TASK	**LJESTVICA**	**ARHIPELAG**	**TOTEM**	**HIPERCIJEVI**	**ROTIRAJ**	**MNOGOMET**

source code	`ljestvica.pas`
	`ljestvica.c`
	`ljestvica.cpp`
		`arhipelag.pas`
		`arhipelag.c`
		`arhipelag.cpp`
			`totem.pas`
			`totem.c`
			`totem.cpp`
				`hipercijevi.pas`
				`hipercijevi.c`
				`hipercijevi.cpp`
					`rotiraj.pas`
					`rotiraj.c`
					`rotiraj.cpp`
						`mnogomet.pas`
						`mnogomet.c`
						`mnogomet.cpp`

input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	1 second	1 second
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	128 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Problems translated from Croatian by: **Ivan Pilat**

Veronica attends a music academy. She was given a music sheet of a composition with only notes (without annotations), and needs to recognise the scale used. In this problem, we will limit ourselves to only the two most frequently used (and usually taught in schools first) scales: A-minor and C-major. This doesn't make them simpler or more basic than other minor and major scales – all minor scales are mutually equivalent save for translation, and so are major scales.

Still, out of the 12 tones of an octave {A, A#, B, C, C#, D, D#, E, F, F#, G, G#} used in modern music1, A-minor and C-major scales do use the tones with shortest names: A-minor is defined as an ordered septuple (A, B, C, D, E, F, G), and C-major as (C, D, E, F, G, A, B).

Notice that the sets of tones of these two scales are equal. What's the difference? The catch is that not only the set of tones, but also their usage, determines a scale. Specifically, the **tonic** (the first tone of a scale), **subdominant** (the fourth tone) and **dominant** (the fifth tone) are the primary candidates for accented tones in a composition. In A-minor, these are A, D, and E, and in C-major, they are C, F, and G. We will name these tones **main tones**.

Aren't the scales still equivalent save for translation? They are not: for example, the third tone of A-minor (C) is *three* half-tones higher than the tonic (A), while the third tone of C-major (E) is *four* half-tones higher than the tonic (C). The difference, therefore, lies in the intervals. This makes minor scales "sad" and major scales "happy".

Write a program to decide if a composition is more likely written in A-minor or C-major by counting whether there are more **main tones** of A-minor or of C-major among the **accented** tones (**the first tones in each measure**). If there is an **equal** number of main tones, determine the scale based on the **last tone** (which is guaranteed to be either A for A-minor or C for C-major in any such test case).

For example, examine the well-known melody "Frère Jacques"[2]:

CD|EC|CD|EC|EF|G|EF|G|GAGF|EC|GAGF|EC|CG|C|CG|C

The character "|" separates measures, so the accented tones are, in order: C, E, C, E, E, G, E, G, G, E, G, E, C, C, C, C. Ten of them (C, C, G, G, G, G, C, C, C, C) are main tones of C-major, while six (E, E, E, E, E, E) are main tones of A-minor. Therefore, our best estimate is that the song was written in C-major.

INPUT

The first and only line of input contains a sequence of at least 5, and at most 100, characters from the set {"A", "B", "C", "D", "E", "F", "G", "|"}. This is a simplified notation for a composition, where the character "|" separates measures. The characters "|" will never appear adjacent to one another, at the beginning, or at the end of the sequence.

OUTPUT

The first and only line of output must contain the text "C-dur" (for C-major) or "A-mol" (for A-minor).

---

1 This is the international, more consistent notation. In Croatia, the German notation is usually used, where A# (or Bb) is renamed to B, and B is renamed to H.

2 "Are You Sleeping?" in English; "Bratec Martin" in Croatian.

SAMPLE TESTS

input AEB|C **output** C-dur	**input** CD|EC|CD|EC|EF|G|EF|G|GAGF|EC|GAGF|EC|CG|C|CG|C **output** C-dur

A popular tourist destination country is situated on a breathtakingly beautiful archipelago constantly bathed by the sun. The country's residents are very proud of their numerous islands. However, global warming has them very worried: raising sea levels are resulting in rapidly increasing loss of dry land, which is diminishing the beauty of the archipelago.

The map of the archipelago is represented by a grid of **R** by **C** squares (characters). The character 'X' (**uppercase** letter x) represents dry land, while '.' (period) represents sea.

It has been estimated that, in fifty years, sea will have flooded every square of land that is currently surrounded by sea on **three** or on all **four** sides (north, south, east, west). Assume that all squares outside the map (along the edges) are covered by sea.

Your task is computing the map of the archipelago in fifty years (after the described sea level rise). Since there will probably be less land than today, you shouldn't print out the whole map, but only its **smallest rectangular part that contains all land squares**. It is guaranteed that at least one square of land will remain in all test cases.

INPUT

The first line of input contains two positive integers, **R** and **C** ($1 \le$ **R**, **C** ≤ 10), the dimensions of the current map.

Each of the following **R** lines contains **C** characters. These **R** by **C** characters represent the current map of the archipelago.

OUTPUT

The output must contain an appropriate number of lines representing the required rectangular part of the future (flooded) map.

SAMPLE TESTS

input

```
5 3
...
.X.
.X.
.X.
...
```

output

```
X
```

input

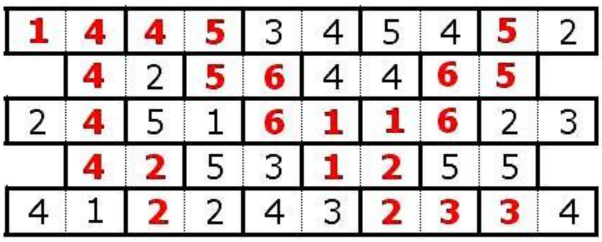
```
3 10
..........
..XXX.XXX.
XXX.......
```

output

```
.XX...X
XX.....
```

Mister No (real name Jerry Drake) is a comic book character who frequently gets himself into a lot of trouble, which he is usually able to get out of. However, this time it's not so easy. He was searching for ancient Mayan treasures and, in the process, stumbled upon a lost temple. Inside the temple there is a large hall, and inside the hall stands a stone Totem with inscriptions that are the key to understanding the purpose of life (42). However, getting to the Totem is a great challenge.

The Totem is situated on the opposite side of the hall form the entrance. The floor of the hall is covered with stone tiles that bear a striking resemblance to domino tiles. Each tile is divided into two halves (squares), and each half has a number from one to six, inclusive, chiseled into it. Tiles are arranged in **N rows**, with **N** tiles in each odd-numbered row and **N** - 1 tiles in each even-numbered row. The image below corresponds to the first test example (**N** = 5):



It is only possible to step from one tile to an adjacent one if the two tiles have matching numbers on halves that share an edge. Help Mister No find the shortest path to the Totem by determining the sequence of tiles (outputting the sequence of tiles' labels, described below) that need to be stepped on, in order, **from the first to the last tile on the path**. If there is no possible path to the Totem, find the shortest path to the tile with the **largest label** (so that Mister No can make a heroic jump). The stone tiles are labelled in **row-major** order: in the first row, the first tile has the label 1, and the last one **N**; in the second row, the first tile is **N** + 1, and the last one 2***N** - 1, and so on. The entrance leads to the tile with label 1, and the totem is on the last tile in the last row. Mister No always starts from the entrance.

INPUT

The first line of input contains the positive integer **N** ($1 \le N \le 500$), the number of stone tile rows.

Each of the following **N** * **N** - **N** / 2 lines (where "/" stands for integer division) contains two positive integers A_i and B_i ($1 \le A_i, B_i \le 6$, $1 \le i \le N * N - N / 2$), the values chiseled into the left and right halves, respectively, of tile **i**.

OUTPUT

The first line of output must contain the length (in tiles) of the required shortest path.

The second line of output must contain a sequence of space-separated positive integers, the labels of tiles on the shortest path. As there can exist more than one shortest path, output any one of them.

SCORING

If only the first line of output is correct, the solution is awarded 50% of points for that test case.

SAMPLE TESTS

input	input	input
5 1 4 4 5 3 4 5 4 5 2 4 2 5 6 4 4 6 5 2 4 5 1 6 1 1 6 2 3 4 2 5 3 1 2 5 5 4 1 2 2 4 3 2 3 3 4	3 1 2 2 3 6 6 2 4 3 5 6 6 4 5 5 6	4 1 5 5 3 5 5 5 6 5 3 6 4 4 5 2 5 2 4 4 3 2 4 5 2 1 4 1 6
output	**output**	**output**
7 1 2 7 12 17 22 23	4 1 2 5 8	7 1 5 8 12 9 10 13

In a galaxy far, far away, the fastest method of transportation is using hypertubes. Each hypertube directly connects **K** stations with each other. What is the minimum number of stations that we need to pass through in order to get from station 1 to station **N**?

INPUT

The first line of input contains three positive integers: **N** ($1 \le N \le 100\ 000$), the number of stations, **K** ($1 \le K \le 1\ 000$), the number of stations that any single hypertube directly interconnects, and **M** ($1 \le M \le 1\ 000$), the number of hypertubes.

Each of the following **M** lines contains the description of a single hypertube: **K** positive integers, the labels of stations connected to that hypertube.

OUTPUT

The first and only line of output must contain the required minimum number of stations. If it isn't possible to travel from station 1 to station **N**, output -1.

SAMPLE TESTS

input 9 3 5 1 2 3 1 4 5 3 6 7 5 6 7 6 8 9 **output** 4	**input** 15 8 4 11 12 8 14 13 6 10 7 1 5 8 12 13 6 2 4 10 15 4 5 9 8 14 12 11 12 14 3 5 6 1 13 **output** 3

Clarification of the first example: It is possible to travel from station 1 to station 9 using only four stations in the following ways: 1-3-6-9, or 1-5-6-9.

Mislav and Marko have devised a new game, creatively named Rotate. First, Mirko imagines a number sequence of length **N** and divides it into sections, with each section containing **K** numbers (**K** evenly divides **N**). The first section contains numbers in the first **K** positions in the sequence, the second section the following **K** positions, and so on.

Then, Marko asks Mislav to apply a number of operations on the sequence, with each operation being one of the following two types:

1. Rotate the numbers in each section to the left/right by **X** positions
2. Rotate the whole sequence to the left/right by **X** positions

Notice that an operation of type 2 can change the numbers belonging to each section. After applying all the operations, Mislav reveals the final sequence to Marko. Marko's task is finding Mislav's starting sequence. He has asked you for help.

INPUT

The first line of input contains three positive integers: **N** (1 ≤ **N** ≤ 100 000), the length of the sequence, **K** (1 ≤ **K** ≤ 100 000), the size of each section, and **Q** (1 ≤ **Q** ≤ 100 000), the number of operations.

Each of the following **Q** lines contains two integers: **A** (1 ≤ **A** ≤ 2), the operation type, and **X** (-100 000 ≤ **X** ≤ 100 000), the number of positions to rotate by. A negative number represents rotation to the left, while a positive one represents rotation to the right.

The last line of input contains **N** space-separated integers $\mathbf{Z}_i$ (0 ≤ $\mathbf{Z}_i$ ≤ 100 000) representing the final sequence (after applying all operations).

OUTPUT

The first and only line of output must contain the required starting sequence.

SCORING

In test data worth at least 40% of total points, **N** will be at most 100.

In test data worth at least 70% of total points, **K** will be at most 100.

SAMPLE TESTS

input	input	input
4 2 2 2 2 1 1 3 2 1 0	8 4 4 1 3 1 15 1 -5 2 -1 6 10 14 19 2 16 17 1	9 3 5 1 1 2 -8 2 9 1 1 2 -4 3 1 8 7 4 5 2 6 9
output	**output**	**output**
0 1 2 3	6 10 14 1 2 16 17 19	5 3 6 9 7 1 8 2 4

Clarification of the first example: The starting sequence is 0 1 2 3. After the first operations, the sequence is 2 3 0 1, and after the second operation, it becomes 3 2 1 0. Ths corresponds to the final sequence.

2N people are playing football (soccer), divided into two teams. Each player wears a dress with the team logo and a unique (in the team) positive integer between 1 and **N**, inclusive. For each player, we know his **precision**, the set of **teammates** whom he can pass the ball to (**F**) and the set of **opponents** who can take the ball from him (**E**). When a player comes into possession of the ball, after **exactly one second** one of the following events will happen:

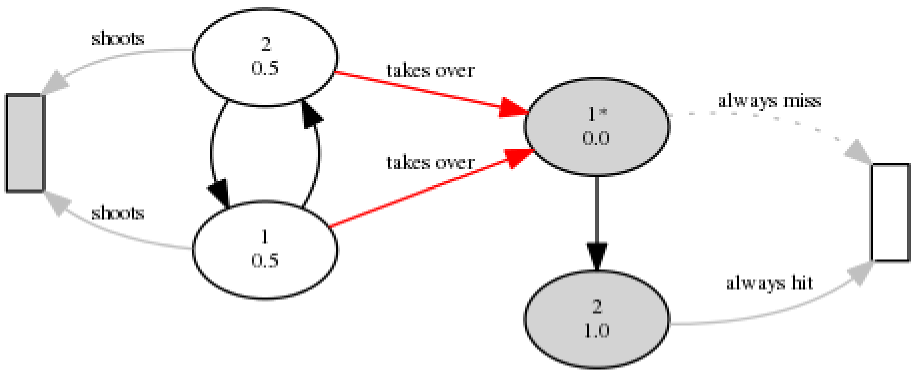
- the player passes the ball to a **random** teammate from the set **F**,
- a **random** opponent from the set **E** takes the ball from him,
- the player attempts a shot at the goal.

If the player attempts a shot, the **probability of scoring a goal** is equal to his **precision**. After the shot, whether it was successful or not, the ball is awarded to the player number 1 from the **opposing team**.

The probabilities of different events are in the proportion |**F**| : |**E**| : 1, in order, and depend only on the player currently in possession of the ball (|**S**| determines the size of the set **S** corresponding to the current player), not on any previous events in the game. The word "random" means that all players from the set **F** (or **E**) have the **same probability** of being passed (or taking) the ball by (from) the player that is currently in the ball's possession. The time that a ball spends outside of a player's possession is negligible.

The match begins with player 1 from the first team in possession of the ball and ends either when one team has scored **R goals** or when **T seconds** have elapsed, whichever happens first. For **each possible final score**, determine the **probability** that the match will end with it.

The following image illustrates the player arrangement for the second test example:



INPUT

The first line of input contains three positive integers: **N** ($1 \le N \le 100$), the number of players in each team, **R** ($1 \le R \le 10$), the number of goals needed for victory, and **T** ($1 \le T \le 500$), the maximum duration of the match.

The following **N** lines contain descriptions of the first team's players, one per line, while the next **N** lines after that contain descriptions of the second team's players. A description of a single player consists of a real number **p** ($0 \le p \le 1$), the player's precision, followed by two positive integers, **nF** ($0 \le nF \le N - 1$) and **nE** ($0 \le nE \le N$), the sizes of the sets **F** and **E**, respectively, followed by **nF** + **nE** player labels representing the sets **F** and **E** themselves (in that order), all space-separated. Note that the labels from **F** represent players from one team, and labels from **E** the other team. The set **F** will not contain the label of the player currently being described.

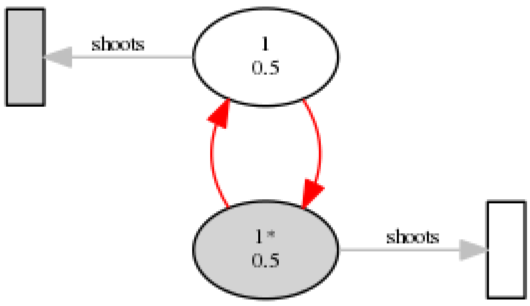
OUTPUT

The match can theoretically end with one of **R** * (**R** + 2) different final results. For each result, output the **probability** of its realization as a real number, one per line. Order the results first by the number of goals scored by the first team, then by the number of goas scored by the second team, in ascending order. The permitted difference from the exact value for each probability is 0.000001.

SAMPLE TESTS

input 1 1 2 0.5 0 1 1 0.5 0 1 1 **output** 0.56250 0.18750 0.25000	**input** 2 2 5 0.0 1 2 2 1 2 1.0 0 0 0.5 1 0 2 0.5 1 0 1 **output** 0.2578125 0.2812500 0.0703125 0.1718750 0.1640625 0.0234375 0.0156250 0.0156250

Clarification of the first example:



The star denotes the player in possession of the ball in the beginning. The match lasts for only **T** = 2 moves or until someone scores **R** = 1 goal. Since **N** = 1, there are only two players in the match, playing one against the other. Both players have the precision of 0.5, which means that each has a 50% chance of scoring a goal when trying to shoot, after which the ball is awarded to the opponent.

Let us label the grey player as A, and the white player as B. Under these assumptions, there are only 6 possible matches. Each of them is described in the table below, with the corresponding probability, description and outcome:

0.25	A decides to shoot and scores!	1:0
0.25 * 0.25	A decides to shoot, but misses. B decides to shoot and scores!	0:1
0.25 * 0.25	A decides to shoot, but misses. B decides to shoot, but also misses.	0:0
0.50 * 0.25	A loses the ball to B. B decides to shoot and scores!	0:1
0.50 * 0.50	A loses the ball to B. B loses the ball to A.	0:0
0.50 * 0.25	A loses the ball to B. B decides to shoot, but misses.	0:0

By summing probabilities for particular final results, we obtain the following solution:

0:0	0.25 * 0.25 + 0.5 * 0.5 + 0.5 * 0.25	0.5625
0:1	0.25 * 0.25 + 0.5 * 0.25	0.1875
1:0	0.25	0.25