

## Task overview

---

Task	PATULJCI	NPUZZLE	TROJKE	TENKICI	BICIKLI	LISTA
Input	standard input ( <b>keyboard</b> )					
Output	standard output ( <b>screen</b> )					
Memory limit (heap)	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
Memory limit (stack)	8 MB	8 MB	8 MB	8 MB	8 MB	8 MB
Time limit (per test)	1 sec	1 sec	1 sec	1 sec	1 sec	1 sec
Number of tests	10	10	10	14	10	9
Points per test	2	2	3	5	7	10
Total points	<b>20</b>	<b>20</b>	<b>30</b>	<b>70</b>	<b>70</b>	<b>90</b>
	<b>300</b>					

**Note:** The time limit is based on a computer running two AMD Athlon MP 2600+ processors and Linux operating system.

C and C++ programs will be compiled with the following options: `-O2 -lm -static`

Pascal programs will be compiled with the following options: `-O1 -XS`

# 1. PATULJCI

---

Every day, while the dwarves are busy in the mines, Snow White prepares dinner for them; seven chairs, seven plates, seven forks and seven knives for seven hungry dwarves.

One day nine dwarves came from the mines instead of seven (nobody knows how or why), each of them claiming to be one of Snow White's seven dwarves.

Luckily, each dwarf wears a hat with a positive integer less than 100 written on it. Snow White, a famous mathematician, realised long ago that the sum of numbers on the hats of her seven dwarves was exactly 100.

Write a program which determines which dwarves are legit, i.e. pick seven of nine numbers that add to 100.

## Input

There are 9 lines of input. Each contains an integer between 1 and 99 (inclusive). All of the numbers will be distinct.

**Note:** The test data will be such that the solution is unique.

## Output

Your program must produce exactly seven lines of output – the numbers on the hats of Snow White's seven dwarves. Output the numbers in any order.

## Sample test data

<b>input</b>	<b>input</b>
7	8
8	6
10	5
13	1
15	37
19	30
20	28
23	22
25	36
<b>output</b>	<b>output</b>
7	8
8	6
10	5
13	1
19	30
20	28
23	22

## 2. NPUZZLE

---

N-puzzle is a puzzle that goes by many names and has many variants. In this problem we will use the 15-puzzle. It consists of a 4-by-4 grid of sliding squares where one square is missing. The squares are labeled with uppercase letters 'A' through 'O', with the desired layout as follows:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	.

It can be useful (for example, when solving the puzzle using a computer) to define the "scatter" of a puzzle as the sum of distances between each square's current position and its position in the desired layout. The distance between two squares is their Manhattan distance (the absolute value of the sum of differences between the two rows and the two columns).

Write a program that calculates the scatter of the given puzzle.

### Input

Four lines of input contain four characters each, representing the state of the puzzle.

### Output

Output the scatter of the puzzle on a single line.

### Sample test data

**input**

```
ABCD
EFGH
IJKL
M.NO
```

**output**

```
2
```

**input**

```
.BCD
EAGH
IJFL
MNOK
```

**output**

```
6
```

### 3. TROJKE

---

Mirko and Slavko are playing a new game, "Trojke" (Triplets). First they use a chalk to draw an N-by-N square grid on the road. Then they write letters into some of the squares. No letter is written more than once in the grid.

The game consists of trying to find three letters on a line as fast as possible. Three letters are considered to be on the same line if there is a line going through the centre of each of the three squares.

After a while it gets harder to find new triplets. Mirko and Slavko need a program that counts all the triplets, so that they know if the game is over or they need to search further.

#### Input

The first line contains an integer N ( $3 \leq N \leq 100$ ), the dimension of the grid.

Each of the N following lines contains N characters describing the grid – uppercase letters and the character '.', which marks an empty square.

#### Output

Output the number of triplets on a single line.

#### Sample test data

**Input**

```
4
...D
..C.
.B..
A...
```

**output**

```
4
```

**input**

```
5
..T..
A....
.FE.R
....X
S....
```

**output**

```
3
```

**input**

```
10
....AB....
..C....D..
.E.....F.
...G..H...
I.....J
K.....L
...M..N...
.O.....P.
..Q....R..
....ST....
```

**output**

```
0
```

## 4. TENKICI

---

Mirko found a collection of  $N$  toy tanks dating back to the Second World War on his grandfather's attic. He promptly called his friend Slavko to play with him. They made a battlefield – a wooden board consisting of squares in  $N$  rows and  $N$  columns.

Each tank can be moved to one of the four neighbouring squares in a single move. A tank can shoot at any square in the same row and column. The tank is said to be guarding the row and column it is in.

Additionally, no two tanks can be in the same square at any time.

After many hours of play and two previous attempts, Mirko's mom yelled at them to come down for lunch again, and they decided to rearrange the tanks so that each tank guards a different row and column (meaning also that each row and column contains only one tank).

However, they want to do this using the minimum number of moves.

Write a program that finds the minimum number of moves required to rearrange the tanks so that each row and each column contains a single tank, and one such shortest sequence of moves.

### Input

The first line of input contains the integer  $N$  ( $5 \leq N \leq 500$ ).

Each of the following  $N$  lines contains two integers  $R$  and  $C$  ( $1 \leq R, S \leq N$ ), the row and column of a single tank at the moment of mom's call. No two tanks are on the same square.

Rows and columns are marked 1 through  $N$ , top-down and left-to-right.

### Output

Output the minimum number of moves (call this number  $K$ ) on the first line.

Each of the next  $K$  lines should contain the tank being moved and the direction it is moved in, separated by a single space.

Tanks are numbered 1 through  $N$ , in the order in which they are given in the input.

The direction can be one of four uppercase letters: 'L' for left, 'R' for right, 'U' for up and 'D' for down.

**Note:** The sequence need not be unique.

### Scoring

If both the number  $K$  and the sequence of moves are correct, your program will score full points on the test case.

If your program outputs the correct number  $K$  and does not output the sequence of moves, or the sequence of moves is incorrect, you will get 60% of the points for that test case.

## 4. TENKICI

---

### Sample test data

**input**

5  
1 1  
1 2  
1 3  
1 4  
1 5

**output**

10  
1 D  
2 D  
3 D  
4 D  
1 D  
2 D  
3 D  
1 D  
2 D  
1 D

**input**

5  
2 3  
3 2  
3 3  
3 4  
4 3

**output**

8  
1 R  
1 R  
2 U  
2 U  
4 D  
4 D  
5 L  
5 L

**input**

6  
1 1  
1 2  
2 1  
5 6  
6 5  
6 6

**output**

8  
2 R  
2 D  
3 D  
3 R  
4 U  
4 L  
5 L  
5 U

## 5. BICIKLI

---

A bicycle race is being organized in a land far, far away. There are  $N$  towns in the land, numbered 1 through  $N$ . There are also  $M$  one-way roads between the towns. The race will start in town 1 and end in town 2.

How many different ways can the route be set? Two routes are considered different if they do not use the exact same roads.

### Input

The first line of input contains two integers  $N$  and  $M$  ( $1 \leq N \leq 10\,000$ ,  $1 \leq M \leq 100\,000$ ), the number of towns and roads.

Each of the next  $M$  lines contains two different integers  $A$  and  $B$ , representing a road between towns  $A$  and  $B$ .

Towns may be connected by more than one road.

### Output

Output the number of distinct routes that can be set on a single line. If that number has more than nine digits, output only the last nine digits of the number. If there are infinitely many routes, output "inf".

### Sample test data

**input**

6 7  
1 3  
1 4  
3 2  
4 2  
5 6  
6 5  
3 4

**output**

3

**input**

6 8  
1 3  
1 4  
3 2  
4 2  
5 6  
6 5  
3 4  
4 3

**output**

inf

**input**

31 60  
1 3  
1 3  
3 4  
3 4  
4 5  
4 5  
5 6  
5 6  
6 7  
6 7  
...  
...  
...  
28 29  
28 29  
29 30  
29 30  
30 31  
30 31  
31 2  
31 2

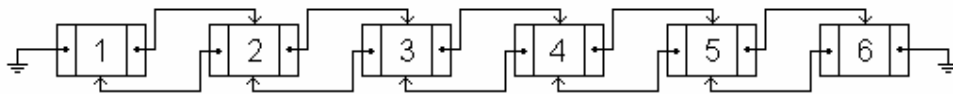
**output**

073741824

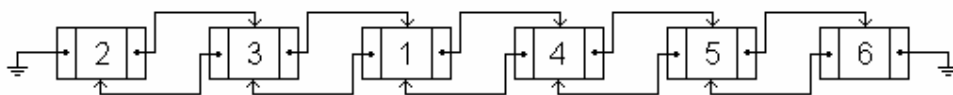
## 6. LISTA

Mirko received a birthday present from his aunt in the US – a brand-new doubly-linked list (an example of which is shown in the figure below). The list contains  $N$  nodes numbered 1 through  $N$ . Two types of moves can be done on the list:

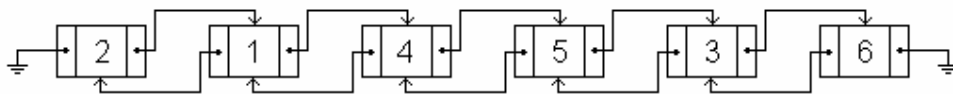
- A) Move node  $X$  **in front of** node  $Y$ .
- B) Move node  $X$  **after** node  $Y$ .



*An example of a list with 6 nodes.*



*The list after the move "A 1 4".*



*The list after another move, "B 3 5".*

Mirko played with his new toy for hours, writing down each move on a piece of paper so that he can reconstruct the list's initial state (nodes 1 through  $N$  in order from left to right).

When he decided to reconstruct the list, Mirko was astonished to find that there is no easy way to invert the moves and restore the list's initial state. Mirko cannot know where node  $X$  was prior to each move, only where it ended up.

Seeing how Mirko is still recovering from the shock, write a program that finds a minimal sequence of moves that restored the list's initial state from Mirko's logs.

### Input

The first line of input contains two integers  $N$  and  $K$  ( $2 \leq N \leq 500\,000$ ,  $0 \leq K \leq 100\,000$ ), the number of nodes and the number of moves made by Mirko.

Each of the next  $M$  rows contains a description of a single move made by Mirko – the type of move ('A' or 'B') and two integers  $X$  and  $Y$ .

### Output

Output the minimum number of moves (call this number  $K$ ) on the first line.

Each of the next  $K$  lines should contain a description of a single move in the same format as in the input.

**Note:** The sequence need not be unique.

## 6. LISTA

---

### Scoring

If both the number  $K$  and the sequence of moves are correct, your program will score full points on the test case.

If your program outputs the correct number  $K$  and does not output the sequence of moves, or the sequence of moves is incorrect, you will get 60% of the points for that test case.

### Sample test data

**input**

2 1  
A 2 1

**output**

1  
A 1 2

**input**

4 3  
B 1 2  
A 4 3  
B 1 4

**output**

2  
A 1 2  
B 4 3

**input**

6 5  
A 1 4  
B 2 5  
B 4 2  
B 6 3  
A 3 5

**output**

3  
A 4 5  
B 6 5  
A 2 3