Problem A. Parallelepiped

Using the formula for the volume of a parallelepiped, we get the answer $n \cdot m \cdot k$.

Problem B. Grandchildren

The answer is $x \cdot (x+1) \cdot (x+2)$.

Problem C. Colorful Balls

Problem Author:	**Oleksandr Tymkovich**
Prepared by:	**Oleksandr Tymkovich, Feysa Bohdan**
Editorial by:	**Oleksandr Tymkovich**

In this problem, it is sufficient to consider all possible cases: which box will contain only blue balls, which will contain only yellow balls, and which will remain empty.

For example, if you want a box to contain only yellow balls, then you need to rearrange all the blue balls from it. If the task is to make the box empty, then you need to rearrange all the balls from it.

Problem D. Colorful String

Problem Author:	**Oleksandr Tymkovich**
Prepared by:	**Oleksandr Tymkovich**
Tutorial by:	**Oleksandr Tymkovich**

If the number of zeros in the string is equal to the number of ones, the answer is «`Yes`». One of the possible ways is to color all the ones in red. We will get all zeros in the first string and all ones in the second.

Can there be other cases where the answer is «`Yes`»? No, because in other cases, there will either be too many zeros, which will lead to having 0 in the same position in both strings, or too many ones.

Problem E. Painting stones

Problem Author:	**Oleksandr Tymkovich**
Prepared by:	**Oleksandr Tymkovich**
Editorial by:	**Oleksandr Tymkovich**

Solution for $n \le 1\,000$:

Let's iterate over the color in which all the stones will be painted in the end. To find the minimum number of paintings for a certain color, you can follow the following greedy algorithm:

- if the color of the stone is the one we are iterating over, then move on;
- otherwise, paint the current stone and the next one in the color we are iterating over.

Full solution:

Let's assume we are iterating over the color C. For which minimum number of operations can we paint the segment $[l, r]$ where the end stones are of color C, and there are no other stones of color C except the end ones? From the algorithm for the solution $n \le 1000$, the answer is $\lfloor \frac{r-l}{2} \rfloor$.

We can divide the array into such segments, and the overall answer for the color C is the sum of the answers for each segment. (Also, don't forget that you may need to paint the prefix and suffix).

So the solution is to calculate each color separately. For the color C, we have provided an algorithm with a complexity of $O(occ(C))$ where $occ(C)$ is the number of occurrences of the number C in the array. In total, for all colors, the solution has a complexity of $O(n)$.

Problem F. Pairwise Product

Problem Author: **Pavlo Tsytsi**
Problem Prepared by: **Pavlo Tsytsi**
Editorial by: **Pavlo Tsytsi**

$(a_1 + ... + a_n)^2 = a_1^2 + ... + a_n^2 + 2 \cdot (a_1 \cdot a_2 + ... + a_1 \cdot a_n + a_2 \cdot a_3 + ... + a_2 \cdot a_n + ... + a_{n-1} \cdot a_n)$. The part we need is in the second set of parentheses, so all we need is the square of the sum on the segment and the sum of squares on the segment. This can be found using prefix sum. After that, divide by 2 modulo.

Problem G. Sashko-Array Constructor

Problem Author: **Tsitsei Pavlo**
Problem Setter: **Bogdan Feysa**
Editorialist: **Oleksandr Tymkovich**

If the maximum prime divisor of x is greater than d, then the answer is -1. This is because to obtain a product equal to n, all its prime divisors must be used.

To find an array of minimum length, you can follow the same greedy algorithm:

- if the number is equal to 1, then finish the work.
- let k be the maximum divisor of x that is not greater than d. Write the number k to the array and divide x by k.

Finding the divisor can be done with a complexity of $O(\sqrt{x})$.

Problem H. Sets

Problem Author: **Tsitsei Pavlo**
Problem Setter: **Tsitsei Pavlo**
Editorial: **Tsitsei Pavlo**

The answer is $(k+1)^n$.

For $k = 1$, the answer is 2^n, as it is the number of subsets. Consider any $k > 1$. Since the specific set being considered does not matter, and only the number of elements is important, all subsets can be grouped by the number of elements. Then the answer will be $\sum_{i=0}^{n} \binom{n}{i} \cdot f(i, k-1)$. Assuming $f(i, k-1) = k^i$, therefore $\sum_{i=0}^{n} \binom{n}{i} \cdot f(i, k-1) = \sum_{i=0}^{n} \binom{n}{i} \cdot k^i = (k+1)^n$ using the binomial coefficient formula.