Science Against Spam (spam)

Giorgio is working on a scientific paper involving Bayesian spam filtering, a statistical method used by mail providers to detect spam emails. This method can be very efficient in practice, but Giorgio wants to measure how much exactly.

For this research, he will first implement a naïve *content-based* filter. This filter compares each word in the email text against a set of "bad" words: if the email contains a bad word (e.g. *dollars*) then it's likely to be spam. Moreover, the filter compares the words against another set, this time made of "good" words: if the email contains a good word (e.g. *paper*) then it's likely to be a legitimate mail.

Help Giorgio count how many of his emails are likely spam and how many are likely legitimate. If an email is both likely spam and likely legitimate at the same time (or if Giorgio doesn't have enough information to tell if it's likely spam or likely legitimate) then it's a *dubious* mail: it should not be counted at all.

> ☞ Among the attachments of this task you may find a template file `spam.*` with a sample incomplete implementation.

Input

The first line contains an integer B, the number of bad words. The following line contains B space-separated words (the bad ones). The following line contains an integer G, the number of good words. The following line contains G space-separated words (the good ones). The following line contains an integer E, the number of emails in Giorgio's inbox. Then E emails follow: each email is represented by two consecutive lines, the first one contains an integer N_i, the number of words in the i-th email, the second one contains N_i space-separated words (the email content).

Output

You need to write a single line with two integers: the number of emails likely to be spam, and the number of emails likely to be legitimate.

Constraints

- $1 \leq B \leq 1000$.
- $1 \leq G \leq 1000$.
- $1 \leq E \leq 1000$.
- $1 \leq N_i \leq 1000$ for each $i = 0 \ldots E - 1$.
- Every word is between 1 and 20 characters long and contains `'a'`—`'z'` letters only.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [**5 points**]: Examples.
- **Subtask 2** [**25 points**]: $N_i = 1$.
- **Subtask 3** [**40 points**]: $B, G, E \leq 100$.
- **Subtask 4** [**30 points**]: No additional limitations.

Examples

input.txt	output.txt
3 dollars invest enlarge 4 scientific paper deadline professor 3 6 hey giorgio send me the paper 8 now is the time to invest in bitcoin 11 i wonder why we dont invest more for the scientific advancement	1 1
1 yikes 1 yummy 5 2 yummy yummy 2 likes bikes 2 yummy bikes 2 yummy yikes 2 bikes yikes	1 2

Explanation

In the **first sample case** the last email is the only dubious one (*invest*, *scientific*). The first one is likely legitimate, the second is likely spam.

In the **second sample case** the last email is likely spam, while the first and third emails are likely legitimate.