A. Rock-Paper-Scissors

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

Nikephoros and Polycarpus play rock-paper-scissors. The loser gets pinched (not too severely!).

Let us remind you the rules of this game. Rock-paper-scissors is played by two players. In each round the players choose one of three items independently from each other. They show the items with their hands: a rock, scissors or paper. The winner is determined by the following rules: the rock beats the scissors, the scissors beat the paper and the paper beats the rock. If the players choose the same item, the round finishes with a draw.

Nikephoros and Polycarpus have played n rounds. In each round the winner gave the loser a friendly pinch and the loser ended up with a fresh and new red spot on his body. If the round finished in a draw, the players did nothing and just played on.

Nikephoros turned out to have worked out the following strategy: before the game began, he chose some sequence of items $A = (a_1, a_2, ..., a_m)$, and then he cyclically showed the items from this sequence, starting from the first one. Cyclically means that Nikephoros shows signs in the following order: $a_1, a_2, ..., a_m, a_1, a_2, ..., a_m, a_1, ...$ and so on. Polycarpus had a similar strategy, only he had his own sequence of items $B = (b_1, b_2, ..., b_k)$.

Determine the number of red spots on both players after they've played n rounds of the game. You can consider that when the game began, the boys had no red spots on them.

Input

The first line contains integer n $(1 \le n \le 2 \cdot 10^9)$ — the number of the game's rounds.

The second line contains sequence A as a string of m characters and the third line contains sequence B as a string of k characters $(1 \le m, k \le 1000)$. The given lines only contain characters "R", "S" and "P". Character "R" stands for the rock, character "S" represents the scissors and "P" represents the paper.

Output

Print two space-separated integers: the numbers of red spots Nikephoros and Polycarpus have.

Sample test(s)

input

```
7
RPS
RSPP
```

output

```
3 2
```

input

```
5
RRRRRRRR
R
```

output

```
0 0
```

Note

In the first sample the game went like this:

- R - R. Draw.
- P - S. Nikephoros loses.
- S - P. Polycarpus loses.
- R - P. Nikephoros loses.
- P - R. Polycarpus loses.
- S - S. Draw.
- R - P. Nikephoros loses.

Thus, in total Nikephoros has 3 losses (and 3 red spots), and Polycarpus only has 2.

B. Chamber of Secrets

time limit per test: 2 seconds

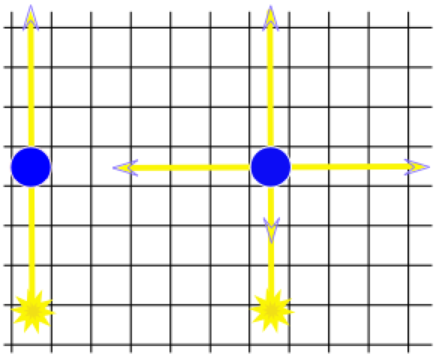
memory limit per test: 256 megabytes

input: standard input

output: standard output

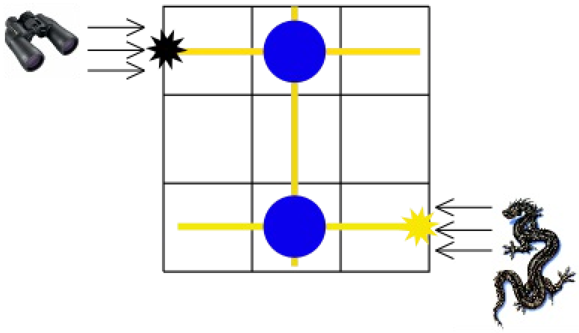
"The Chamber of Secrets has been opened again" — this news has spread all around Hogwarts and some of the students have been petrified due to seeing the basilisk. Dumbledore got fired and now Harry is trying to enter the Chamber of Secrets. These aren't good news for Lord Voldemort. The problem is, he doesn't want anybody to be able to enter the chamber. The Dark Lord is going to be busy sucking life out of Ginny.

The Chamber of Secrets is an $n \times m$ rectangular grid in which some of the cells are columns. A light ray (and a basilisk's gaze) passes through the columns without changing its direction. But with some spell we can make a column magic to reflect the light ray (or the gaze) in all four directions when it receives the ray. This is shown in the figure below.



The left light ray passes through a regular column, and the right ray — through the magic column.

The basilisk is located at the right side of the lower right cell of the grid and is looking to the left (in the direction of the lower left cell). According to the legend, anyone who meets a basilisk's gaze directly dies immediately. But if someone meets a basilisk's gaze through a column, this person will get petrified. We know that the door to the Chamber is located on the left side of the upper left corner of the grid and anyone who wants to enter will look in the direction of its movement (in the direction of the upper right cell) from that position.



This figure illustrates the first sample test.

Given the dimensions of the chamber and the location of regular columns, Lord Voldemort has asked you to find the minimum number of columns that we need to make magic so that anyone who wants to enter the chamber would be petrified or just declare that it's impossible to secure the chamber.

Input

The first line of the input contains two integer numbers n and m $(2 \le n, m \le 1000)$. Each of the next n lines contains m characters. Each character is either "`.`" or "`#`" and represents one cell of the Chamber grid. It's "`.`" if the corresponding cell is empty and "`#`" if it's a regular column.

Output

Print the minimum number of columns to make magic or -1 if it's impossible to do.

Sample test(s)

input

```
3 3
.#.
...
.#.
```

output

```
2
```

input

```
4 3
##.
...
.#.
.#.
```

output

```
2
```

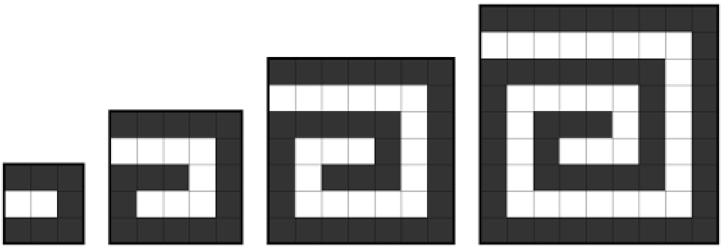
Note

The figure above shows the first sample test. In the first sample we should make both columns magic. The dragon figure represents the basilisk and the binoculars represent the person who will enter the Chamber of secrets. The black star shows the place where the person will be petrified. Yellow lines represent basilisk gaze moving through columns.

C. Spiral Maximum

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

Let's consider a $k \times k$ square, divided into unit squares. Please note that $k \geq 3$ and is odd. We'll paint squares starting from the upper left square in the following order: first we move to the right, then down, then to the left, then up, then to the right again and so on. We finish moving in some direction in one of two cases: either we've reached the square's border or the square following after the next square is already painted. We finish painting at the moment when we cannot move in any direction and paint a square. The figure that consists of the painted squares is a *spiral*.



The figure shows examples of spirals for $k = 3, 5, 7, 9$.

You have an $n \times m$ table, each of its cells contains a number. Let's consider all possible spirals, formed by the table cells. It means that we consider all spirals of any size that don't go beyond the borders of the table. Let's find the sum of the numbers of the cells that form the spiral. You have to find the maximum of those values among all spirals.

Input

The first line contains two integers n and m $(3 \leq n, m \leq 500)$ — the sizes of the table.

Each of the next n lines contains m space-separated integers: the j-th number in the i-th line a_{ij} ($-1000 \leq a_{ij} \leq 1000$) is the number recorded in the j-th cell of the i-th row of the table.

Output

Print a single number — the maximum sum of numbers among all spirals.

Sample test(s)

input

```
6 5
0 0 0 0 0
1 1 1 1 1
0 0 0 0 1
1 1 1 0 1
1 0 0 0 1
1 1 1 1 1
```

output

```
17
```

input

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

output

```
6
```

input

```
6 6
-3 2 0 1 5 -1
4 -1 2 -3 0 1
-5 1 2 4 1 -2
0 -2 1 3 -1 2
3 1 4 -3 -2 0
-1 2 -1 3 1 2
```

output

```
13
```

Note

In the first sample the spiral with maximum sum will cover all 1's of the table.

In the second sample the spiral may cover only six 1's.

D. Deputies

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Trinitarian kingdom has exactly $n = 3k$ cities. All of them are located on the shores of river Trissisipi, which flows through the whole kingdom. Some of the cities are located on one side of the river, and all the rest are on the other side.

Some cities are connected by bridges built between them. Each bridge connects two cities that are located on the opposite sides of the river. Between any two cities exists no more than one bridge.

The recently inaugurated King Tristan the Third is busy distributing his deputies among cities. In total there are k deputies and the king wants to commission each of them to control exactly three cities. However, no deputy can be entrusted to manage the cities, which are connected by a bridge — the deputy can set a too high fee for travelling over the bridge to benefit his pocket, which is bad for the reputation of the king.

Help King Tristan the Third distribute the deputies between the cities, if it is possible.

Input

The first line contains two integers n and m — the number of cities and bridges ($3 \le n < 10^5$, $n = 3k$, $0 \le m \le 10^5$). Next m lines describe the bridges. The i-th line contains two integers a_i and b_i — the numbers of cities that are connected by the i-th bridge ($1 \le a_i, b_i \le n$, $a_i \ne b_i$, $1 \le i \le m$).

It is guaranteed that no bridge connects a city with itself and that any two cities are connected with no more than one bridge.

Output

If distributing the deputies in the required manner is impossible, print in a single line "`NO`" (without the quotes).

Otherwise, in the first line print "`YES`" (without the quotes), and in the second line print which deputy should be put in charge of each city. The i-th number should represent the number of the deputy (from 1 to k), who should be in charge of city numbered i-th in the input — overall there should be n numbers.

If there are multiple solutions, print any of them.

Sample test(s)

input

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

output

```
YES
1 2 1 2 2 1
```

input

```
3 1
1 2
```

output

```
NO
```

E. Camping Groups

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

A club wants to take its members camping. In order to organize the event better the club directors decided to partition the members into several groups.

Club member i has a responsibility value r_i and an age value a_i. A *group* is a non-empty subset of club members with one member known as group leader. A group leader should be one of the most responsible members of the group (his responsibility value is not less than responsibility of any other group member) and his age absolute difference with any other group member should not exceed k.

Some club members are friends and want to be in the same group. They also like their group to be as large as possible. Now you should write a program that answers a series of questions like "What's the largest size of a group containing club member x and club member y?". It's possible for x or y to be the group leader.

Input

The first line contains two integers n and k $(2 \le n \le 10^5, 0 \le k \le 10^9)$ — the number of club members and the age restriction for one group.

The next line contains integer numbers $r_1, r_2, ..., r_n$ $(1 \le r_i \le 10^9)$ separated by space: r_i denotes the i-th club member's responsibility. In the same way there are integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ in the third line: a_i denotes the i-th club member's age.

The next line contains an integer q denoting the number of questions that you should answer $(1 \le q \le 10^5)$. The next q lines describe the questions. Each line contains two space-separated integers x_i and y_i $(1 \le x_i, y_i \le n, x_i \ne y_i)$ — the indices of the club members that should end up in the same group.

Output

For each question print the maximum size of the group in a line. If making such a group is impossible print -1 instead.

Sample test(s)

input

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

output

```
4
3
-1
4
```

Note

In the first query the largest group with members 3 and 5 is $\{1, 3, 4, 5\}$ where member 3 is the leader.

In the second query member 2 should be the leader so the group will be $\{1, 2, 3\}$.

In the third query the leader of the group should have age 3 so the only leader can be member 3, who is less responsible than member 2. So making a group is impossible.

The group for the fourth query is the same as first query.