Mei and the Magic Runes
Problem ID: meimagicrune

Mei, the renowned archaeologist, discovered some ancient text about runes believed to have magical properties. There are three basic runes: "heaven", "earth", and "life." A sequence of basic runes forms a composed rune. However, a composed rune can never begin with "life," as placing "life" before "heaven" and "earth" was considered blasphemy by the ancient people.

According to the ancient text, any two composed runes can be combined to form a new rune, but the exact rules of the combining operation are lost. Fortunately, Mei has already uncovered some examples of combining runes as listed below:

1. Combining "heaven life" and "heaven" results in "heaven heaven."
2. Combining "heaven life" and "earth" results in "heaven earth."
3. Combining "heaven life" and "heaven life" results in "earth life."
4. Combining "heaven" and "heaven" results in "earth."
5. Combining "heaven" and "earth" results in "heaven life."
6. Combining "earth" and "earth" results in "heaven heaven."
7. Combining "heaven life life" and "earth life" results in "heaven earth life."
8. Combining "earth life heaven" and "earth earth earth" results in "heaven earth life life."
9. Combining "earth earth earth earth" and "heaven" results in "heaven life life life life."
10. The result of combining two runes, A and B, is the same as combining B and A.
11. The result of combining any three runes is the same no matter which two are combined first.

Can you believe it? According to Mei, this ancient culture once followed a formal, computational philosophy that explained the universe as an immense computing machine. The rune combining rules is the key to understanding their culture in more detail. Can you guess what was in the mind of the ancient people? Write a program that outputs the result of combining the two input runes.

Input

The first line of the input contains two positive numbers: $1 \leq n, m \leq 1000$, the number of basic runes in the two composed runes to be combined. The second and the third line each describes a composed rune. They contain space-separated basic runes: "heaven", "earth", or "life" (without quotes, lowercase only).

Output

Output a single line of space-separated basic runes that describes the result of composing the two input runes.

Sample Input 1	Sample Output 1
3 3 heaven heaven heaven heaven heaven life	earth earth heaven

Ellis and the Ethereal Cat
Problem ID: ellisetherealcat

Ellis, a regular high-school student, is on their way home after a typical school day. But then they spots an ethereal cat floating around! Out of curiosity, they begins to chase after it. However, to Ellis's surprise, the ethereal cat can manipulate space, time, and even travel between parallel universes!!

Initially, the distance between Ellis and the cat is d unit distance. If the distance between Ellis and the cat is not zero, Ellis always moves toward the cat at a unit distance per second. The ethereal cat can take the following actions:

1. Wait for x units of time without moving in the current universe.
2. Teleport d unit distance further away from Ellis instantly in the current universe.
3. Rewind time backward x units. This action creates and enters a new parallel universe where everything that happened in this universe until the current time minus x (including any teleportation at exactly that time) has already happened in the newly created parallel universe.
4. Return to the parallel universe from which it entered the current universe. That universe's time and the positions of Ellis and the cat are the same as when the cat left it.

At any time, Ellis of the current universe may also ask you to compute the closest distance ever between them and the cat in the current universe's history. "But that would be too easy for you!" says the cat. So it would not tell you the details of its further plans unless you could answer Ellis's questions correctly.

Input

The first line of the input contains two integers $(1 \leq n \leq 2 * 10^5)$ and $(1 \leq d \leq 10^9)$, the total number of actions and the initial distance between Ellis and the ethereal cat.

Each of the following n lines contains two encrypted integers $(1 \leq x', y' \leq 2147483647)$. The first integer describes the type of the action, and the second integer is the parameter.

To decrypt the integers, you must compute the xor value of the input and the answer to the previous query or 0 if there haven't been any queries.

Let x and y be the decrypted values of x', y' respectively. It is guaranteed that $0 \leq x \leq 4$ and $0 \leq y \leq 10^9$. If x is 0, this is a query you must answer, and y is not used; otherwise, this is the x-th kind of action the cat may perform described above, and y would be the parameter for a wait, teleport, or rewind action, and not used for the return action.

It is guaranteed that the cat will not move time before time 0 or perform a return action in the initial universe.

Output

For each query outputs a single integer, the minimal distance between Ellis and the cat in the query's parallel universe's history so far.

Sample Input 1	Sample Output 1
8 10 2 5 1 8 0 10 4 4 7 3 14 1 11 0 10 15	7 10 0

Ivy and the Programming Contest
Problem ID: ivyprogrammingcontest

Ivy and her friends are participating as a team in a programming contest for elementary school students today. Although it is their first time competing, Ivy, as the team captain, is trying to develop the optimal strategy for the team to be ranked as high as possible.

All teams have t minutes to work on n problems in this contest. Teams are first ranked by the number of problems they solve and then by their penalty time as the tie-breaker. If a team solves their first problem x_1 minutes after the contest starts, their second at x_2 minutes, ..., their last problem at x_n minutes, that team gets a total of $x_1+x_2+...+x_n$ penalty time. A team solving no problems has a penalty time of 0. A team cannot solve a problem at or after the t-th minute.

Ivy's team has m contestants. After some quick assessment, Ivy estimates that it would take c_{ij} minutes for the i-th team member to solve the j-th problem. Ivy can assign problems to team members. A single member can only work on one problem at a time, and a problem can only be assigned to a single member. But different members can work on different problems in parallel. Consider the following example where the contest has 3 problems with a 30-minute time limit, and Ivy's team has two members:

	A	B	C
Member 1	8	20	32
Member 2	15	22	29

Ivy's best strategy is to have the first member solve problems A and B in that order and have the second member work on problem C. In theory, the team would solve problem A 8 minutes after the contest, problem B at $8 + 20 = 28$ minutes, and problem C at 29 minutes, resulting in a total penalty time of 65 minutes. Any other assignment would result in fewer problems solved or more time.

Could you write a program to compute the maximal number of solved problems and the minimal penalty time of solving that many for Ivy?

Input

The first line of the input contains 3 space-separated integers, $1 \leq n \leq 15, 1 \leq m \leq 6, 1 \leq t \leq 10^9$, the number of problems, the number of team members in Ivy's team, and the total time limit. The next m lines each contains n space-separated integers. The j-th integer in the $i+1$-th line is $1 \leq t_{ij} \leq 10^9$, the estimated time for the i-th team member to finish the j-th problem.

Output

Output two integers in a single line separated by a single space, the maximal number of solved problems, and the minimal penalty time of solving that many problems.

Sample Input 1

```
3 2 30
8 20 32
15 22 29
```

Sample Output 1

```
3 65
```

Sample Input 2

```
5 3 120
10 10 30 60 1440
8 12 15 45 1440
20 20 40 5 1440
```

Sample Output 2

```
4 46
```

Shirley and the Mimicking Mimics
Problem ID: shirleymimics

Shirley, a "consulting detective", came across a cave full of treasure chests! But Shirley thinks that some of the treasure chests might actually be mimics, a type of monster that mimics the look of a treasure chest and makes fun of the unfortunate people who foolishly think they are treasure chests. The problem is: there is no way to distinguish between the real and the mimic only by their looks!

However, as an expert detective, Shirley managed to record some dialogues between the treasure chests and the mimic chests. (Yes, the chests speak, and you can understand them if you are an expert detective!) And she knows from her experience that authentic chests only speak the truth and mimic chests only tell lies. Based on this information, the identity of some chests could be deduced. Consider the following scenario of three chests, A, B, and C:

1. Chest A: C is mimic.
2. Chest B: C is real.
3. Chest C: Both A and B are mimics.

Since A and B disagree on the identity of C, it must be the case that one of them is real and the other is a mimic. Thus, what C said must be false, and it is a mimic. Then A tells the truth, and B is lying. Therefore, A is real, and B and C are mimics.

Given the dialogue logs of the chests, can you help Shirley find out which chests are real?

Input

The first line of the input contains two integers, $1 \leq n \leq 10$, the number of chests, and $1 \leq m \leq 1000$, the number of log entries. Each of the following m lines describes a statement made by one of the chests. Chests are conveniently labeled by the first n uppercase letters starting from A. Each statement can be one of the following patterns. X, Y, and Z are placeholders for chests' labels.

1. X says Y is mimic
2. X says Y is real
3. X says both Y and Z are mimics
4. X says both Y and Z are real

It is guaranteed that the identity of the chests can be uniquely deduced from the input.

Output

Output n lines. For the i-th line, output "real" (without quotes) if the chest labeled by the i-th letter is real, "mimic" (without quotes) otherwise.

Sample Input 1	Sample Output 1
2 1 A says both A and B are mimics	mimic real

Sample Input 2	Sample Output 2
3 2 B says both A and C are real C says B is mimic	mimic mimic real

Naomi and the Gemini Bots
Problem ID: naomigeminibots

Naomi, the roboticist, is testing her latest invention - the Gemini Bots. The testing ground is a 1000x1000 grid with walls on all four sides. The lower-left corner has the coordinates $(0, 0)$, and the upper-right corner has the coordinates $(999, 999)$.

The Gemini Bots, as the name suggests, consist of two robots connected through the Gemini System. The first robot starts at (x_1, y_1) and needs to go to (x_2, y_2). The second robot starts at (x_3, y_3) and needs to go to (x_4, y_4). In each step, Naomi can send one of four commands: up, right, down, and left. The first robot will move exactly one cell in the direction of the command. So a "right" command increases the first coordinate by one, and "left" decreases it by one. "up" increases the second coordinate by one, and "down" decreases it by one. However, the second robot in the system will move in the direction rotated 90 degrees clockwise, i.e., right for up, down for right, left for down, and up for left. As an additional restriction, if a move would cause a robot to move outside of the testing ground, it will stay in place instead. The two robots may occupy the same cell and won't interfere with each other's movement.

Help Naomi conduct the test by finding a sequence of commands that would move the Gemini Robots from the starting locations to the target locations, if there is one.

Input

The single line contains 8 space-separated integers, $1 \leq x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4 \leq 999$, the initial and target coordinates of the first bot and the initial and target coordinates of the second bot. It is guaranteed that at least one robot's initial coordinates are different from its target coordinates.

Output

If no sequence of commands can make both bots reach the target coordinates, print a single line containing the string "IMPOSSIBLE" (without quotes).

Otherwise, the output should contain a single string, no longer than 20000 characters, only consisting of "URDL", which corresponds to Up, Right, Down, Left, respectively, describing a sequence of the commands.

Any sequence of commands that move the robots from the initial positions to the target positions will be judged as correct. Your output doesn't have to be the shortest of such sequences.

Sample Input 1	Sample Output 1
`0 0 0 1 0 0 1 0`	`U`

Sample Input 2	Sample Output 2
`0 0 1 1 2 2 1 1`	`DDUR`

Sample Input 3	Sample Output 3
`0 0 1 1 2 2 1 1`	`DDRUUD`

Beatrice and the Tea party
Problem ID: beatriceteaparty

Beatrice enjoys fine tea and fun parties, and she is having a wonderful tea party with her friends. To make the party more enjoyable, Beatrice comes up with a rule to have her friends change their seats once in a while. The seats are numbered from 1 to n. And each time Beatrice rings a bell, the person who sits at seat i will move to seat p_i. So Beatrice's friends can socialize with different people each time.

But Beatrice's little brother, Brian, messed up p_i. So some of Beatrice's friends might find out that they must move to the same seat. When this happens, the person whose last seat number is the smallest will take the new seat, and the others will sadly leave the party.

Despite this, Beatrice thinks at least some friends will still stick around the party indefinitely. Given the sequence p_i, can you tell Beatrice how many friends will stay?

Input

The first line contains a single number, n, $1 \leq n \leq 10^5$, the number of seats, and the number of friends in the tea party initially. The second line contains n space-separated integers, p_i, $1 \leq p_i \leq n$, as described above.

Output

Output a single integer, the number of friends who will stay indefinitely.

Sample Input 1	Sample Output 1
10 1 1 4 5 3 9 10 7 8 9	8

Sample Input 2	Sample Output 2
6 2 3 4 5 6 6	1