Jenga

Time Limit: 4 s *Memory Limit: 512 MB*

Jenga is a simple game of removing individual blocks from a tower without crashing it. Even if you're familiar with the rules, please read the description because the rules of the game have been slightly changed for the purpose of this task.

At the beginning of the game, a tower of 3-block storeys or levels is built so that blocks in each storey are perpendicular to those directly below them. Two players take turns removing blocks from the tower. When it is his/her turn, a player must remove one block from anywhere, except from the topmost storey and then add it to the topmost storey. If the topmost storey is already full (has 3 blocks), a player starts building a new storey by placing the block on top of it. A player who crashes the tower on his/her turn loses the game. Players may only touch blocks to remove them from the tower and place them on top. A player is not allowed to move or touch blocks in any other way.

You are a reigning champion in Jenga. You are skilled to the point where - if physically possible - you can remove any block without crashing the tower. But so are your opponents. A new challenger wants to take away your prestigious champion title. Just like you, he is highly skilled and will not crash the tower by accident. Can you win the match?

Initially, the tower has N storeys, numbered $1 \ldots N$. Each storey has 3 blocks, numbered 1, 2 and 3 (with block number 2 as the middle one). If at some point in the game, a storey has no more blocks, has only block 1 or only block 3, the tower crashes. Otherwise, the tower will not crash. For the purpose of this task, we will ignore centre of mass and other physical phenomena.

As a current champion, you may choose to go first or second. On each of your turns, you must remove a block and place it on the top storey. The game ends when one of the players crashes the tower.

Input and output

This task is interactive. Your program should read from standard input and write to standard output. Your program should win the match of Jenga against the grader.

The first line of the input contains the number of storeys N. Your first line of output should be "first" if you want to start first or "second" if you want to start second.

On your turn, your program should print two numbers separated with a space, L and B. This means, that you are removing block B from storey L and placing it on the top. The grader (opponent) will format the input for your program in the same way. Your program should end when the tower crashes.

Note that the interaction must end with a move crashing the tower by you or preferably your opponent. After that your program has to successfully exit (regardless of the winner). Otherwise the verdict might be Time Limit Exceeded. Don't forget to flush any output printed to the standard output. Your grader will get points for each winning match against the grader.

Constraints

- $2 \leq N \leq 1000$

Subtask 1 (20 points)

- $N = 2$

Subtask 2 (20 points)

- $N \leq 10$

Subtask 3 (20 points)

- $N \leq 40$

Subtask 4 (40 points)

- no additional constraints

Example

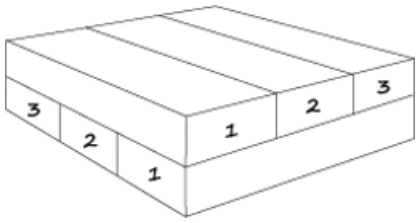
Grader: Program:

```
2
         first
           1 1
1 3
           2 2
2 1
```

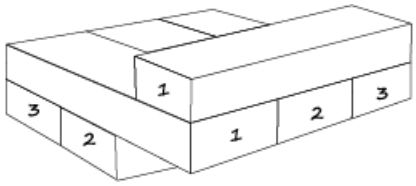
Comment

This example shows a valid output of the program. Note that input and output are an interactive conversation. The left aligned text is the input to the program and the right aligned text is the output of the program.

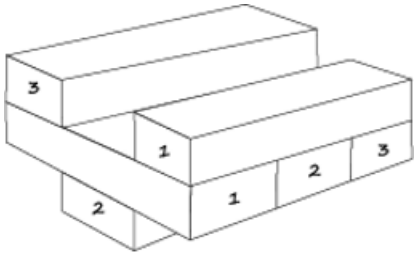
At the beginning we have a tower with two storeys and we decide to go first.



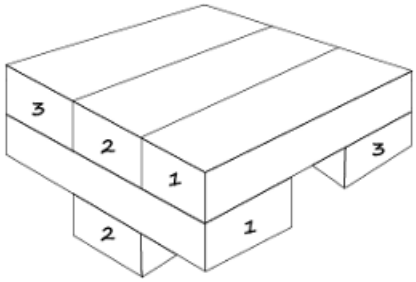
We remove block 1 from storey 1.



The opponent (grader) removes block 3 from storey 1.



We remove block 2 from storey 2.



The grader cannot remove the blocks from the topmost storey. Any other block he pulls will crush the tower. They choose block 1 from storey 2 and lose.

Multiply

Time Limit: 2 s *Memory Limit: 128 MB*

Write a program that computes a product of two non-negative integers A and B. The integers are represented in decimal notation and have N and M digits, respectively.

Constraints

- $1 \leq N, M \leq 50\,000$

Subtask 1 (20 points)

- $N, M \leq 4$

Subtask 2 (20 points)

- $N, M \leq 9$

Subtask 3 (30 points)

- $N, M \leq 5\,000$

Subtask 4 (30 points)

- no additional constraints

Input

The first line contains the lengths N and M, separated by a space. A is given on the second and B on the third line. The numbers will not have leading zeros.

Output

Output the product of A and B without leading zeros.

Examples

Input	Output
3 4 123 4567	561741

Input	Output
3 1 100 0	0

Museum

Time Limit: 3 s *Memory Limit: 1024 MB*

A tourist just walked into a museum that houses a treasured collection of clean drinking water from different parts of the world. Fortunately, it is only a temporary exhibition to raise awareness but might become a permanent thing in the future.

The museum consists of n rooms (numbered from 1 to n) that are connected with each other by doors and passages. Each passage connects two rooms directly, without passing through other rooms. The layout of the museum is such that between every pair of rooms, there is exactly one simple path (possibly passing through one or more intermediary rooms). The tourist is currently located in room x. He has a map of the museum and thus knows for every passage i that it connects rooms a_i and b_i, and that it takes c_i time to walk the length of that passage.

He would like to visit k different rooms (including the starting room x). He will spend an insignificant amount of time in every room. It doesn't matter in which room he finishes his visit. What is the shortest possible time in which he can achieve this?

Constraints

- $1 \leq n \leq 10\,000$
- $1 \leq k, x \leq n$
- $1 \leq a_i, b_i \leq n$
- $0 \leq c_i \leq 10\,000$

Subtask 1 (20 points)

- $n \leq 20$

Subtask 2 (25 points)

- $k \leq 100$
- every room has at most 3 adjacent rooms

Subtask 3 (35 points)

- $k \leq 100$

Subtask 4 (20 points)

- no additional constraints

Input

First line contains integers n, k and x. The following $n-1$ lines describe passages between rooms with integers a_i, b_i and c_i, indicating that there is a passage between rooms a_i and b_i that takes c_i time to move through.

Output

Output the minimum time required to visit k rooms.

Examples

Input	Output
11 8 3 1 3 3 3 2 5 6 4 5 1 11 3 9 1 2 9 10 2 3 7 10 6 7 1 7 8 1 7 5 1	29

Input	Output
3 1 1 1 2 4 2 3 0	0