Task: GCD
Gingerbread cookies (author: Kostiantyn Denysov)

BOI 2025, Day 1: Analysis.

If you try to design an interesting large example you immediately hit a wall. Obviously, there are examples with answer 0, where the GCD of the input numbers is 1. There are also examples with answer 1, where the input numbers have some common prime divisor (having a divisor greater than 1 is the same as having a prime divisor; we prefer to talk about prime divisors), and after increasing some number we get the GCD of 1. But can we have answer 2 or more? This would mean that no matter which number we increase by 1, we obtain a list of numbers with some common prime divisor. In other words, for every index i there should be a prime number p_i that divides all $a_1, \ldots, a_{i-1}, a_i + 1, a_{i+1}, \ldots, a_n$. But note that the same prime cannot divide both $a_i + 1$ and a_i, so all the primes $p_1, \ldots, p_n$ are different. Moreover, a_n has to be divisible by all those primes, except p_n, so also by their product. We know that $a_n \leq 10^7$, while already the product of the smallest 9 prime numbers is $\approx 2 \cdot 10^8$. It follows that for $n \geq 10$ the answer is necessarily at most 1. Moreover, we can easily distinguish between answers 0 and 1 by computing the GCD of all the input numbers.

Interesting things start to happen for $n \leq 9$. But this size of inputs suggests that a brute-force algorithm should be enough here. How does it work? For consecutive r, starting from $r = 0$, we consider all sequences of numbers obtained from the initial one by adding r cookies. If for any such sequences has GCD of 1, we output r and we finish; if not, we increase r. Some effort is needed to generate all those sequences, but this can be somehow done using a recursive procedure (it decides how many cookies will be added to the first unhandled box, and then it calls itself recursively to distribute the remaining cookies among the remaining boxes). The number of possibilities to check is $\binom{n+r-1}{r}$ (we want to split the r cookies into n sets; to this end, among $n + r - 1$ positions we can draw r cookies and $n - 1$ lines splitting the sets). We just need to get convinced that this number is not too high.

We can see that after adding at most 11 cookies, the GCD will become 1, for any $n \geq 2$. This ensures that the algorithm is fast enough, because $\binom{n+r-1}{r} \leq 10^5$ for any $n \leq 9$ and $r \leq 11$. Thus, suppose to the contrary that no matter how we add at most 11 cookies, the GCD of the first two numbers will be greater than 1. Among the numbers $a_1 + b$ for $b \in \{0, \ldots, 5\}$ there are precisely two that are not divisible neither by 2 nor by 3 (i.e., are 1 or 5 modulo 6). The difference between them is 2 or 4, so at least one of them is not divisible by 5. Thus, fix $b \in \{0, \ldots, 5\}$ such that $a_1 + b$ is not divisible neither by 2, nor by 3, nor 5. Then, for each $c \in \{0, \ldots, 6\}$ there should be a prime number that divides both $a_1 + b$ and $a_2 + c$, and this is not $2, 3, 5$. For each c this has to be a different number (the same $p \geq 7$ cannot divide two numbers which differ by less than 7). The product of all of them is at least $7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \approx 2 \cdot 10^8$, while it divides $a_1 + b \leq 10^7 + 5$; a contradiction.

Let us remark, that the actual bounds for the result are even a bit better. Namely for $n = 2$ the result can be at most 5, for $n = 3$ it can be at most 3, for $4 \leq n \leq 7$ it can be at most 2, and for $n \geq 8$ it is necessarily 0 or 1. Another remark is that the limit of 10^7 for a_i occurring in this problem is completely arbitrary. The same naive solution would work equally well for $a_i \leq 10^{18}$, maybe only it would be a bit more difficult to get convinced that indeed in all cases the program is fast enough.