A. Room Leader

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

Let us remind you part of the rules of Codeforces. The given rules slightly simplified, use the problem statement as a formal document.

In the beginning of the round the contestants are divided into rooms. Each room contains exactly n participants. During the contest the participants are suggested to solve five problems, A, B, C, D and E. For each of these problem, depending on when the given problem was solved and whether it was solved at all, the participants receive some points. Besides, a contestant can perform hacks on other contestants. For each successful hack a contestant earns 100 points, for each unsuccessful hack a contestant loses 50 points. The number of points for every contestant is represented by the sum of points he has received from all his problems, including hacks.

You are suggested to determine the leader for some room; the leader is a participant who has maximum points.

Input

The first line contains an integer n, which is the number of contestants in the room ($1 \le n \le 50$). The next n lines contain the participants of a given room. The i-th line has the format of "$handle_i$ $plus_i$ $minus_i$ a_i b_i c_i d_i e_i" — it is the handle of a contestant, the number of successful hacks, the number of unsuccessful hacks and the number of points he has received from problems A, B, C, D, E correspondingly. The handle of each participant consists of Latin letters, digits and underscores and has the length from 1 to 20 characters. There are the following limitations imposed upon the numbers:

- $0 \le plus_i, minus_i \le 50$;
- $150 \le a_i \le 500$ or $a_i = 0$, if problem A is not solved;
- $300 \le b_i \le 1000$ or $b_i = 0$, if problem B is not solved;
- $450 \le c_i \le 1500$ or $c_i = 0$, if problem C is not solved;
- $600 \le d_i \le 2000$ or $d_i = 0$, if problem D is not solved;
- $750 \le e_i \le 2500$ or $e_i = 0$, if problem E is not solved.

All the numbers are integer. All the participants have different handles. It is guaranteed that there is exactly one leader in the room (i.e. there are no two participants with the maximal number of points).

Output

Print on the single line the handle of the room leader.

Examples

input

```
5
Petr 3 1 490 920 1000 1200 0
tourist 2 0 490 950 1100 1400 0
Egor 7 0 480 900 950 0 1000
c00lH4x0R 0 10 150 0 0 0 0
some_participant 2 1 450 720 900 0 0
```

output

```
tourist
```

Note

The number of points that each participant from the example earns, are as follows:

- `Petr` — 3860
- `tourist` — 4140
- `Egor` — 4030
- `c00lH4x0R` — -350
- `some_participant` — 2220

Thus, the leader of the room is `tourist`.

B. Train

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

A stowaway and a controller play the following game.

The train is represented by n wagons which are numbered with positive integers from 1 to n from the head to the tail. The stowaway and the controller are initially in some two different wagons. Every minute the train can be in one of two conditions — moving or idle. Every minute the players move.

The controller's move is as follows. The controller has the movement direction — to the train's head or to its tail. During a move the controller moves to the neighbouring wagon correspondingly to its movement direction. If at the end of his move the controller enters the 1-st or the n-th wagon, that he changes the direction of his movement into the other one. In other words, the controller cyclically goes from the train's head to its tail and back again during all the time of a game, shifting during each move by one wagon. Note, that the controller always have exactly one possible move.

The stowaway's move depends from the state of the train. If the train is moving, then the stowaway can shift to one of neighbouring wagons or he can stay where he is without moving. If the train is at a station and is idle, then the stowaway leaves the train (i.e. he is now not present in any train wagon) and then, if it is not the terminal train station, he enters the train again into any of n wagons (not necessarily into the one he's just left and not necessarily into the neighbouring one). If the train is idle for several minutes then each such minute the stowaway leaves the train and enters it back.

Let's determine the order of the players' moves. If at the given minute the train is moving, then first the stowaway moves and then the controller does. If at this minute the train is idle, then first the stowaway leaves the train, then the controller moves and then the stowaway enters the train.

If at some point in time the stowaway and the controller happen to be in one wagon, then the controller wins: he makes the stowaway pay fine. If after a while the stowaway reaches the terminal train station, then the stowaway wins: he simply leaves the station during his move and never returns there again.

At any moment of time the players know each other's positions. The players play in the optimal way. Specifically, if the controller wins, then the stowaway plays so as to lose as late as possible. As all the possible moves for the controller are determined uniquely, then he is considered to play optimally always. Determine the winner.

Input

The first line contains three integers n, m and k. They represent the number of wagons in the train, the stowaway's and the controller's initial positions correspondingly ($2 \le n \le 50$, $1 \le m, k \le n$, $m \ne k$).

The second line contains the direction in which a controller moves. "to head" means that the controller moves to the train's head and "to tail" means that the controller moves to its tail. It is guaranteed that in the direction in which the controller is moving, there is at least one wagon. Wagon 1 is the head, and wagon n is the tail.

The third line has the length from 1 to 200 and consists of symbols "0" and "1". The i-th symbol contains information about the train's state at the i-th minute of time. "0" means that in this very minute the train moves and "1" means that the train in this very minute stands idle. The last symbol of the third line is always "1" — that's the terminal train station.

Output

If the stowaway wins, print "Stowaway" without quotes. Otherwise, print "Controller" again without quotes, then, separated by a space, print the number of a minute, at which the stowaway will be caught.

Examples

input

```
5 3 2
to head
0001001
```

output

```
Stowaway
```

input

```
3 2 1
to tail
0001
```

output

```
Controller 2
```

C. Chessboard Billiard

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

Let's imagine: there is a chess piece *billiard ball*. Its movements resemble the ones of a bishop chess piece. The only difference is that when a billiard ball hits the board's border, it can reflect from it and continue moving.

More formally, first one of four diagonal directions is chosen and the billiard ball moves in that direction. When it reaches the square located on the board's edge, the billiard ball reflects from it; it changes the direction of its movement by 90 degrees and continues moving. Specifically, having reached a corner square, the billiard ball is reflected twice and starts to move the opposite way. While it moves, the billiard ball can make an infinite number of reflections. At any square of its trajectory the billiard ball can stop and on that the move is considered completed.

It is considered that one billiard ball a beats another billiard ball b if a can reach a point where b is located.

You are suggested to find the maximal number of billiard balls, that pairwise do not beat each other and that can be positioned on a chessboard $n \times m$ in size.

Input

The first line contains two integers n and m ($2 \le n, m \le 10^6$).

Output

Print a single number, the maximum possible number of billiard balls that do not pairwise beat each other.

Please do not use the `%lld` specificator to read or write 64-bit numbers in C++. It is preferred to use `cin` (also you may use the `%I64d` specificator).

Examples

input

```
3 4
```

output

```
2
```

input

```
3 3
```

output

```
3
```

D. Hanger

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In one very large and very respectable company there is a cloakroom with a coat hanger. It is represented by n hooks, positioned in a row. The hooks are numbered with positive integers from 1 to n from the left to the right.

The company workers have a very complicated work schedule. At the beginning of a work day all the employees are not there and the coat hanger in the cloakroom is empty. At some moments of time the employees arrive and some of them leave.

When some employee arrives, he hangs his cloak on one of the available hooks. To be of as little discomfort to his colleagues as possible, the hook where the coat will hang, is chosen like this. First the employee chooses the longest segment among available hooks following in a row. If there are several of such segments, then he chooses the one closest to the right. After that the coat is hung on the hook located in the middle of this segment. If the segment has an even number of hooks, then among two central hooks we choose the one closest to the right.

When an employee leaves, he takes his coat. As all the company workers deeply respect each other, no one takes somebody else's coat.

From time to time the director of this respectable company gets bored and he sends his secretary to see how many coats hang on the coat hanger from the i-th to the j-th hook inclusive. And this whim is always to be fulfilled, otherwise the director gets angry and has a mental breakdown.

Not to spend too much time traversing from the director's office to the cloakroom and back again, the secretary asked you to write a program, emulating the company cloakroom's work.

Input

The first line contains two integers n, q ($1 \le n \le 10^9$, $1 \le q \le 10^5$), which are the number of hooks on the hanger and the number of requests correspondingly. Then follow q lines with requests, sorted according to time. The request of the type "$0\ i\ j$" ($1 \le i \le j \le n$) — is the director's request. The input data has at least one director's request. In all other cases the request contains a positive integer not exceeding 10^9 — an employee identificator. Each odd appearance of the identificator of an employee in the request list is his arrival. Each even one is his leaving. All employees have distinct identificators. When any employee arrives, there is always at least one free hook.

Output

For each director's request in the input data print a single number on a single line — the number of coats hanging on the hooks from the i-th one to the j-th one inclusive.

Examples

input

```
9 11
1
2
0 5 8
1
1
3
0 3 8
9
0 6 9
6
0 1 9
```

output

```
2
3
2
5
```

E. Shift It!

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

There is a square box 6×6 in size. It contains 36 chips 1×1 in size. Those chips contain 36 different characters — "`0`"-"`9`" and "`A`"-"`Z`". There is exactly one chip with each character.

You are allowed to make the following operations: you may choose one of 6 rows or one of 6 columns and cyclically shift the chips there to one position to the left or to the right (for the row) or upwards or downwards (for the column). Those operations are allowed to perform several times.

To solve the puzzle is to shift the chips using the above described operations so that they were written in the increasing order (exactly equal to the right picture). An example of solving the puzzle is shown on a picture below.

Write a program that finds the sequence of operations that solves the puzzle. That sequence **should not necessarily be shortest**, but you should not exceed the limit of 10000 operations. It is guaranteed that the solution always exists.

Input

The input data are represented by 6 lines containing 6 characters each. They are the puzzle's initial position. Those lines contain each character from the string "`0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ`" exactly once.

Output

On the first line print number n, which is the number of operations. On the next n lines print the sequence of operations one per line. An operation is described by a word consisting of two characters. The first character shows the direction where the row or the column will be shifted. The possible directions are "`L`", "`R`" (to the left, to the right correspondingly, we shift a row), "`U`", "`D`" (upwards, downwards correspondingly, we shift a column). The second character is the number of the row (or the column), it is an integer from "`1`" to "`6`". The rows are numbered from the top to the bottom, the columns are numbered from the left to the right.

The number of operations should not exceed 10^4. If there are several solutions, print any of them.

Examples

input

```
01W345
729AB6
CD8FGH
IJELMN
OPKRST
UVQXYZ
```

output

```
2
R2
U3
```

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov
The only programming contests Web 2.0 platform