TASK	PAPRIKA	NIKO	METEOR	KRATKI	NEO	WTF
input	standard input (*stdio*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	0.9 seconds	1 second
memory limit	32 MB	32 MB	64 MB	32 MB	32 MB	256 MB
score	**50**	**80**	**100**	**120**	**140**	**160**
	total 650					

Problems translated from Croatian by: **Paula Gombar**

Young Marin is working as an assistant chef in a popular restaurant called *Plavi 9* where the daily offer includes, among other things, stuffed peppers. Every assistant chef knows that the peppers are stuffed when they are **young**, so he decided to prepare the meal using only peppers not older than X days. Marin will serve all the other peppers **fresh**, as an appetizer. Luckily, as Croatia entered the EU, there is a new law enforced. The law states that every pepper has to have its ID card at any moment. Marin can easily determine the age of a pepper by looking at its ID card.

It is less known that peppers have, besides official documents, their own life purposes and ambitions. More precisely, every pepper knows from an early age whether they want to be served as a fresh or stuffed pepper when they grow up. Bearing that in mind, you are aware of the problems N peppers are facing while waiting in line to be filled. The life purpose of some peppers is to be a part of the dish, but they are too old, and some peppers want to be served fresh, but will be filled.

Because the peppers aren't aware of Marin's number X, they have decided to rectify the injustice using the following strategy. The first pepper tries to switch its ID card with second pepper, then the second pepper tries to switch its ID card with the third pepper and so on until the end of the line. Two peppers will switch their ID cards if the pepper that has a larger number on the ID card it's **currently holding** wants to become a filled pepper, and the one with the smaller number doesn't. The peppers will not switch ID cards if they have equal numbers on them.

Your task is to determine the number of peppers that will have their life purpose completed.

INPUT

The first line of input contains two integers N and X $(1 \leqslant N, X \leqslant 1\,000)$ from the task.

Each of the following N lines contains two integers a $(1 \leqslant a \leqslant 1\,000)$ and b $(0 \leqslant b \leqslant 1)$ that describe the peppers in the order which they're waiting in line to be filled.

The number a is printed on the ID card of a pepper and represents its age in days, and number b represents it's life purpose (0 if pepper wants to be served fresh or 1 if it wants to be served as a filled pepper).

OUTPUT

The first and only line of output must contain the number of peppers that will have their life purpose completed.

SAMPLE TESTS

<table>
<tr>
<td>

input

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

output

```
0
```

</td>
<td>

input

```
5 5
3 1
2 0
13 1
2 0
10 1
```

output

```
5
```

</td>
<td>

input

```
6 10
15 1
12 1
8 0
10 1
3 0
1 1
```

output

```
4
```

</td>
</tr>
</table>

Clarification of the first example: All peppers are young enough, but not a single one of them wants to be stuffed.

Clarification of the second example: Every two adjacent peppers switched their ID cards.

It is the year 2018, Russia. History repeats itself and the not-so-young football coach Niko is leading the national team in a clash with Brazil.

Choosing a formation is the first step when it comes to preparing the strategy for a football game. A formation can be represented with three integers O, V and N, where O denotes the number of defensive players, V denotes the number of midfielders and N denotes the number of offensive players. It is evident that it must hold $O + V + N = 10$.

After choosing a formation, the coach needs to carefully choose the players because not every player is a defensive wizard or a world class striker. The coach knows his players very well. He knows which line suits each player. Setting a player in a line he doesn't know how to play would be tactical suicide.

The coach and his team of expertise have put together a list of formations that can be taken into consideration against Brazil, but in all their expertise they didn't have time to determine the formations which they have suitable players for. Help them!

INPUT

The first line of input contains the integer N ($1 \leqslant N \leqslant 10$), the number of formations taken into consideration.

Each of the following N lines contains the formation given as O-V-N. The numbers O, V i N are positive integers and it holds $O + V + N = 10$.

The following line contains the integer M ($10 \leqslant M \leqslant 22$), the number of players who want to join the national team.

The i^{th} of the following M lines contains the list of lines that the i^{th} player can play in. Letter 'O' denotes defense, letter 'V' midfield positions and letter 'N' offense.

OUTPUT

Output exactly N lines. The i^{th} line of output should be "DA" if the coach has suitable players for the i^{th} formation from the input, or "NE" if he doesn't.

SAMPLE TESTS

<table>
<tr>
<td>

input

```
2
4-4-2
10-0-0
10
O
O
O
O
O
O
O
O
ON
NO
```

output

```
NE
DA
```

</td>
<td>

input

```
3
4-4-2
3-5-2
4-3-3
11
OV
OV
OVN
OV
OV
V
V
N
O
O
O
```

output

```
DA
DA
NE
```

</td>
</tr>
</table>

Clarification of the first example: We see that Niko has almost only defensive players at his service, so he can only use the formation 10-0-0, the infamous "parking the bus".

Clarification of the second example: For formation 4-4-2, he can put players 1, 2, 9 and 10 as defense, players 4, 5, 6 and 7 as midfielders, and players 3 and 8 as offense. For formation 3-5-2, he can put players 4, 9 and 10 as defense, players 1, 2, 5, 6 and 7 as midfielders, and players 3 and 8 as offense. Formation 4-3-3 can't be set up because the coach has only 2 offensive players.

A photograph of a small meteor of an unusual shape was posted on the Internet. In that photo, the meteor is falling from a great height towards an uneven ground. There was also a photograph taken just after the meteor fell, but it is sadly lost and needs to be reconstructed.

The photograph is simplified and represented as a matrix of characters. The character 'X' represents a part of the **meteor**, the character '#' represents a part of the **ground** and the rest of the image (**air**) consists of the characters '.'.

The meteor is connected. In other words, a path exists between each two parts of the meteor that passes only through the meteor and consists of steps up, down, left and right. Also, all parts of the ground are connected in the same way.

In the given photograph, the meteor is located strictly above ground. More precisely, there is at least one row of air (dots), the meteor is completely above it and the ground is completely below it. In addition, the entire bottom row of the image is a part of the ground.

The meteor was falling vertically downward. When it fell on the ground, **it kept its shape**, and the same goes for the ground. Reconstruct the photograph after the meteor fall!

INPUT

The first line of input contains the integers R and S ($3 \leqslant R, S \leqslant 3\,000$), the number of rows and the number of columns of the photograph.

The following R lines contain the photograph described in the task.

OUTPUT

Output the required photograph (dimensions $R \times S$) after the meteor fall.

SAMPLE TESTS

input
```
5 6
.XXXX.
...X..
......
#..###
######
```
output
```
......
.XXXX.
...X..
#..###
######
```

input
```
9 7
XXX.XXX
X.XXX.X
X..X..X
X.....X
.......
.#...#.
.##.##.
.#####.
#######
```
output
```
.......
.......
.......
.......
XXX.XXX
X#XXX#X
X##X##X
X#####X
#######
```

All of you are probably very familiar with the problem of finding the longest monotone subsequence. You probably think you know all about it. In order to convince us, solve the problem "opposite" to finding the longest monotone subsequence.

For given N and K, find the sequence that consists of numbers from 1 to N such that each of the numbers in it appears exactly once and the length of its longest monotone subsequence (ascending or descending) is exactly K.

INPUT

The first line of input contains the integers N and K ($1 \leqslant K \leqslant N \leqslant 10^6$), the length of the sequence and the required length of the longest monotone subsequence.

OUTPUT

If the required sequence doesn't exist, output -1 in the first and only line.
If the required sequence exists, output the required sequence of N numbers in the first and only line. Separate the numbers with a single space.
The required sequence (if it exists) is not necessarily unique, so you can output any valid sequence.

SAMPLE TESTS

input	input	input
4 3	5 1	5 5
output	**output**	**output**
1 4 2 3	-1	1 2 3 4 5

Clarification of the first sample test: A sequence of length 4 with longest monotone subsequence of length 3 is $(1, 4, 2, 3)$. The longest monotone sequence is $(1, 2, 3)$.

Let us denote $A_{i,j}$ as the element from matrix A located in the i^{th} row and j^{th} column. We say that the matrix A is *cool* if this holds:

- $r, s > 1$
- $A_{1,1} + A_{r,s} \leqslant A_{1,s} + A_{r,1}$

where r denotes the number of rows, and s the number of columns of matrix A.

Additionally, we say that a matrix is *extremely cool* if each of its submatrices with at least two rows and two columns is cool.

It is your task to determine the largest number of elements that are contained in an extremely cool submatrix of the given matrix.

INPUT

The first line of input contains two integers R, S ($2 \leqslant R, S \leqslant 1\,000$) which represent the dimensions of the matrix.

Each of the following R lines contains S integers that represent the elements in the matrix. The elements in the matrix will be integers from the interval $[-10^6, 10^6]$.

OUTPUT

The first and only line of output must contain the maximal number of elements that are contained in an extremely cool submatrix of the matrix from the input. If an extremely cool submatrix doesn't exist, output 0.

SCORING

In test cases worth 60% of total points, it will additionally hold $R, S \leqslant 350$.

SAMPLE TESTS

input	input	input
3 3 1 4 10 5 2 6 11 1 3	3 3 1 3 1 2 1 2 1 1 1	5 6 1 1 4 0 3 3 4 4 9 7 11 13 -3 -1 4 2 8 11 1 5 9 5 9 10 4 8 10 5 8 8
output	**output**	**output**
9	4	15

Clarification of the third example: The solution is a matrix with an upper left corner in (3,2) and lower right corner in (5,6).

Assume you are given an array **A** of N integers, array **ID** of $N+1$ integers from the interval $[1, N-1]$ and an integer R.

We are doing a Warshall-Turing-Fourier transformation[1] on array **A** in the following way:

```
sum = 0

for i = 1 to N
    index = min{ ID[i], ID[i+1] }
    sum = sum + A[index]
    rotate array A to the right by R places

change the signs of all elements in A

for i = 1 to N
    index = max{ ID[i], ID[i+1] }
    index = index + 1
    sum = sum + A[index]
    rotate array A to the right by R places
```

You are given the array **A** and constant R, but you are not familiar with the array **ID**. What is the largest possible value of variable `sum` after execution of the above algorithm?

INPUT

The first line of input contains the integers N and R ($2 \leqslant N \leqslant 3\,000$, $1 \leqslant R < N$) from the task.

The second line of input contains the elements of array **A**, respectively from $\mathbf{A}[1]$ to $\mathbf{A}[N]$. These are integers from the interval $[-10^4, 10^4]$.

OUTPUT

The first line of output must contain the maximal value of `sum`.
The second line of output must contain the array **ID** of $N+1$ integers from the interval [1, N - 1] for which the algorithm outputs the maximal sum. If there are multiple such arrays, output any.

If only the first line is correct (regardless of whether the second is printed), you will get 50% of points for the corresponding test case.

SCORING

In test cases worth 20% of total points, it will hold $N \leqslant 7$.
In test cases worth 60% of total points, it will hold $N \leqslant 300$.

SAMPLE TESTS

input	input
5 3 1 -1 1 -1 1	6 5 2 5 4 1 3 5
output	**output**
10 1 1 1 2 2 3	16 3 2 1 1 5 4 1

[1] This doesn't really exist.