

Misere Business

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

points: 100 · problem code: miserebusiness · author: suisei

After a long day of sending RSA-encrypted love letters to each other, Alice and Bob decide to grab dinner.

The waiter presents their meal: a dish elegantly divided into n equal pieces.

Alice and Bob decide to take turns picking at least 1 and at most k pieces per turn — just enough to avoid appearing gluttonous.

For various reasons, neither Alice nor Bob wants to take the last piece of food.

Bob, being the gentleman that he is, lets Alice choose whether she wants to go first or second.

You need to help Alice decide if she should go first or second, and subsequently, how many pieces of food she should take in response to how many Bob takes.

Interaction

This is an interactive problem, which means your program and the judge system communicate via input and output.

The first line of input contains integers n and k .

Your code should then either print:

```
First
x
```

If Alice should go first, where x is the number of pieces she should take first, or

```
Second
```

If Alice should go second.

Then, interaction follows a simple pattern. The judge will print a single integer, representing the number of pieces Bob has selected.

Your program should then output a single integer, representing the number of pieces Alice should select.

After the final piece is selected, both programs complete without producing any more output.

Example interaction

This is just an example to show input/output. Each player may or may not be playing optimally in this example.

The judge first inputs

```
10 5
```

Your code might then print:

```
First
4
```

Now, 6 pieces remain. The judge might respond with

```
2
```

Now, 4 pieces remain. Your code might then print

```
3
```

Now, 1 piece remains. Bob eats the final piece, and both programs complete.

Constraints

- $1 \leq n \leq 10^9$
- $1 \leq k \leq 10^9$

Python Template

```
n,k = map(int,input().split())

def get_move():
    # IMPLEMENT THIS
    pass

# if you want to go first:
print("First",flush=True)
move = get_move()
print(move)

# if you want to go second:
print("Second",flush=True)

while n>0:

    # judge's move
    response = int(input())

    # process the response

    if n > 0:
        # make your move
        move = get_move()
        print(move,flush=True)

    # remember to track your changes accordingly
```