A1. Death Stars (easy)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The stardate is 1977 and the science and art of detecting Death Stars is in its infancy. Princess Heidi has received information about the stars in the nearby solar system from the Rebel spies and now, to help her identify the exact location of the Death Star, she needs to know whether this information is correct.

Two rebel spies have provided her with the maps of the solar system. Each map is an $N \times N$ grid, where each cell is either occupied by a star or empty. To see whether the information is correct, Heidi needs to know whether the two maps are of the same solar system, or if possibly one of the spies is actually an Empire double agent, feeding her false information.

Unfortunately, spies may have accidentally rotated a map by 90, 180, or 270 degrees, or flipped it along the vertical or the horizontal axis, before delivering it to Heidi. If Heidi can rotate or flip the maps so that two of them become identical, then those maps are of the same solar system. Otherwise, there are traitors in the Rebel ranks! Help Heidi find out.

Input

The first line of the input contains one number N $(1 \leq N \leq 10)$ – the dimension of each map. Next N lines each contain N characters, depicting the first map: 'X' indicates a star, while 'O' indicates an empty quadrant of space. Next N lines each contain N characters, depicting the second map in the same format.

Output

The only line of output should contain the word `Yes` if the maps are identical, or `No` if it is impossible to match them by performing rotations and translations.

Examples

input

```
4
XOOO
XXOO
OOOO
XXXX
XOOO
XOOO
XOXO
XOXX
```

output

```
Yes
```

input

```
2
XX
OO
XO
OX
```

output

```
No
```

Note

In the first test, you can match the first map to the second map by first flipping the first map along the vertical axis, and then by rotating it 90 degrees clockwise.

A2. Death Stars (medium)

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The stardate is 1983, and Princess Heidi is getting better at detecting the Death Stars. This time, two Rebel spies have yet again given Heidi two maps with the possible locations of the Death Star. Since she got rid of all double agents last time, she knows that both maps are correct, and indeed show the map of the solar system that contains the Death Star. However, this time the Empire has hidden the Death Star very well, and Heidi needs to find a place that appears on both maps in order to detect the Death Star.

The first map is an $N \times M$ grid, each cell of which shows some type of cosmic object that is present in the corresponding quadrant of space. The second map is an $M \times N$ grid. Heidi needs to align those two maps in such a way that they overlap over some $M \times M$ section in which all cosmic objects are identical. Help Heidi by identifying where such an $M \times M$ section lies within both maps.

Input

The first line of the input contains two space-separated integers N and M ($1 \leq N \leq 2000$, $1 \leq M \leq 200$, $M \leq N$). The next N lines each contain M lower-case Latin characters (a-z), denoting the first map. Different characters correspond to different cosmic object types. The next M lines each contain N characters, describing the second map in the same format.

Output

The only line of the output should contain two space-separated integers i and j, denoting that the section of size $M \times M$ in the first map that starts at the i-th row is equal to the section of the second map that starts at the j-th column. Rows and columns are numbered starting from 1.

If there are several possible ways to align the maps, Heidi will be satisfied with any of those. It is guaranteed that a solution exists.

Example

input

```
10 5
somer
andom
noise
mayth
eforc
ebewi
thyou
hctwo
again
noise
somermayth
andomeforc
noiseebewi
againthyou
noisehctwo
```

output

```
4 6
```

Note

The 5-by-5 grid for the first test case looks like this:

```
mayth
eforc
ebewi
thyou
hctwo
```

A3. Death Stars (hard)

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The stardate is 2015, and Death Stars are bigger than ever! This time, two rebel spies have yet again given Heidi two maps with the possible locations of the Death Stars.

Heidi has now received two maps with possible locations of N Death Stars. She knows that each of the maps is possibly corrupted, and may contain some stars that are not Death Stars. Furthermore, each of the maps was created from a different point of view. Hence, stars that are shown in one of the maps are rotated and translated with respect to the other map. Now Heidi wants to find out which of the stars shown in both maps are actually Death Stars, and the correspondence between the Death Stars on the two maps.

Input

The first line of the input contains an integer N ($1000 \leq N \leq 50000$) – the number of Death Stars. The second line of the input contains an integer N_1 ($N \leq N_1 \leq 1.5 \cdot N$) – the number of stars in the first map. The next N_1 lines specify the coordinates of the stars in the first map. The i-th line contains two space-separated floating-point numbers x_i and y_i with two decimal digits of precision each, representing the coordinates of the i-th star in the first map.

The next line of the input contains an integer N_2 ($N \leq N_2 \leq 1.5 \cdot N$) – the number of stars in the second map. The next N_2 lines contain locations of the stars in the second map, given in the same format as for the first map.

Output

You should output exactly N lines, each containing a space-separated pair of integers i_1 and i_2. Each such line should indicate that the star numbered i_1 in the first map corresponds to the star numbered i_2 in the second map. Your answer will be considered correct if over 90% of the distinct pairs listed in your output are indeed correct.

Note

The tests are generated in the following way:

- The number of Death Stars N is pre-selected in some way.
- The numbers of stars on the first and on the second map, N_1 and N_2, are selected uniformly at random between $1.0 \times N$ and $1.5 \times N$.
- N Death Stars are generated at random, with coordinates between -10000 and 10000.
- Additional $N_1 - N$ and $N2 - N$ stars for the first and for the second map respectively are generated in the same way.
- A translation vector (dx, dy) is generated, with dx and dy selected uniformly at random between -10000 and 10000. Each point in the first map is translated by (dx, dy).
- A rotation angle θ is generated, with θ selected uniformly at random between 0 and 2π. Each point in the first map is rotated by an angle of θ around the origin.
- Translations and rotations for the second map are generated and applied in the same way.
- The order of points is randomly permuted for both maps.
- The test case is saved, with each point written with two decimal digits of precision.

B1. Maximum Control (easy)

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Resistance is trying to take control over all planets in a particular solar system. This solar system is shaped like a tree. More precisely, some planets are connected by bidirectional hyperspace tunnels in such a way that there is a path between every pair of the planets, but removing any tunnel would disconnect some of them.

The Resistance already has measures in place that will, when the time is right, enable them to control every planet that is not *remote*. A planet is considered to be remote if it is connected to the rest of the planets only via a single hyperspace tunnel.

How much work is there left to be done: that is, how many remote planets are there?

Input

The first line of the input contains an integer N $(2 \le N \le 1000)$ – the number of planets in the galaxy.

The next $N - 1$ lines describe the hyperspace tunnels between the planets. Each of the $N - 1$ lines contains two space-separated integers u and v $(1 \le u, v \le N)$ indicating that there is a bidirectional hyperspace tunnel between the planets u and v. It is guaranteed that every two planets are connected by a path of tunnels, and that each tunnel connects a different pair of planets.

Output

A single integer denoting the number of remote planets.

Examples

input

```
5
4 1
4 2
1 3
1 5
```

output

```
3
```

input

```
4
1 2
4 3
1 4
```

output

```
2
```

Note

In the first example, only planets 2, 3 and 5 are connected by a single tunnel.

In the second example, the *remote* planets are 2 and 3.

Note that this problem has only two versions – easy and medium.

B2. Maximum Control (medium)

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input

output: standard output

The Resistance is trying to take control over as many planets of a particular solar system as possible. Princess Heidi is in charge of the fleet, and she must send ships to some planets in order to maximize the number of controlled planets.

The Galaxy contains N planets, connected by bidirectional hyperspace tunnels in such a way that there is a unique path between every pair of the planets.

A planet is controlled by the Resistance if there is a Resistance ship in its orbit, or if the planet lies on the shortest path between some two planets that have Resistance ships in their orbits.

Heidi has not yet made up her mind as to how many ships to use. Therefore, she is asking you to compute, for every $K = 1, 2, 3, \ldots, N$, the maximum number of planets that can be controlled with a fleet consisting of K ships.

Input

The first line of the input contains an integer N ($1 \le N \le 10^5$) – the number of planets in the galaxy.

The next $N - 1$ lines describe the hyperspace tunnels between the planets. Each of the $N - 1$ lines contains two space-separated integers u and v ($1 \le u, v \le N$) indicating that there is a bidirectional hyperspace tunnel between the planets u and v. It is guaranteed that every two planets are connected by a path of tunnels, and that each tunnel connects a different pair of planets.

Output

On a single line, print N space-separated integers. The K-th number should correspond to the maximum number of planets that can be controlled by the Resistance using a fleet of K ships.

Examples

input

```
3
1 2
2 3
```

output

```
1 3 3
```

input

```
4
1 2
3 2
4 2
```

output

```
1 3 4 4
```

Note

Consider the first example. If $K = 1$, then Heidi can only send one ship to some planet and control it. However, for $K \ge 2$, sending ships to planets 1 and 3 will allow the Resistance to control all planets.

C1. Encryption (easy)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rebel spy Heidi has just obtained the plans for the Death Star from the Empire and, now on her way to safety, she is trying to break the encryption of the plans (of course they are encrypted – the Empire may be evil, but it is not stupid!). The encryption has several levels of security, and here is how the first one looks.

Heidi is presented with a screen that shows her a sequence of integers A and a positive integer p. She knows that the encryption code is a single number S, which is defined as follows:

Define the score of X to be the sum of the elements of X modulo p.

Heidi is given a sequence A that consists of N integers, and also given an integer p. She needs to split A into 2 parts such that:

- Each part contains at least 1 element of A, and each part consists of contiguous elements of A.
- The two parts do not overlap.
- The total sum S of the scores of those two parts is maximized. This is the encryption code.

Output the sum S, which is the encryption code.

Input

The first line of the input contains two space-separated integer N and p ($2 \le N \le 100\ 000$, $2 \le p \le 10\ 000$) – the number of elements in A, and the modulo for computing scores, respectively.

The second line contains N space-separated integers which are the elements of A. Each integer is from the interval $[1, 1\ 000\ 000]$.

Output

Output the number S as described in the problem statement.

Examples

input

```
4 10
3 4 7 2
```

output

```
16
```

input

```
10 12
16 3 24 13 9 8 7 5 12 12
```

output

```
13
```

Note

In the first example, the score is maximized if the input sequence is split into two parts as $(3, 4)$, $(7, 2)$. It gives the total score of $(3 + 4 \mod 10) + (7 + 2 \mod 10) = 16$.

In the second example, the score is maximized if the first part consists of the first three elements, and the second part consists of the rest. Then, the score is $(16+3+24 \mod 12)+(13+9+8+7+5+12+12 \mod 12) = 7+6 = 13$.

C2. Encryption (medium)

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Heidi has now broken the first level of encryption of the Death Star plans, and is staring at the screen presenting her with the description of the next code she has to enter. It looks surprisingly similar to the first one – seems like the Empire engineers were quite lazy...

Heidi is once again given a sequence A, but now she is also given two integers k and p. She needs to find out what the encryption key S is.

Let X be a sequence of integers, and p a positive integer. We define the score of X to be the sum of the elements of X modulo p.

Heidi is given a sequence A that consists of N integers, and also given integers k and p. Her goal is to split A into k part such that:

- Each part contains at least 1 element of A, and each part consists of contiguous elements of A.
- No two parts overlap.
- The total sum S of the scores of those parts is maximized.

Output the sum S – the encryption code.

Input

The first line of the input contains three space-separated integer N, k and p ($k \le N \le 20\ 000$, $2 \le k \le 50$, $2 \le p \le 100$) – the number of elements in A, the number of parts A should be split into, and the modulo for computing scores, respectively.

The second line contains N space-separated integers that are the elements of A. Each integer is from the interval $[1, 1\ 000\ 000]$.

Output

Output the number S as described in the problem statement.

Examples

input

```
4 3 10
3 4 7 2
```

output

```
16
```

input

```
10 5 12
16 3 24 13 9 8 7 5 12 12
```

output

```
37
```

Note

In the first example, if the input sequence is split as $(3, 4)$, (7), (2), the total score would be $(3 + 4 \mod 10) + (7 \mod 10) + (2 \mod 10) = 7 + 7 + 7 + 2 = 16$. It is easy to see that this score is maximum.

In the second example, one possible way to obtain score 37 is to make the following split: $(16, 3, 24)$, $(13, 9)$, (8), (7), $(5, 12, 12)$.

C3. Encryption (hard)

time limit per test: 2.2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Heidi is now just one code away from breaking the encryption of the Death Star plans. The screen that should be presenting her with the description of the next code looks almost like the previous one, though who would have thought that the evil Empire engineers would fill this small screen with several million digits! It is just ridiculous to think that anyone would read them all...

Heidi is once again given a sequence A and two integers k and p. She needs to find out what the encryption key S is.

Let X be a sequence of integers, and p a positive integer. We define the score of X to be the sum of the elements of X modulo p.

Heidi is given a sequence A that consists of N integers, and also given integers k and p. Her goal is to split A into k parts such that:

- Each part contains at least 1 element of A, and each part consists of contiguous elements of A.
- No two parts overlap.
- The total sum S of the scores of those parts is **minimized** (not maximized!).

Output the sum S, which is the encryption code.

Input

The first line of the input contains three space-separated integers N, k and p ($k \le N \le 500\,000$, $2 \le k \le 100$, $2 \le p \le 100$) – the number of elements in A, the number of parts A should be split into, and the modulo for computing scores, respectively.

The second line contains N space-separated integers that are the elements of A. Each integer is from the interval $[1, 1\,000\,000]$.

Output

Output the number S as described in the problem statement.

Examples

input

```
4 3 10
3 4 7 2
```

output

```
6
```

input

```
10 5 12
16 3 24 13 9 8 7 5 12 12
```

output

```
13
```

Note

In the first example, if the input sequence is split as (3), $(4, 7)$, (2), the total score would be $(3 \mod 10) + (4 + 7 \mod 10) + (2 \mod 10) = 3 + 1 + 2 = 6$. It is easy to see that this score is the smallest possible.

In the second example, one possible way to obtain score 13 is to make the following split: $(16, 3)$, (24), (13), $(9, 8)$, $(7, 5, 12, 12)$.

D1. Hyperspace Jump (easy)

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Rebel fleet is on the run. It consists of m ships currently gathered around a single planet. Just a few seconds ago, the vastly more powerful Empire fleet has appeared in the same solar system, and the Rebels will need to escape into hyperspace. In order to spread the fleet, the captain of each ship has independently come up with the coordinate to which that ship will jump. In the obsolete navigation system used by the Rebels, this coordinate is given as the value of an arithmetic expression of the form $\frac{a+b}{c}$.

To plan the future of the resistance movement, Princess Heidi needs to know, for each ship, how many ships are going to end up at the same coordinate after the jump. You are her only hope!

Input

The first line of the input contains a single integer m ($1 \le m \le 200\ 000$) – the number of ships. The next m lines describe one jump coordinate each, given as an arithmetic expression. An expression has the form `(a+b)/c`. Namely, it consists of: an opening parenthesis `(`, a positive integer a of up to two decimal digits, a plus sign `+`, a positive integer b of up to two decimal digits, a closing parenthesis `)`, a slash `/`, and a positive integer c of up to two decimal digits.

Output

Print a single line consisting of m space-separated integers. The i-th integer should be equal to the number of ships whose coordinate is equal to that of the i-th ship (including the i-th ship itself).

Example

input

```
4
(99+98)/97
(26+4)/10
(12+33)/15
(5+1)/7
```

output

```
1 2 2 1
```

Note

In the sample testcase, the second and the third ship will both end up at the coordinate 3.

Note that this problem has only two versions – easy and hard.

D2. Hyperspace Jump (hard)

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

It is now 125 years later, but humanity is still on the run from a humanoid-cyborg race determined to destroy it. Or perhaps we are getting some stories mixed up here... In any case, the fleet is now smaller. However, in a recent upgrade, all the navigation systems have been outfitted with higher-dimensional, linear-algebraic jump processors.

Now, in order to make a jump, a ship's captain needs to specify a *subspace* of the d-dimensional space in which the events are taking place. She does so by providing a generating set of vectors for that subspace.

Princess Heidi has received such a set from the captain of each of m ships. Again, she would like to group up those ships whose hyperspace jump subspaces are equal. To do so, she wants to assign a group number between 1 and m to each of the ships, so that two ships have the same group number if and only if their corresponding subspaces are equal (even though they might be given using different sets of vectors).

Help Heidi!

Input

The first line of the input contains two space-separated integers m and d ($2 \le m \le 30\ 000$, $1 \le d \le 5$) – the number of ships and the dimension of the full underlying vector space, respectively. Next, the m subspaces are described, one after another. The i-th subspace, which corresponds to the i-th ship, is described as follows:

The first line contains one integer k_i ($1 \le k_i \le d$). Then k_i lines follow, the j-th of them describing the j-th vector sent by the i-th ship. Each of the j lines consists of d space-separated integers a_j, $j = 1, \ldots, d$, that describe the vector $(a_1, a_2, \ldots, a_d) \in \mathbb{R}^d$; it holds that $|a_j| \le 250$. The i-th subspace is the linear span of these k_i vectors.

Output

Output m space-separated integers $g_1, \ldots, g_m$, where $g_i \in \{1, \ldots, m\}$ denotes the group number assigned to the i-th ship. That is, for any $1 \le i < j \le m$, the following should hold: $g_i = g_j$ if and only if the i-th and the j-th subspaces are equal. In addition, the sequence $(g_1, g_2, \ldots, g_m)$ should be lexicographically minimal among all sequences with that property.

Example

input

```
8 2
1
5 0
1
0 1
1
0 1
2
0 6
0 1
2
0 1
1 0
```

```
2
-5 -5
4 3
2
1 1
0 1
2
1 0
1 0
```

output

```
1 2 2 2 3 3 3 1
```

Note

In the sample testcase, the first and the last subspace are equal, subspaces 2 to 4 are equal, and subspaces 5 to 7 are equal.

Recall that two subspaces, one given as the span of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and another given as the span of vectors $w_1, \ldots, w_k \in \mathbb{R}^d$, are equal if each vector v_i can be written as a linear combination of vectors $w_1, \ldots, w_k$ (that is, there exist coefficients $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ such that $v_i = \alpha_1 w_1 + \ldots + \alpha_k w_k$) and, similarly, each vector w_i can be written as a linear combination of vectors $v_1, \ldots, v_n$.

Recall that a sequence $(g_1, g_2, \ldots, g_m)$ is lexicographically smaller than a sequence $(h_1, h_2, \ldots, h_m)$ if there exists an index i, $1 \le i \le m$, such that $g_i < h_i$ and $g_j = h_j$ for all $j < i$.

E1. Guard Duty (easy)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Rebel fleet is afraid that the Empire might want to strike back again. Princess Heidi needs to know if it is possible to assign R Rebel spaceships to guard B bases so that every base has exactly one guardian and each spaceship has exactly one assigned base (in other words, the assignment is a perfect matching). Since she knows how reckless her pilots are, she wants to be sure that any two (straight) paths – from a base to its assigned spaceship – do not intersect in the galaxy plane (that is, in 2D), and so there is no risk of collision.

Input

The first line contains two space-separated integers $R, B (1 \le R, B \le 10)$. For $1 \le i \le R$, the $i + 1$-th line contains two space-separated integers x_i and y_i ($|x_i|, |y_i| \le 10000$) denoting the coordinates of the i-th Rebel spaceship. The following B lines have the same format, denoting the position of bases. It is guaranteed that no two points coincide and that no three points are on the same line.

Output

If it is possible to connect Rebel spaceships and bases so as satisfy the constraint, output `Yes`, otherwise output `No` (without quote).

Examples

input

```
3 3
0 0
2 0
3 1
-2 1
0 3
2 2
```

output

```
Yes
```

input

```
2 1
1 0
2 2
3 1
```

output

```
No
```

Note

For the first example, one possible way is to connect the Rebels and bases in order.

For the second example, there is no perfect matching between Rebels and bases.

E2. Guard Duty (medium)

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Princess Heidi decided to give orders to all her K Rebel ship commanders in person. Unfortunately, she is currently travelling through hyperspace, and will leave it only at N specific moments $t_1, t_2, \ldots, t_N$. The meetings with commanders must therefore start and stop at those times. Namely, each commander will board her ship at some time t_i and disembark at some later time t_j. Of course, Heidi needs to meet with all commanders, and no two meetings can be held during the same time. Two commanders cannot even meet at the beginnings/endings of the hyperspace jumps, because too many ships in one position could give out their coordinates to the enemy.

Your task is to find minimum time that Princess Heidi has to spend on meetings, with her schedule satisfying the conditions above.

Input

The first line contains two integers K, N ($2 \le 2K \le N \le 500000$, $K \le 5000$). The second line contains N distinct integers $t_1, t_2, \ldots, t_N$ ($1 \le t_i \le 10^9$) representing the times when Heidi leaves hyperspace.

Output

Output only one integer: the minimum time spent on meetings.

Examples

input

```
2 5
1 4 6 7 12
```

output

```
4
```

input

```
3 6
6 3 4 2 5 1
```

output

```
3
```

input

```
4 12
15 7 4 19 3 30 14 1 5 23 17 25
```

output

```
6
```

Note

In the first example, there are five valid schedules: $[1, 4], [6, 7]$ with total time 4, $[1, 4], [6, 12]$ with total time 9, $[1, 4], [7, 12]$ with total time 8, $[1, 6], [7, 12]$ with total time 10, and $[4, 6], [7, 12]$ with total time 7. So the answer is 4.

In the second example, there is only 1 valid schedule: $[1, 2], [3, 4], [5, 6]$.

For the third example, one possible schedule with total time 6 is: $[1, 3], [4, 5], [14, 15], [23, 25]$.

E3. Guard Duty (hard)

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Now that Heidi knows that she can assign Rebel spaceships to bases (recall the easy subtask), she is asking you: how *exactly* to do this? Now, given positions of N spaceships and N bases on a plane, your task is to connect spaceships and bases with line segments so that:

- The segments do not intersect.
- Such a connection forms a perfect matching.

Input

The first line contains an integer N ($1 \le n \le 10000$). For $1 \le i \le N$, the $i+1$-th line contains two integers x_i and y_i ($|x_i|, |y_i| \le 10000$) denoting the coordinates of the i-th spaceship. The following N lines have the same format, denoting the position of bases. It is guaranteed that no two points coincide and no three points are on the same line.

Output

The output should have N lines. The i-th line should contain an integer p_i, the index of the base to which the i-th spaceship is connected. The sequence $p_1, \ldots, p_N$ should form a permutation of $1, \ldots, N$.

It is guaranteed that a solution exists. If there are multiple solutions, you can output any one of them.

Example

input

```
4
6 6
```

```
5 1
2 4
4 0
5 4
1 2
2 1
3 5
```

output

```
4
1
2
3
```

F1. Lightsabers (easy)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is unrest in the Galactic Senate. Several thousand solar systems have declared their intentions to leave the Republic. Master Heidi needs to select the Jedi Knights who will go on peacekeeping missions throughout the galaxy. It is well-known that the success of any peacekeeping mission depends on the colors of the lightsabers of the Jedi who will go on that mission.

Heidi has n Jedi Knights standing in front of her, each one with a lightsaber of one of m possible colors. She knows that for the mission to be the most effective, she needs to select some contiguous interval of knights such that there are exactly k_1 knights with lightsabers of the first color, k_2 knights with lightsabers of the second color, $\dots$, k_m knights with lightsabers of the m-th color. Help her find out if this is possible.

Input

The first line of the input contains n $(1 \le n \le 100)$ and m $(1 \le m \le n)$. The second line contains n integers in the range $\{1, 2, \dots, m\}$ representing colors of the lightsabers of the subsequent Jedi Knights. The third line contains m integers $k_1, k_2, \dots, k_m$ (with $1 \le \sum_{i=1}^{m} k_i \le n$) – the desired counts of lightsabers of each color from 1 to m.

Output

Output `YES` if an interval with prescribed color counts exists, or output `NO` if there is none.

Example

input

```
5 2
1 1 2 2 1
1 2
```

output

```
YES
```

F2. Lightsabers (medium)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is unrest in the Galactic Senate. Several thousand solar systems have declared their intentions to leave the Republic. Master Heidi needs to select the Jedi Knights who will go on peacekeeping missions throughout the galaxy. It is well-known that the success of any peacekeeping mission depends on the colors of the lightsabers of the Jedi who will go on that mission.

Heidi has n Jedi Knights standing in front of her, each one with a lightsaber of one of m possible colors. She knows that for the mission to be the most effective, she needs to select some contiguous interval of knights such that there are exactly k_1 knights with lightsabers of the first color, k_2 knights with lightsabers of the second color, $\dots$, k_m knights with lightsabers of the m-th color.

However, since the last time, she has learned that it is not always possible to select such an interval. Therefore, she decided to ask some Jedi Knights to go on an indefinite unpaid vacation leave near certain pits on Tatooine, if you know what I mean. Help Heidi decide what is the minimum number of Jedi Knights that need to be let go before she is able to select the desired interval from the subsequence of remaining knights.

Input

The first line of the input contains n $(1 \le n \le 2 \cdot 10^5)$ and m $(1 \le m \le n)$. The second line contains n integers in the range $\{1, 2, \dots, m\}$ representing colors of the lightsabers of the subsequent Jedi Knights. The third line contains m integers $k_1, k_2, \dots, k_m$ (with $1 \le \sum_{i=1}^{m} k_i \le n$) – the desired counts of Jedi Knights with lightsabers of each color from 1 to m.

Output

Output one number: the minimum number of Jedi Knights that need to be removed from the sequence so that, in what remains, there is an interval with the prescribed counts of lightsaber colors. If this is not possible, output -1.

Example

input
8 3 3 3 1 2 2 1 1 3 3 1 1
output
1

F3. Lightsabers (hard)

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There used to be unrest in the Galactic Senate. Several thousand solar systems had declared their intentions to leave the Republic. But fear not! Master Heidi was able to successfully select the Jedi Knights that have restored peace in the galaxy. However, she knows that evil never sleeps and a time may come when she will need to pick another group of Jedi Knights. She wants to be sure she has enough options to do so.

There are n Jedi Knights, each of them with a lightsaber of one of m colors. Given a number k, compute the number of differently colored collections of k lightsabers that some k Jedi Knights might have. Jedi Knights with lightsabers of the same color are indistinguishable (it's not the person, it's the lightsaber color that matters!), and their order does not matter; that is, we consider two collections of Jedi Knights to be different if and only if their vectors of counts of lightsabers of each color (like what you were given in the easy and the medium versions) are different. We count all subsets, not only contiguous subsegments of the input sequence. Output the answer modulo 1009.

Input

The first line of the input contains n $(1 \le n \le 2 \cdot 10^5)$, m $(1 \le m \le n)$ and k $(1 \le k \le n)$. The second line contains n integers in the range $\{1, 2, \ldots, m\}$ representing colors of the lightsabers of subsequent Jedi Knights.

Output

Output one number: the number of differently colored collections of k lightsabers modulo 1009.

Example

input
4 3 2 1 2 3 2
output
4