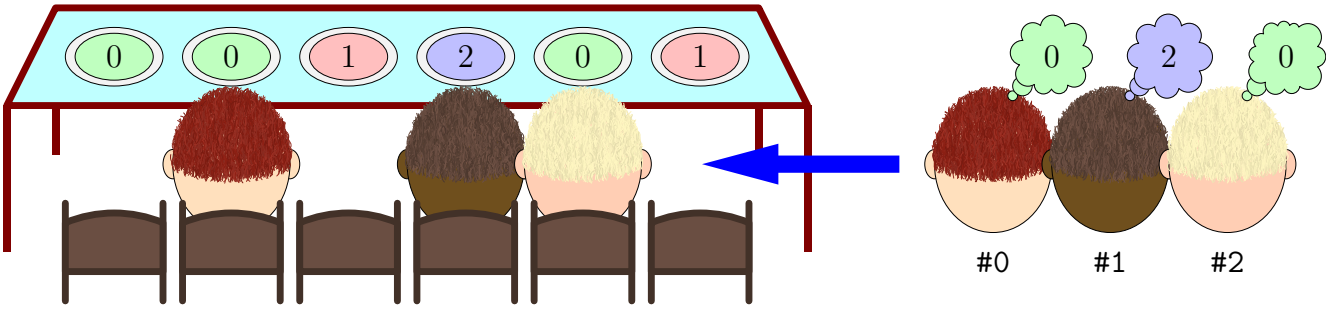
Cena di gala (cena)

After the competition, the P participants of the Italian Olympiad in Informatics (labeled from 0 to $P-1$) will attend the gala dinner, in a traditional Molise fashion.

The banquet consists of S plates labeled from 0 to $S-1$, each plate containing one of the "100 best Molise delicacies" personally chosen by the chef: *composta molisana*, *spezzatino di pecora*, *agnello con cicoria* and other 97 typical dishes from Molise.

Once at the gala dinner, the participants will head towards the beginning of the buffet, going from right to left (from plate $S-1$ to plate 0) and they will choose to either sit down or skip to the next plate. Each participant has a favorite dish, so he/she will sit down only in one of the spots where his/her favorite dish is served. The participants will arrive at the banquet in the same order they were ranked in the contest.

It is not allowed to climb over others, so when a participant chooses to sit down in front of the i-th plate, then all the plates in positions $j < i$ will be unreachable to the following participants. Once seated, the participants will keep the same order in which they arrived. It is possible to have "empty spots" where no one is seated. The participants don't necessarily need to choose the first "favorite" dish they see on the table: *they can choose any plate* among those containing their favorite dish, as shown below.



Unfortunately there is a problem! Mojito and Chupito, Monica's dog and cat respectively, escaped her sight and entered the banquet room before the participants' arrival. The two brats jumped on the long table with the intent of eating some of the dishes:

- Mojito jumped on the *left* side of the table, to eat plates starting from the 0-th.
- Chupito jumped on the *right* side of the table, to eat plates starting from the $(S-1)$-th.

Monica, realizing that her pets were missing, quickly ran to the buffet and now she has to choose how and when to stop them. It is possible to stop Mojito and Chupito at any point, even before they start to eat. However, touched by the cute fluffiness of her puppies, Monica would like to let them eat for a while. Of course, she should make sure that there's still food for the contestants.

Let's denote with A and B respectively the number of dishes eaten by Mojito and by Chupito. Compute **how many different pairs** (A, B) exist such that, even "sacrificing" the first A and the last B plates of the table, **there exists at least a strategy** that allows the participants to all sit down in front of a plate containing their favorite dish.

✍ **Important:** participants are curious to try Molise delicacies. Thus, regardless of the value of P, the number of participants *with the same favorite dish* is not greater than **1000**.

Implementation

You should submit only one file, with extension `.c` or `.cpp`.

> ☞ Among the attachments of this task you will find a template `cena.c` and `cena.cpp` with an example of implementation.

You have to implement the following function:

C/C++	`long long conta(int S, int s[], int P, int p[]);`

- The integer S is the number of plates on the table.
- The array s (indexed from 0 to $S-1$) specifies the type of food for the i-th plate.
- The integer P is the number of participants.
- The array p (indexed from 0 to $P-1$) specifies the preference of the i-th participant.

The grader will call the function `conta` and will print the returned value in the output file.

Grader di prova

In the directory of this problem there is a simplified version of the grader used during evaluation, that you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you have to implement and writes in `stdout`.

The input file is formed by three lines:

- Line 1: integers S and P.
- Line 2: values `s[`i`]` for $i = 0 \dots S-1$.
- Line 3: values `p[`i`]` for $i = 0 \dots P-1$.

Output file is formed by just one line, containing:

- Line 1: the value returned by function `conta`.

Constraints

- $1 \leq P \leq S \leq 100\,000$.
- $1 \leq P \leq 50\,000$.
- $0 \leq$ `s[`i`]`, `p[`i`]` < 100.
- There aren't 1001 participants with the same preference.

Scoring

Your program will be tested on several test cases grouped in subtask. To achieve the score of a subtask, you need to correctly solve all of its test cases.

- **Subtask 1 [0 points]**: Sample cases.
- **Subtask 2 [13 points]**: $S \leq 100$ and $P \leq 100$.
- **Subtask 3 [23 points]**: $P \leq 1000$ and all the participants have the same preference.
- **Subtask 4 [27 points]**: $S \leq 10\,000$ and $P \leq 100$.
- **Subtask 5 [16 points]**: $P \leq 100$.
- **Subtask 6 [21 points]**: No additional limitations.

Examples

stdin	stdout
6 3 0 0 1 2 0 1 0 2 0	4
9 3 0 1 0 1 1 0 1 0 1 0 0 0	8

Explanation

The **first sample case** is the one illustrated in the task description. To guarantee a sitting strategy for all participants, there are 4 possible ways to stop Mojito and Chupito.

- 0[01<u>2</u>0]1 – Monica lets Mojito eat 1 dish and Chupito eat 1 dish.
- [00<u>1</u>2<u>0</u>1] – Mojito 0 dishes, Chupito 0 dishes.
- [<u>0</u>01<u>2</u>0]1 – Mojito 0 dishes, Chupito 1 dish.
- 0[01<u>2</u>0<u>1</u>] – Mojito 1 dish, Chupito 0 dishes.

The table area which hasn't been touched by the two hungry pets is indicated between square brackets. Also, to show that all these ways guarantee a sitting strategy for participants, a possible disposition of the participants is underlined.

In the **second sample case** all participants have the same preference, any way to stop Mojito and Chupito that leaves at least 3 plates of type '0' is thus valid.

- [<u>0</u>10<u>1</u>1<u>0</u>]101
- [<u>0</u>1<u>0</u>11<u>0</u>1]01
- [01<u>0</u>11<u>0</u>1<u>0</u>]1
- [<u>0</u>1011<u>0</u>1<u>0</u>1]
- 0[1<u>0</u>11<u>0</u>1<u>0</u>]1
- 0[1<u>0</u>11<u>0</u>1<u>0</u>1]
- 01[<u>0</u>11<u>0</u>1<u>0</u>]1
- 01[<u>0</u>11<u>0</u>1<u>0</u>1]