hypothesis.

Note that we can extend the arguments for the case where a has duplicate elements.

Code: https://pastecode.io/s/f4mggaos

D. Number Test

Let's create an integer variable (initially set to 0) that will contain the number of considered liked integers. Let's iterate over all positive integers starting with 1. Let's increase the variable only when the considered number is liked. If the variable is equal to k, let's stop the iteration and output the last considered number.

Since the answer for $k = 1000$ is $x = 1666$, the count of considered numbers is at most 1666 so the solution will work on the specified limitations fast enough.

Code: https://pastecode.io/s/a2tqzjd4

E. Energy Sequence

To solve this problem you should simply read the numbers to strings and add leading zeroes to the shorter one until the numbers will be of the same length. After that you should simply compare them alphabetically.

Code: https://pastecode.io/s/f5ypawc2

F. Yet Yet Yet Another Treasure Hunt

Let's keep an array a of booleans, a_i denoting whether or not some team has found the i-th treasure already. Now we can iterate through the string from left to right and keep a running total tot. If a_i is true (the i-th treasure has already been found), increase tot by 1; otherwise, increase tot by 2 and set a_i to be true.

The time complexity is $\mathcal{O}(n)$.

Bonus: the answer is always $n + \text{number of distinct characters in } s$. Can you see why?

Code: https://pastecode.io/s/h3oaxqb4

G. Coolant Flowing

It's obvious that we should block several largest unblocked nozzles. Let's first sort them. After that, let's iterate through the number of blocked nozzles, maintaining the sum of sizes of non-blocked nozzles S. With the value S it is easy to compute if the flow from the first nozzle is large enough or not. Just output the number of blocked pipes at the first moment when the flow is large enough. The complexity is $O(n)$.

Code: https://pastecode.io/s/5crjiz8n

H. Energy Level 3

Let cnt_i be the number of elements of a with the remainder $i\%3$. Then the initial answer can be represented as cnt_0 and we have to compose numbers with remainders 1 and 2 somehow optimally. It can be shown that the best way to do it is the following:

Firstly, while there is at least one remainder 1 and at least one remainder 2, let's compose them into one 0. After this, at least one of the numbers cnt_1, cnt_2 will be zero, then we have to compose remaining numbers into numbers divisible by 3. If $cnt_1 = 0$ then the maximum remaining number of elements we can obtain is $\lfloor \frac{cnt_2}{3} \rfloor$ (because $2 + 2 + 2 = 6$), and in the other case ($cnt_2 = 0$) the maximum number of elements is $\lfloor \frac{cnt_1}{3} \rfloor$ (because $1 + 1 + 1 = 3$).

Code: https://pastecode.io/s/htytg32o

I. Musical Night

What does it mean that no pair of adjacent musicians should have the same instrument? It means that either all musicians on odd positions play flutes and all musicians on even positions play drums, or vice versa. So, we need to check these two cases.

Let's try to solve a case when we have to find a number d such that $a_1, a_3, \cdots$ are divisible by d, and $a_2, a_4, \cdots$ are not. What does it mean that d divides all of the numbers $a_1, a_3, \ldots$? It means that d divides the $gcd(a_1, a_3, \ldots)$, where gcd represents the greatest common divisor. Let's calculate this gcd using the Euclidean algorithm or some built-in functions in $O(n + \log A)$.

Okay, now we need to check all divisors of the $gcd(a_1, a_3, \ldots)$ and find if any of them does not divide $a_2, a_4, \ldots$. So, we have to factorize gcd and generate all of its divisors...

or do we? In fact, if $gcd(a_1, a_3, \dots)$ divides any of the numbers $a_2, a_4, \dots$, then every divisor of gcd also divides that number. So, the only two numbers we have to check as candidates for the answer are $gcd(a_1, a_3, \dots)$ and $gcd(a_2, a_4, \dots)$.

Code: https://pastecode.io/s/odk4p4wi

J. Energy Grid

To solve this problem we will store two sets — the set of the verticals, in which there is at least one orb, and the set of the horizontals, in which there is at least one orb. We need to iterate through the orbs in the order that we put them on the grid and calculate the number of the cells affected, after each orb.

Let we put current orb in the cell (x, y). We need to add y in the set of verticals and add x in the set of horizontals. Let after it the size of the set with verticals equals to szY and the size of the set with horizontals equals to szX. Then it is easy to show, that the number of the cells affected equals to $cnt = szY \cdot n + szX \cdot n - szX \cdot szY$, so, the number of cells which are not affected equals to $n \cdot n - cnt$.

Code: https://pastecode.io/s/czc5eqgf