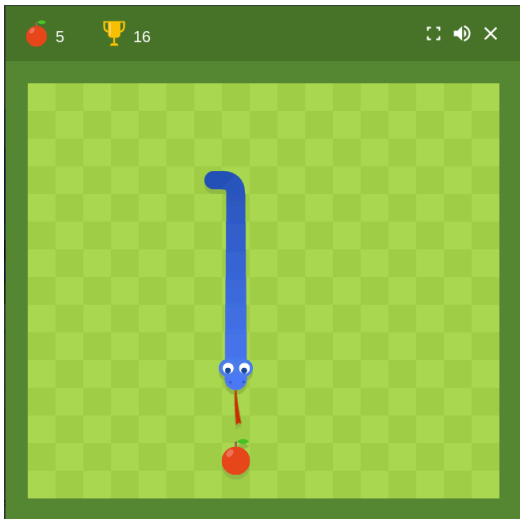
Snake



For the parameters given, the problem can be solved by exploring all possible positions that are reachable from the initial position of the snake.

Each position can be encoded using L+1 parameters by specifying the position of the head, the relative position of the neck (the second snake piece following the head), and then a series of left, right or straight turns. For instance, the following position:

	0	1	2	3	4	5
0						
1		0				
2		1	2	3		
3				4		
4				5	6	
5						

Could be encoded as

((1, 1), Down, Left, Straight, Right, Straight, Left)

Therefore, an upper bound on the number of possible states is $4mn3^{L-2}$. (mxn possible positions for the head, Up/Down/Right/Left for the neck, and then Left/Straight/Right for the remainder of the snake.) This is a theoretical maximum of ~191M states. In addition, we can use the following observation to shorten the search: If the snake ever manages to enter a square previously occupied by its body, it will be able to get to the apple because it has found a circular path on which it can move and then take a turn straight to the apple.

Therefore, this problem can be solved by exploring all possible states using a breadth-first search with a reasonably efficient encoding of the snake's position.

Be aware that the snake cannot run backwards, this input should yield 0 (no).

```
3 1
0
1
A
```