Hour 5:

After successfully opening the truck, you lug all the solar panels out of the truck and place them on the ground, hooking them up to a large scale battery. Each solar panel has an "power" number. You set up the solar panels into a line, which can be represented as an array of values.

The array has length n. You can do two things to this array in order to make it work, but any task costs manpower. You can either remove some subset of length $m < n$ panels, but it will require $m * a$ workers to do so. This can only be done once. You can also edit the power number of any solar panel in the array by increasing or decreasing the number by 1, which takes b workers. However, you can only edit each panel once before it locks you out. a and b are given.

The only way to successfully link up the solar panels into a working array is if the greatest common divisor of all the power numbers in it is greater than one. However, you also would rather use as little workers as you can for this task. Calculate the minimum number of workers necessary to use in order to successfully link up the solar panel array.

Input format: the first line contains an integer which is x, the number of integers on line 2. Line 2 contains x space-separated integers. Line 3 contains a and b respectively, separated by a space.

Output format: an integer containing the minimum number of workers.

Example Input
3
4 2 3
1 4

Example Output
1

Example Walkthrough
In this example, we are given that the length of the array is 3. The array's power number contents are 4, 2, and 3. Value a is 1 and value b is 4. The best way to minimize the amount of workers that you use such that the remaining power numbers have a greatest common divisor greater than 1 is to remove 3 from the array, leaving 4 and 2. 4 and 2 have a greatest common divisor of 2, which is greater than 1. Since we remove a subset of length 1, $m = 1$. Since a is given to equal 1 and the only operation we perform is removing 3, we conclude that the only amount of workers necessary is 1 * 1, or 1.

The submission form for this hour can be found at https://forms.gle/qYfxqKFutfwjJRZW6

Rules for submission of programs

The solution must be in a common enough language that it can be executed or compiled without great effort to hunt down an interpreter or compiler. If a program requires a specific command line flag, makefile, etc, include that and delineate it.

The solution must be written in good faith, be submitted unobfuscated / unminimized, and written to perform the problem the task requires and only the problem the task requires. If you are wondering whether or not something breaks the rules (or are trying to exploit a loophole), don't do it.

The input for the program will be provided in a file called `input` in the local directory, and the output for the program must be written into a file called `output` in the local directory.

Solutions may not:

- Open, read, or write to any files besides the input and output files
- Connect to the internet or any devices on the local network
- Use more than 512MB of memory or 15 seconds of CPU time
- Require anything other than default user privileges to run
- Be dependent on operating-system specific APIs
- Be submitted precompiled or in standalone executable form
- Play sound, write to the display, or access any output device
- Be larger than 10MB
- Use any libraries other than the standard library that comes with the language. To run, no packages or other code should need to be installed.