

Czech Technical University in Prague
ICPC Foundation
Czech ACM Chapter

Central Europe Regional Contest 2019



Crimson Sexy Jalapeños

`crimson.c`, `crimson.cpp`, `Crimson.java`, `crimson.py`

The central piece of the Tainted chocolate game is a classic chocolate bar divided into square pieces by a rectangular grid of grooves parallel to the sides of the bar. Some squares have been tainted with extremely bitter substance that makes the tainted square (nearly) indigestible.

The game is played by two players who alternate in their moves. In a valid move, one player is obliged to consume some part of the chocolate bar. It is allowed to divide the current chocolate bar along one of the grooves into two smaller bars and then eat just one of them. The player who consumes a bar containing at least one tainted square loses the game.

The positions of all tainted squares are known at the beginning of the game. All other squares are safe to eat. Each player tries to avoid eating a bar of chocolate containing one or more tainted squares, because when this happens, the player involuntarily makes their personal most disgusted grimace becoming a source of great amusement not only to the other player but also to other people watching the game.

In this problem, you are to write a program to play the Tainted chocolate game. We neglect the part of the code which simulates players grimaces and chocolate consumption, and instead focus only on the winning moves.

A valid move is described by a directional string and a positive integer X. The directional string is one of the four strings "`top`", "`bottom`", "`left`", or "`right`". The description means that the bar is divided by the X-th groove, counted from that side of the currently remaining bar which is specified by the given directional string. The player then consumes the part on that side.

Input Specification

The first line of input contains integers R, C, K ($1 \leq R, C \leq 10^4; 1 \leq K \leq 100$). R is the number of rows, C is the number of columns, and K is the number of tainted squares in the chocolate bar. Each of the next K lines contains two integer values A and B ($1 \leq A \leq R, 1 \leq B \leq C$), the coordinates of one tainted square. The coordinates of the top-left corner square are $(1, 1)$.

The rest of the input depends on your output. For each of your valid moves, there will appear one input line containing either an opponent's valid move description or the string "`yuck!`" which indicates the opponent lost the game. In the latter case your program should terminate.

Output Specification

After reading the chocolate bar description, you may decide whether you want to start the game. If you want to play second, print one line containing the string "`pass`". This special string may only appear on the very first line of your output.

Then, for each of your moves, print one line containing a description of a valid move. Your program will be given a *Wrong Answer* if it produces anything else than a sequence of valid moves leading to the victory.

After printing each move description, flush the output buffer. For example, you may use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, or `stdout.flush()` in Python.

Sample Input 1

```
4 6 2
2 3
4 4
```

```
right 2
```

```
yuck!
```

Output for Sample Input 1

```
top 1
```

```
left 2
```

For the purpose of clarity, the above data are interleaved to illustrate the order of interaction between your program and the system. Note that there will be no empty lines in real data and there must not be any empty lines in your output.

Sample Input 2

```
3 5 1
2 3
left 1
left 1
right 1
yuck!
```

Output for Sample Input 2

```
pass
right 1
top 1
bottom 1
```

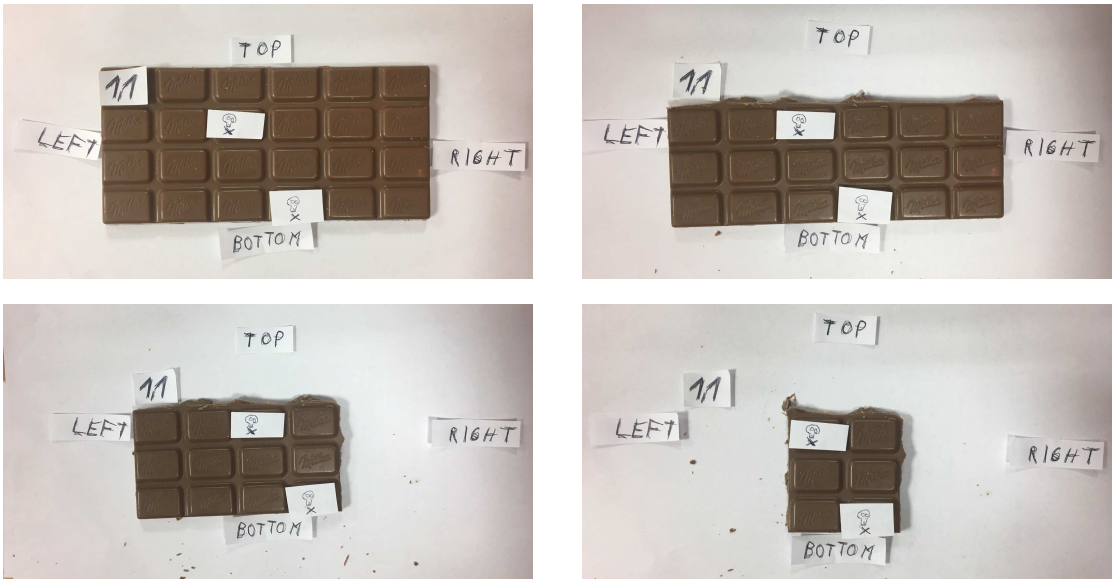


Figure 1: Illustration of the game from Sample Input 1