Double Dealing

Take a deck of n unique cards. Deal the entire deck out to k players in the usual way: the top card to player 1, the next to player 2, the k^{th} to player k, the $(k+1)^{st}$ to player 1, and so on. Then pick up the cards – place player 1's cards on top, then player 2, and so on, so that player k's cards are on the bottom. Each player's cards are in reverse order – the last card that they were dealt is on the top, and the first on the bottom.

How many times, including the first, must this process be repeated before the deck is back in its original order?

Input

There will be up to $32\,000$ test cases in the input. Each case will consist of a single line with two integers, n and k ($1 \leq n \leq 800$, $1 \leq k \leq 800$). The input will end with a line with two 0s.

Output

For each test case in the input, print a single integer, indicating the number of deals required to return the deck to its original order. Output each integer on its own line, with no extra spaces, and no blank lines between answers. All possible inputs yield answers which will fit in a signed 64-bit integer.

Sample Input 1

```
1 3
10 3
52 4
0 0
```

Sample Output 1

```
1
4
13
```