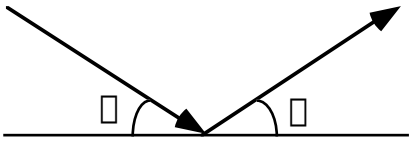
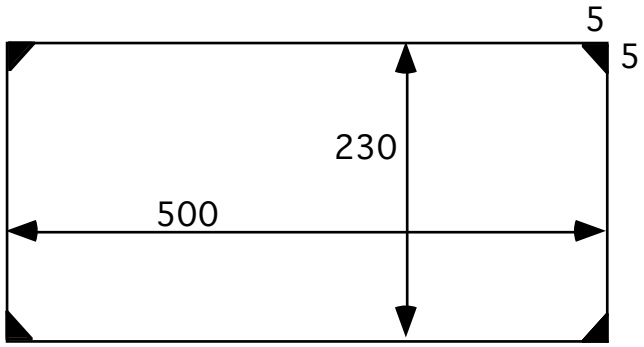
Problem S5
Follow the Bouncing Ball

Suppose you want to create a screen saver which has a ball bouncing around the edges. The action will continue until the ball hits a corner (like a pool table). At that point, the ball disappears off the screen and the screen goes black.

Balls bounce off the walls according to the laws of physics. The angle of the ball as it approaches the wall (the angle of incidence) will equal the angle of the ball as it bounces away (the angle of reflection).



Write a program which will determine how many bounces it will take before the bouncing ball will be swallowed by a corner, if at all. The screen dimensions are in the range [100, 1000] units. Consider the ball as a point on the screen. The corner pockets are 5 units along each wall, and if a ball hits the wall at the edge of a pocket it will continue bouncing.
A sample screen is shown below.



Input will consist of four integers, each on a separate line.

n – the width of the screen, $100 \le n \le 1000$
m – the height of the screen, $100 \le m \le 1000$
p – the initial position of the ball at the bottom of the screen, $5 \le p \le n - 5$
q – the position of the ball when it bounces off the right wall, $5 \le q \le m - 5$

The output consists of an integer which indicates the number of bounces the ball makes before it disappears in a corner. Count the hit off the right wall as the first bounce. It is possible that the ball will never bounce into a corner (assuming a frictionless surface), in this case, output should be 0. Note that when the ball hits a pocket is not counted as a bounce. You do not have to program the actual animation.

Sample Input (Input file : ball.in)

```
300
200
200
100
```

Output for Sample Input (Output file : ball.out)

```
2
```

Graeme,

This note was included in the contest package.

Note to Canadian Computing Competition Supervisors:

Please inform students writing the **Senior Division** of the CCC that in question **S5**,
p is the distance measured from the left side of the rectangle, and
q is the distance measured from the bottom of the rectangle.