A. A Student's Dream

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

Statistics claims that students sleep no more than three hours a day. But even in the world of their dreams, while they are snoring peacefully, the sense of impending doom is still upon them.

A poor student is dreaming that he is sitting the mathematical analysis exam. And he is examined by the most formidable professor of all times, a three times Soviet Union Hero, a Noble Prize laureate in student expulsion, venerable Petr Palych.

The poor student couldn't answer a single question. Thus, instead of a large spacious office he is going to apply for a job to thorium mines. But wait a minute! Petr Palych decided to give the student the last chance! Yes, that is possible only in dreams.

So the professor began: "Once a Venusian girl and a Marsian boy met on the Earth and decided to take a walk holding hands. But the problem is the girl has a_l fingers on her left hand and a_r fingers on the right one. The boy correspondingly has b_l and b_r fingers. They can only feel comfortable when holding hands, when no pair of the girl's fingers will touch each other. That is, they are comfortable when between any two girl's fingers there is a boy's finger. **And in addition, no three fingers of the boy should touch each other.** Determine if they can hold hands so that the both were comfortable."

The boy any the girl don't care who goes to the left and who goes to the right. The difference is only that if the boy goes to the left of the girl, he will take her left hand with his right one, and if he goes to the right of the girl, then it is vice versa.

Input

The first line contains two positive integers not exceeding 100. They are the number of fingers on the Venusian girl's left and right hand correspondingly. The second line contains two integers not exceeding 100. They are the number of fingers on the Marsian boy's left and right hands correspondingly.

Output

Print `YES` or `NO`, that is, the answer to Petr Palych's question.

Sample test(s)

input
```
5 1
10 5
```
output
```
YES
```

input
```
4 5
3 3
```
output
```
YES
```

input
```
1 2
11 6
```
output
```
NO
```

Note

The boy and the girl don't really care who goes to the left.

B. Tyndex.Brome

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

Tyndex is again well ahead of the rivals! The reaction to the release of Zoozle Chrome browser was the release of a new browser Tyndex.Brome!

The popularity of the new browser is growing daily. And the secret is not even the Tyndex.Bar installed (the Tyndex.Bar automatically fills the glass with the finest 1664 cognac after you buy Tyndex.Bottles and insert in into a USB port). It is highly popular due to the well-thought interaction with the user.

Let us take, for example, the system of automatic address correction. Have you entered `codehorses` instead of `codeforces`? The gloomy Zoozle Chrome will sadly say that the address does not exist. Tyndex.Brome at the same time will automatically find the closest address and sent you there. That's brilliant!

How does this splendid function work? That's simple! For each potential address a function of the F error is calculated by the following rules:

- for every letter c_i from the potential address c the closest position j of the letter c_i in the address (s) entered by the user is found. The absolute difference $|i - j|$ of these positions is added to F. So for every i ($1 \le i \le |c|$) the position j is chosen such, that $c_i = s_j$, and $|i - j|$ is minimal possible.
- if no such letter c_i exists in the address entered by the user, then the length of the potential address $|c|$ is added to F.

After the values of the error function have been calculated for all the potential addresses the most suitable one is found.

To understand the special features of the above described method better, it is recommended to realize the algorithm of calculating the F function for an address given by the user and some set of potential addresses. Good luck!

Input

The first line contains two integers n and k ($1 \le n \le 10^5$, $1 \le k \le 10^5$). They are the number of potential addresses and the length of the address entered by the user. The next line contains k lowercase Latin letters. They are the address entered by the user (s). Each next i-th ($1 \le i \le n$) line contains a non-empty sequence of lowercase Latin letters. They are the potential address. It is guaranteed that the total length of all the lines does not exceed $2 \cdot 10^5$.

Output

On each n line of the output file print a single number: the value of the error function when the current potential address is chosen.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Sample test(s)

input

```
2 10
codeforces
codeforces
codehorses
```

output

```
0
12
```

input

```
9 9
vkontakte
vcontacte
vkontrakte
vkollapse
vkrokodile
vtopke
vkapuste
vpechke
vk
vcodeforcese
```

output

```
18
14
36
47
14
29
30
0
84
```

C. Inquisition

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

In Medieval times existed the tradition of burning witches at steaks together with their pets, black cats. By the end of the 15-th century the population of black cats ceased to exist. The difficulty of the situation led to creating the EIC - the Emergency Inquisitory Commission.

The resolution #666 says that a white cat is considered black when and only when the perimeter of its black spots exceeds the acceptable norm. But what does the acceptable norm equal to? Every inquisitor will choose it himself depending on the situation. And your task is to find the perimeter of black spots on the cat's fur.

The very same resolution says that the cat's fur is a white square with the length of 10^5. During the measurement of spots it is customary to put the lower left corner of the fur into the origin of axes $(0;0)$ and the upper right one — to the point with coordinates $(10^5;10^5)$. The cats' spots are nondegenerate triangles. The spots can intersect and overlap with each other, but it is guaranteed that each pair of the triangular spots' sides have no more than one common point.

We'll regard the perimeter in this problem as the total length of the boarders where a cat's fur changes color.

Input

The first input line contains a single integer n $(0 \le n \le 100)$. It is the number of spots on the cat's fur. The i-th of the last n lines contains 6 integers: $x_{1i}, y_{1i}, x_{2i}, y_{2i}, x_{3i}, y_{3i}$. They are the coordinates of the i-th triangular spot $(0 < x_{ji}, y_{ji} < 10^5)$.

Output

Print a single number, the answer to the problem, perimeter of the union of triangles. Your answer should differ from the correct one in no more than 10^{-6}.

Sample test(s)

input

```
1
1 1 2 1 1 2
```

output

```
3.4142135624
```

input

```
3
3 3 10 3 3 10
1 1 9 4 5 6
2 2 11 7 6 11
```

output

```
37.7044021497
```

D. Wormhouse

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

Arnie the Worm has finished eating an apple house yet again and decided to move. He made up his mind on the plan, the way the rooms are located and how they are joined by corridors. He numbered all the rooms from 1 to n. All the corridors are bidirectional.

Arnie wants the new house to look just like the previous one. That is, it should have exactly n rooms and, if a corridor from room i to room j existed in the old house, it should be built in the new one.

We know that during the house constructing process Arnie starts to eat an apple starting from some room and only stops when he eats his way through all the corridors and returns to the starting room. It is also known that Arnie eats without stopping. That is, until Arnie finishes constructing the house, he is busy every moment of his time gnawing a new corridor. Arnie doesn't move along the already built corridors.

However, gnawing out corridors in one and the same order any time you change a house is a very difficult activity. That's why Arnie, knowing the order in which the corridors were located in the previous house, wants to gnaw corridors in another order. It is represented as a list of rooms in the order in which they should be visited. The new list should be lexicographically smallest, but it also should be strictly lexicographically greater than the previous one. Help the worm.

Input

The first line contains two integers n and m ($3 \le n \le 100, 3 \le m \le 2000$). It is the number of rooms and corridors in Arnie's house correspondingly. The next line contains $m + 1$ positive integers that do not exceed n. They are the description of Arnie's old path represented as a list of rooms he visited during the gnawing. It is guaranteed that the last number in the list coincides with the first one.

The first room described in the list is the main entrance, that's why Arnie should begin gnawing from it.

You may assume that there is no room which is connected to itself and there is at most one corridor between any pair of rooms. However, it is possible to find some isolated rooms which are disconnected from others.

Output

Print $m + 1$ positive integers that do not exceed n. Those numbers are the description of the new path, according to which Arnie should gnaw out his new house. If it is impossible to find new path you should print out `No solution`. The first number in your answer should be equal to the last one. Also it should be equal to the main entrance.

Sample test(s)

input

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

output

```
1 3 2 1
```

input

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

output

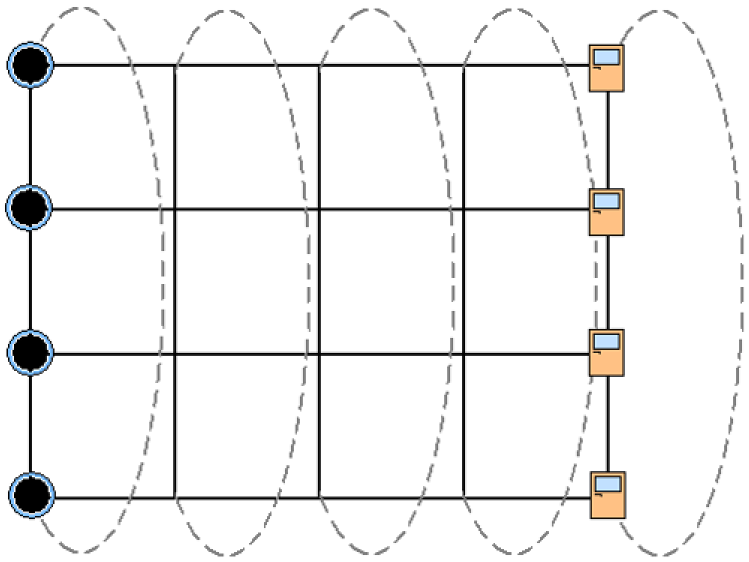
```
No solution
```

E. World Evil

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

As a result of Pinky and Brain's mysterious experiments in the Large Hadron Collider some portals or black holes opened to the parallel dimension. And the World Evil has crept to the veil between their world and ours. Brain quickly evaluated the situation and he understood that the more evil tentacles creep out and become free, the higher is the possibility that Brain will rule the world.

The collider's constriction is a rectangular grid rolled into a cylinder and consisting of n rows and m columns such as is shown in the picture below:



In this example $n = 4$, $m = 5$. Dotted lines are corridores that close each column to a ring, i. e. connect the n-th and the 1-th rows of the grid.

In the leftmost column of the grid the portals are situated and the tentacles of the World Evil are ready to creep out from there. In the rightmost column the exit doors are located. The tentacles can only get out through those doors. The segments joining the nodes of the grid are corridors.

Brain would be glad to let all the tentacles out but he faces a problem: the infinite number of tentacles can creep out of the portals, every tentacle possesses infinite length and some width and the volume of the corridors are, unfortunately, quite limited. Brain could approximately evaluate the maximal number of tentacles that will be able to crawl through every corridor.

Now help the mice to determine the maximal number of tentacles of the World Evil that will crawl out of the Large Hadron Collider.

Input

The first line of the input file contains two integers n and m ($2 \le n \le 5$, $2 \le m \le 10^5$). They are the sizes of the Large Hadron Collider grid. The next m - 1 lines contain n numbers each. They are the horizontal corridors' capacities. The next m lines contain n numbers each. They are the vertical corridors' capacities. Corridors are described from left to right and from top to bottom. Every n-th vertical corridor connects nodes of the n-th and 1-th rows. A corridor's capacity is a non-negative integer that does not exceed 10^9.

Output

Print a single number, the number of the World Evil tentacles Pinky and Brain will command.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cout` (also you may use `%I64d`).

Sample test(s)

input

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

output

```
7
```

input

```
2 2
9 2
2 3
6 1
```

output

```
11
```