Задача A. Candy Mood

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

You have a box of delicious candies. Candies look the same, but have different fillings. Some of them are tasty, others are not. You eat candies by uniformly choosing one candy at a time among the remaining ones and eating it. Your mood is then increased by a specific value that depends on the taste of each candy. If the candy is tasty, the value is nonnegative, otherwise it's negative (and your mood is actually decreased). Besides, if the candy that you've just eaten is not tasty, you won't eat any more candies at all. If it appears that all candies in the box are tasty, you will eat them all.

You know the number of candies and tastiness of each candy. You need to find the expected value of total change of your mood.

Формат входного файла

The first line of input contains a single positive integer n, the number of candies in the box ($1 \le n \le 100$). The second line holds n integers a_i where a_i represents the change of your mood caused by i-th candy ($|a_i| \le 100$). Candy number i is not tasty if a_i is negative.

Формат выходного файла

Output a single number representing expected change of your mood. The absolute or relative error must not exceed 10^{-9}.

Примеры

stdin	stdout
5 4 1 3 0 2	10.0000000000
2 -1 -2	-1.5000000000

Note

The expected value of a random variable is the value one would expect to find if one could repeat the process an infinite number of times and take the average of the values obtained.

Задача B. Marbles

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	256 megabytes

A rectangular board consists of $N \cdot M$ cells (N rows each consisting of M cells). Each cell initially holds one marble ball, each marble's color by itself is different from all other marbles' colors and the board color.

On each turn, you are allowed to take a marble from cell (r, c) and move it to one of the four cells (x, y) adjacent by an edge, removing a marble in the cell (x, y) from the game, if there is one.

Young Jonathan made several moves and then took a photo of the board setup. When Jonathan's father saw the picture he decided to calculate the number of different pictures that can be made from the initial setup (i. e., take the initial setup, make zero or more moves, take a photo, roll back to the initial setup and make turns again to take another photo, etc.). All the photos are considered to be taken from the same position of the camera without any board movements and rotations. The whole board is visible on each photo.

Формат входного файла

The only line of input holds two integers N and M ($1 \le N, M \le 50$): the number of rows and columns respectively.

Формат выходного файла

Output the number of different pictures that can be taken by Jonathan modulo 1 000 200 013.

Примеры

stdin	stdout
1 1	1
2 2	137

Задача C. Fans Placement

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	256 megabytes

The fan sector of the stadium consists of N rows of seats placed one behind another on the stairs. To prevent crowds during football games, the stadium administration set up only one seat in each row.

It is known that K fans are going to visit the next football game. For each fan i, you are given his height h_i. Each fan will cheer for his team and therefore will stand still all game long. All fans obey the rules, so they will not move from their places during the game.

A nice ticket seller Billy decided to distribute tickets to fan sector of the stadium in such way that each fan will comfortably see the game. The requirements for such setup (that is also called a good setup) are below.

Assume that the rows on the stadium are numbered from the bottom to the top of the stadium from 1 to N. A fan assigned to the row i and whose height is h_1 will not block the view for a fan assigned to the row j and whose height is h_2 if at least one of the two conditions is true:

1. $i > j$;
2. $i + h_1 < j + h_2$.

The job of a ticket seller is really boring, so Billy decided to find out how many good setups exist for the given set of fans. Surely, it is impossible to assign one seat to multiple fans or one fan to multiple seats. Unfortunately, Billy doesn't have access to a computer right now, so he asked you to calculate the number of good setups modulo 1 000 200 013. Two setups are different if at least one fan stands in a different place.

Формат входного файла

The first line of the input contains two integers N and K ($1 \le N \le 1000$, $1 \le K \le 10$, $K \le N$) separated by space: the number of rows and the number of fans, respectively. The next line holds K integers: i-th number describes the height h_i ($1 \le h_i \le 1000$) of the fan number i.

Формат выходного файла

Output a single integer: the number of good setups for the given number of rows and given fans modulo 1 000 200 013.

Примеры

stdin	stdout
3 2 1 2	4
5 1 4	5
5 2 4 2	13

Задача D. Interesting Language

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	256 megabytes

While studying the ancient Byteland language, the Byteland scientists faced a specific problem. That language used only the first 12 Latin letters for its alphabet. As a result, many pairs of words in the language were similar to each other. One critical problem arose.

We assume that all words in the ancient Byteland language are enumerated by positive integers starting from 1. Word number i is S_i, its length is L_i. You need to calculate the number of quadruples of indices (i, j, a, b) where

- $i < j$,
- $L_a > L_i$,
- $L_b > L_j$,
- prefix of S_a with length L_i is S_i,
- prefix of S_b with length L_j is S_j,
- after removal of first L_i and L_j characters from words S_a and S_b respectively, the remaining two words will be equal.

Refer to the example for clarification.

Help the scientists to calculate this value.

Формат входного файла

The first line of input contains a single positive integer N: the number of words in Byteland language. Next N lines consist of one word each. Each word consists of a positive number of small Latin letters from "`a`" to "`l`", inclusive. All words are different. The sum of lengths of all words does not exceed 10^6.

Формат выходного файла

Output single integer: the number of index quadruples satisfying the problem statement.

Примеры

stdin	stdout
7 a b ab bb ac bc abc	4
8 d f dd ff ddd fff ddfff fffff	4

Note

In first example, the following quadruples are satisfying: $(1, 2, 3, 4)$, $(1, 2, 5, 6)$, $(1, 3, 5, 7)$, $(2, 3, 6, 7)$.

In second example, the following quadruples are satisfying: $(1, 3, 3, 5)$, $(2, 4, 4, 6)$, $(3, 4, 7, 8)$, $(2, 6, 6, 8)$.

Задача E. Road Problem

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	`4 seconds`
Ограничение по памяти:	256 megabytes

Road system of Byteland is in recession: roads are now one-directional, there can be multiple roads between two cities or even none.

To fix the problem, Byteland parliament pronounced the law that each traveler should pay tax each time he enters a city. Each city has its own predefined tax.

Merchant guilds were very disappointed by the new law, and the king decided to ease the burden: each traveler arriving at city X can do it for free if he or she already paid the tax for the city Y that can be reached **from** city X directly or through other cities, including ones not yet visited by traveler. Nevertheless, a traveler is allowed to pay the tax if he or she wants to. The tax can be paid only at the time a traveler enters a city.

The famous merchant Bit Bytes wants to get from Byteland capital M to one of the biggest northern cities S. Obviously, Bit wants to spend as little money as possible on the taxes for the trip. Note that Bit starts at city M, so he doesn't have to, and cannot, pay the tax there before the start of his trip.

Using the information about Byteland road system and taxes, construct the optimal route for Bit. It is guaranteed that such route exists.

Формат входного файла

The first line of input holds two integers n and m ($2 \le n \le 100\,000$, $1 \le m \le 500\,000$), the number if cities and roads in Byteland. The second line of input holds n integers C_i ($0 \le C_i \le 10\,000$): taxes for each city. Next m lines hold two integers each: x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) which mean that i-th road starts in city x_i and ends in city y_i. Multiple connections between two cities are possible. For convenience, the number of capital city M is 1, and the number of destination city S is n. It is guaranteed that city S is reachable from city M. All integers in a line are separated by single spaces.

Формат выходного файла

Output a single integer: the minimal total payment necessary to reach S from M.

Примеры

stdin	stdout
3 2 1 1 1 1 3 1 3	1
4 4 1 2 3 4 1 2 2 3 3 2 3 4	6
2 2 1 2 1 2 2 1	2

Задача F. Place for Capital

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

During one of his campaigns, an old knight Yury Long-Legged found a huge planar field with an interesting anomaly. When standing on the field, it was only possible to move along specific trajectories: either along the lines that cross a fixed point K or along concentric circles centered at the same point. Moving along any other trajectory was impossible.

The knight was so impressed by the place that he decided to build a city on the field. Point K was pronounced the center of the knight's lands and the starting point of all roads. But then the knight wondered what minimal distance he will have to travel while walking from point A to point B on the field.

While moving, one can drop the current trajectory at any moment and pick another arc or radial line passing through one's current location. The distance traveled is the sum of the distances traveled along all radial and circular parts of the route.

Формат входного файла

The first line of the input contains four integers — coordinates of point A (x_A and y_A) followed by coordinates of point B (x_B and y_B). Point K is located at the origin. All coordinates don't exceed 10^6 by absolute value.

Формат выходного файла

Output a single number: the minimal distance that knight will have to travel on the way from A to B. The answer is considered correct if absolute or relative error is no more than 10^{-6}.

Примеры

stdin	stdout
0 5 4 3	4.636476090008
0 5 4 -3	10.000000000000