

Southeastern European Regional Programming Contest
Bucharest, Romania
October 12, 2013

Problem F
Dragon Pattern

Input File: F.IN
Output File: standard output
Program Source File: F.C, F.CPP, F.JAVA

The dragon curve is a polygonal chain with segments of length 1, defined recursively. We choose some point on the plane and one of four directions, collinear to the coordinate axes. The left(right) dragon of order n is constructed in the following way:

- if n is equal to 0, we construct a segment of length 1 from the current point in the current direction, moving to endpoint;
- otherwise we construct the left dragon of order n−1 from the current point in the current direction, turn 90 degrees to the left(right) in the endpoint, and construct the right dragon of order n−1;

Let us construct the left dragon of order n, starting from the origin (point (0,0)) in the positive direction of the axis OX.

Given some pattern as a sequence of directions, in which sequential movements by unit length is performed, your task is to determine how many times does the given pattern occurs in the constructed dragon curve.

Input
Input file contains integer n ($0 \le n \le 10^9$) and string S consisting of symbols R, L, U and D, which defines the pattern. R denotes rightward movement (positive direction of the axis OX), L denotes leftward movement (negative direction of the axis OX), U denotes upward movement (positive direction of the axis OY), D denotes downward movement (negative direction of the axis OY). The length of the string S is in the range from 1 to 10^6.

Output
Output number of given patterns in the left dragon of the order n, constructed in the positive direction of an axis OX, modulo 10^9+7.

Sample input	Sample output
2 RU	1
3 L	3
4 LDL	2