Turtle Graphics Sabotage   P62974_en
Dinovè Concurs de Programació de la UPC - Final (2021-09-22)

One of the most overlooked cases of animal abuse of the last fifty years has been the large scheme of reptile exploitation staged in the name of computer science education efforts. Under the direction of the likes of Fally Weurzeig, Peymour Sapert, and Synthia Colomon, millions or turtles have been subject to cruel and inhuman treatment since the seventies. Acronyms like OGOL and CISAB have hidden complex programs that put them into forced labor and made them endlessly obey idiotic instructions such as `fd 10` and `rt 90`, or even the utterly incomprehensible `bm100,100u10c1r20g10c2l10`.

However, it is even less known that a group of four of those turtles once tried to break from the programming tyranny. Their names were Degas, Manet, Monet and Renoir, and they managed to sabotage their masters' plans over multiple years. Unfortunately, they were eventually discovered by their evil masters and turned into fake turtle soup. But it is our duty today to type `g++` to pay respect to these forgotten heroes of the digital age. . .

Let's focus on the problem: We have n turtles that will move in order on a 500×500 grid of integer coordinates (x, y), where x is the horizontal dimension. Each turtle will start at some initial position, and will move according to some instructions (*direction*, *length*), where *direction* is 'u' (up, negative y), 'd' (down, positive y), 'l' (left, negative x) or 'r' (right, positive x), and *length* is the number of steps to take in that direction.

Degas et al. will attach torches to some of their colleagues before they start their workday. The path of a turtle with a torch attached will be impassable to later turtles, or even to itself. Each turtle will fully execute its instructions before the next one starts, and it will stop right before it should enter a cell on fire. No further movement will be done by that turtle after that point. A turtle will not even visit its starting cell if it is on fire.

For instance, if the first turtle starts at (30, 10) with instructions `r 10 u 5 l 5 d 6`, it will visit $(30, 10)$, $(40, 10)$, $(40, 5)$, $(35, 5)$ and $(35, 11)$, as well as all the intermediate cells along those segments. But, if the turtle has a torch attached, it will stop at $(35, 9)$, just before visiting $(35, 10)$ for a second time, and therefore will neither move to $(35, 10)$ nor $(35, 11)$.

Can you decide which turtles attach torches to, so as to prevent as many turtles as possible from completing their paths? For instance, this corresponds to the first sample input:
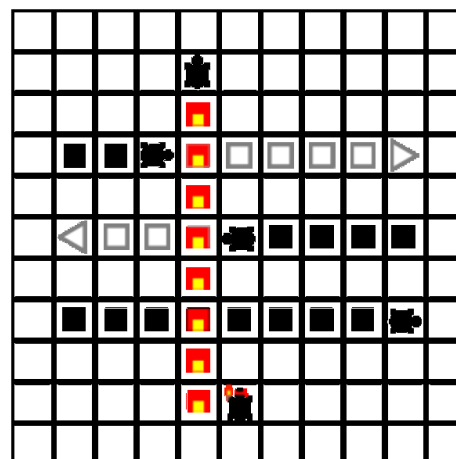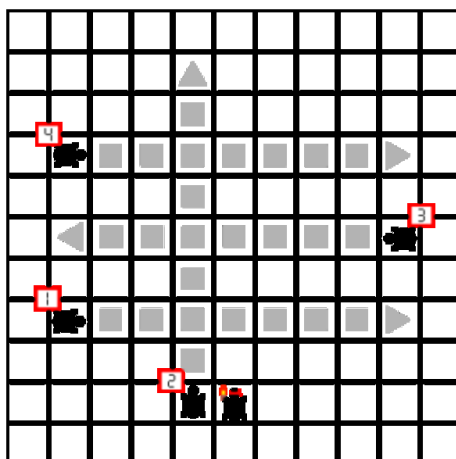
Input

Input consists of several cases. Every case starts with an n between 1 and 16, followed by n lines, each one with the path for turtle i: the initial coordinates (x_i, y_i), both between 0 and 499, a number of instructions m_i between 0 and 100, and the m_i instructions. No given instruction will ever ask a turtle to move off the grid.

Output

For every case, print the minimum number of turtles that can complete their instructions, together with the number of torches needed. If there are multiple ways to minimize the number of turtles, you must also minimize the number of torches.

Hint

A too simple brute-force solution may be too slow for this problem.

Sample input

```
4
1 7  1  r 8
4 9  1  u 8
9 5  1  l 8
1 3  1  r 8
3
1 7  1  r 8
9 5  1  l 8
1 3  1  r 8
2
23 42  0
23 42  0
1
499 499  4  l 499  u 499  r 499  d 499
8
0 0  1  r 10
3 0  2  d 10  r 20
7 0  1  d 15
0 3  1  r 3
0 7  1  r 3
10 3  1  l 3
10 7  1  l 3
0 12  1  r 100
```

Sample output

```
2 1
3 0
1 1
0 1
3 2
```

Problem information

Author : Edgar Gonzalez
Generation : 2024-05-02 20:52:56