

Subject 2018.2.1 – Repair Shop

You need to write a piece of software for a laptop repair shop to help technicians fix incoming devices. The shop has in stock the following types of items: **case**, **keyboard**, **motherboard**, **screen**, **processor**, **memory** and **SSD**. For a laptop to function correctly, it needs to have one of each of the components above working correctly. In service, clients are bringing in laptops in which one or more components are broken, in which case the defective components need to be replaced, or laptops with all hardware parts functional, but with software damage, in which case no hardware repairs are needed. If a laptop can't be repaired, it is immediately stripped down, and functional parts are used to fix other broken laptops that arrive later.

Requirement

Given the amount of parts in the original stock and the laptops that are brought in for repair, with their working and broken parts, you need to determine how many laptops the repair shop will fix.

Input data

The first input line will contain seven natural numbers, separated with spaces, which represent the amount of parts in the repair shop stock, in each category, in the exact order as specified above (in the introduction). On the second line there is exactly one strictly positive integer, **N**, representing the number of laptops brought in for repairs. The following **N** lines each contain a sequence of 7 digits of 0 and 1 (for each of the seven categories), separated by space, where 0 represents a broken component and 1 represents a working component, in the same order as above. All lines end with the *newline* (\n) character, by pressing the *Enter* key.

Output data

The program will output (on the standard output stream - *stdout*) a single integer value representing the number of laptops that the repair shop will fix.

Please read the requirement carefully! Displaying the results must be done exactly as it was requested! In other words, on the standard output stream you must not print anything in addition to the requirement of the problem; because of the automatic evaluation, any additional displayed characters other than those indicated, will leads to an incorrect result and therefore to a "fail" grade.

Restrictions and Additional Information

1. $0 < N < 10000$
2. The laptops are processed in the order they arrive.
3. **Careful:** Depending on the programming language chosen, the file containing the code must have one of the extension .c, .cpp, .java, or .m. The web editor will not automatically add these extensions and their absence would prevent the compilation of the program!
4. **Careful:** The source file must be named <name>.<extension> where name is the surname of the candidate and the extension is chosen according to the previous point. Attention to restrictions imposed by Java for class and file names!

Examples

Input	Output	Explanation
3 4 1 3 0 4 5 4 1 0 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1	3	<p>The original stock is: Case = 3 Keyboard = 4 Motherboard = 1 Screen = 3 Processor = 0 Memory = 4 SSD = 5</p> <p>In the same order we have the components of the first laptop which has a defective keyboard and processor. The processor can't be replaced, because there's none in stock, the laptop is disassembled, and the functional parts will be used to fix following laptops.</p> <p>The second laptop has a broken case and motherboard. They are both in stock, so the laptop is fixed.</p> <p>The third laptop has a broken motherboard. It's no longer in stock, but we can use the one we got from the first laptop which was stripped down, so this laptop is fixed as well.</p> <p>The fourth laptop is entirely functional, so we just count it and get a total of 3 repaired laptops shipped out by the service.</p>
1 1 1 1 1 1 1 5 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1	4	<p>The first four laptops are all fixed by parts from the stock. For the last one, there are no longer any parts available so the output.</p>

Time to work: 120 minutes