Walking Robots

locked

by AvmnuSng

We define the following conditions for n robots numbered from 0 to $n-1$:

- The robots are initially spaced apart on an infinitely long straight line. Some of them begin moving simultaneously at the same constant speed. Two robots moving in the same direction never collide.
- Two robots moving toward each other will crash, break down, and stay at that point forever. This counts as 2 collisions.
- A robot moving toward a non-moving robot will crash into it, break down, and stay at that point forever. This counts as 1 collision.

To clarify, whenever a moving robot crashes onto another robot, it will stay at that point and become a non-moving robot at that location.

We describe their movement instructions in a string, s, consisting of the letters `l`, `r`, and `d`. The i^{th} character in s denotes the i^{th} robot's instruction according to the following rules:

- `l`: Move *left* indefinitely until a collision occurs.
- `r`: Move *right* indefinitely until a collision occurs.
- `d`: Do not move at all.

Complete the function below so that it returns the total number of collisions that eventually occur.

Input Format

The first line contains an integer, q, denoting the total number of queries (i.e., calls to the function).
Each of the q subsequent lines describes a query in the form of a string, s, consisting of the letters `l`, `r`, and `d`.

Constraints

- $1 \leq q \leq 100$
- $1 \leq n \leq 10^5$, where n is the length of s.
- Each s consists of the letters `l`, `r`, and `d` only.

Output Format

Return an integer denoting the total number of collisions that eventually occur after movement begins.

Sample Input 0

```
5
r
lrrl
rrrll
rrdlldrr
rrrdllrllrrl
```

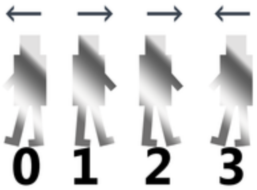
Sample Output 0

```
0
3
5
4
11
```

Explanation 0

We perform the first four of the $q = 5$ queries like so:

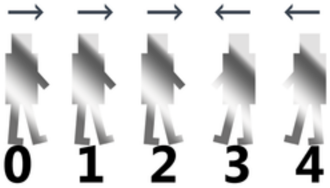
1. There is one robot who will move right indefinitely and never collide with any other robot, so we return 0.
2. There are $n = 4$ robots with the following movement instructions:



- Robots 2 and 3 walk into each other and stop. This counts as two collisions.
- Robot 1 walks into robot 2 (who is stopped) and stops. This counts as one collision.

We then return the total number of collisions, which is $2 + 1 = 3$.

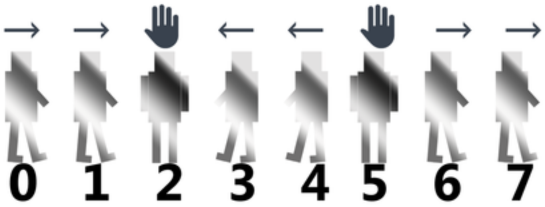
3. There are $n = 5$ robots with the following movement instructions:



- Robots 2 and 3 walk into each other and stop. This counts as two collisions.
- Robot 1 walks into robot 2 (who is stopped) and stops. This counts as one collision.
- Robot 4 walks into robot 3 (who is stopped), and stops. This counts as one collision.
- Robot 0 walks into robot 1 (who is stopped), and stops. This counts as one collision.

We then return the total number of collisions, which is $2 + 1 + 1 + 1 = 5$.

4. There are $n = 8$ robots with the following movement instructions:



- Robot 1 walks into robot 2 (who is stopped) and stops. This counts as one collision.
- Robot 3 walks into robot 2 (who is stopped) and stops. This counts as one collision.
- Robot 0 walks into robot 1 (who is stopped) and stops. This counts as one collision.
- Robot 4 walks into robot 3 (who is stopped) and stops. This counts as one collision.

Note that robots 2 and 5 do not move, so they cannot collide with anything. We then return the total number of collisions, which is $1 + 1 + 1 + 1 = 4$.

Submissions: 946

Max Score: 30
Difficulty: Medium

Rate This Challenge:
★★★★★

More

Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

    static int howManyCollisions(string s){
        // Complete this function
          int sum = 0;
            string nueva = s.Replace("rl", "d");
            sum += ((s.Length - nueva.Length) * 2);

            string estado = "";

            while (estado != nueva)
            {
                estado = nueva;
                string anterior = nueva;
                nueva = anterior.Replace("rl", "d");
                sum += ((anterior.Length - nueva.Length) * 2);

                anterior = nueva;
                nueva = nueva.Replace("rd", "d");
                sum += anterior.Length - nueva.Length;

                anterior = nueva;
                nueva = nueva.Replace("dl", "d");
                sum += anterior.Length - nueva.Length;


            }
            //Console.WriteLine(sum);
            return sum;
    }

    static void Main(String[] args) {
        int q = Convert.ToInt32(Console.ReadLine());
        for(int a0 = 0; a0 < q; a0++){
            string s = Console.ReadLine();
            // Returns the number of times moving robots collide.
            int result = howManyCollisions(s);
            Console.WriteLine(result);
        }
    }
}

```

Line: 33 Col: 24

Upload Code as File □ Test against custom input

Run Code Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature