Question 1: *Promenade Fractions*

A *promenade* is a way of uniquely representing a fraction by a succession of "left or right" choices. As successive choices are made the value of the promenade changes by combining the values of the promenade before the most recent left choice with the value before the most recent right choice.

If the value before the most recent left choice was l/m and the value before the most recent right choice was r/s then the new value will be $(l+r) / (m+s)$. If there has never been a left choice we use $l=1$ and $m=0$; if there has never been a right choice we use $r=0$ and $s=1$.

Fractions are **always** used in their lowest form; recall that a/b is in its lowest form if it is not possible to divide a and b by a common factor. Values generated by the formula will automatically be in their lowest form. Fractions are allowed to have a larger than b.

We will write our promenades as a sequence of Ls (for left choices) and Rs (for right choices).

For example, to form the promenade LRLL (using ∅ to represent the promenade before any choices are made):

- The value of ∅ is (1+0)/(0+1) = 1/1;
- The value of L is 1/2. Before the most recent left choice we bhad ∅ (= 1/1). There has not yet been a right choice, so we use $r=0$ and $s=1$. So the value of L is (1+0)/(1+1) = 1/2;
- LR = 2/3 as we use the values of ∅ (before the left choice) and L (before the right choice);
- LRL = 3/5 as we use the values of LR and L;
- LRLL = 4/7 as we use the values of LRL and L.

1(a) [23 marks]

Write a program which reads in a promenade of between 1 and 10 (inclusive) uppercase letters (each `L` or `R`).

You should output the promenade's value as a fraction in its lowest form.

Sample run 1

```
LRLL
4 / 7
```

Sample run 2

```
RL
3 / 2
```

1(b) [2 marks]

What is LRL + LLLL as a promenade?

1(c) [3 marks]

How many Ls and how many Rs does the promenade representing 1 / 1,000,000 contain?

1(d) [3 marks]

Does any promenade represent a negative fraction? Justify your answer.