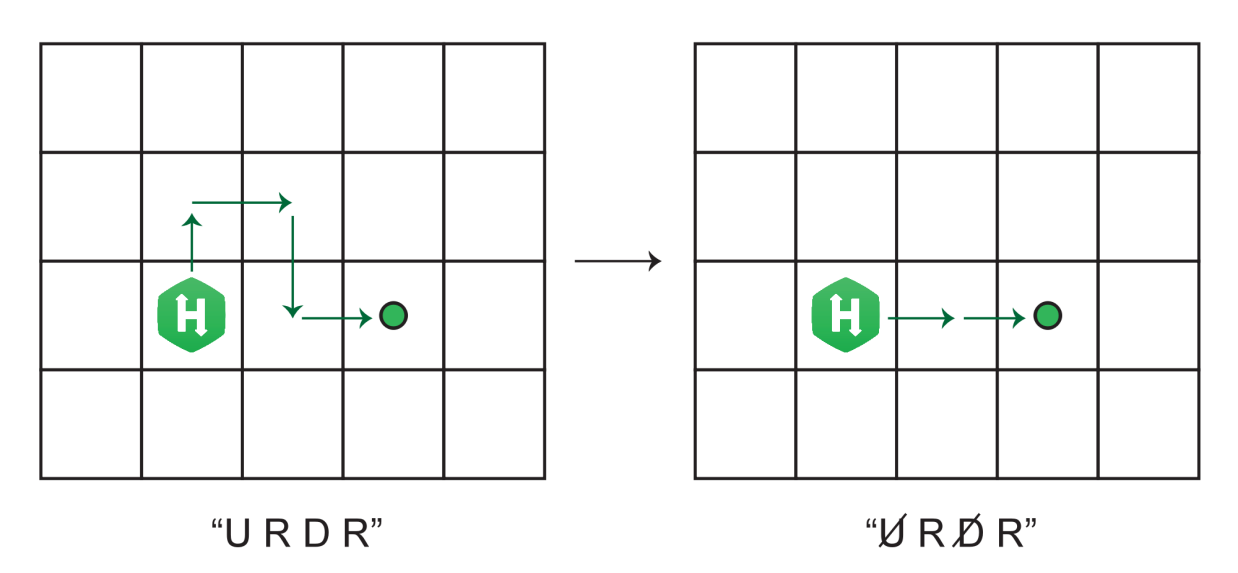
Character Reprogramming

A control device has 4 buttons that can be used to move a character around a screen in 4 directions: Up (U), Down (D), Left (L), and Right (R). The movement needs to be optimized by deleting unnecessary instructions while maintaining the same destination. Given the original set of instructions, what is the maximum number that can be deleted and still have the character reach the destination?

****Note:**** The instructions that are deleted do not need to be contiguous.

**Example**

*s = 'URDR'*



Given an original set of instructions *s = 'URDR',* the final destination is 2 units to the right of the initial position after the character moves up, right, down, and right. If '*U*' and '*D*' are deleted, the destination remains the same. The answer 2 will be returned.

**Function Description**

Complete the function *getMaxDeletions* in the editor below.

getMaxDeletions has the following parameter:

    string *s:*  the original instructions that were programmed

Returns:

*int*: the maximum number of instructions that can be deleted from *s* while maintaining the destination

**Constraints**

1 ≤ *n* ≤ 105

*s* contains only the characters 'U', 'D', 'L', and 'R'.

Input Format For Custom Testing

The first line contains a string, *s*, denoting the instructions.

Sample Case 0

**Sample Input**

STDIN Function

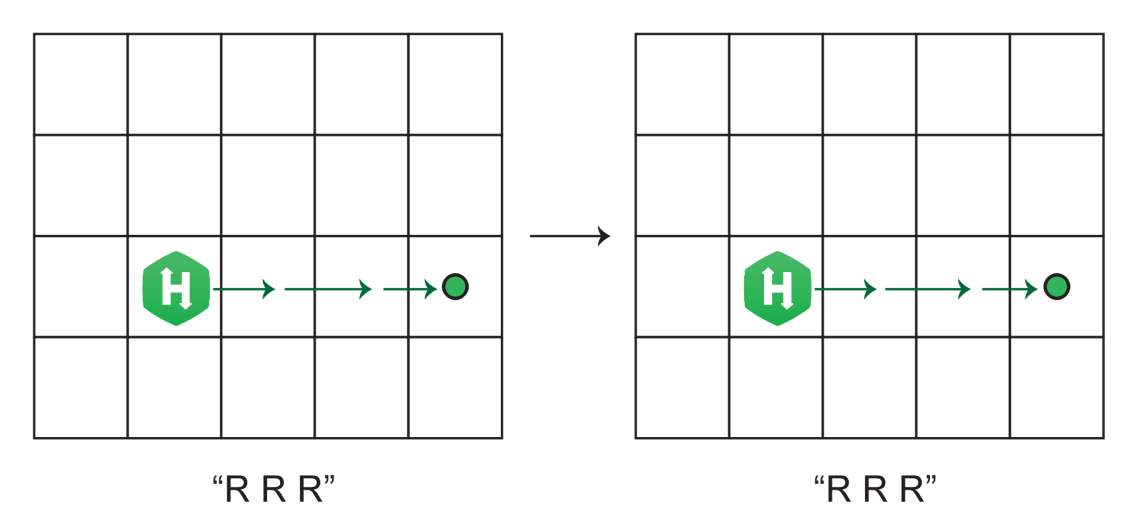
----- -----

RRR → s = 'RRR'

**Sample Output**

0

**Explanation**



There is nothing that can be deleted from these instructions, so the answer is 0.

Sample Case 1

**Sample Input**

STDIN Function

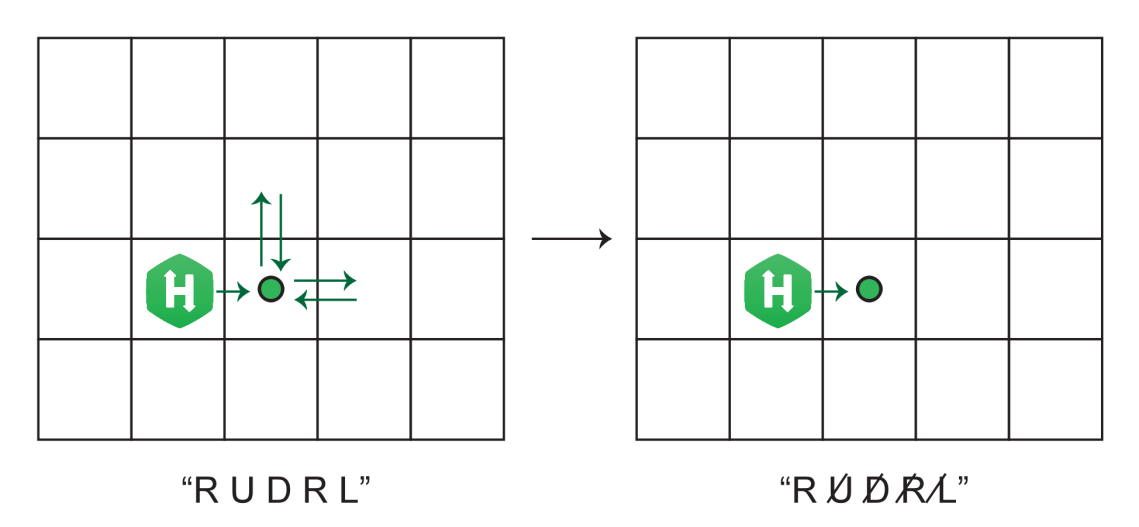
----- -----

RUDRL → s = 'RUDRL'

**Sample Output**

4

**Explanation**



The final destination is 1 unit to the right of the initial position. Deleting the final 2 instructions, '*RL*', will not affect the destination. Deleting *'UD'* also will not affect the destination. Because the goal is to delete the maximum number of movements, both of these instances, '*UDRL*', should be deleted, which is a total of 4 deletions.

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.function.\*;

import java.util.regex.\*;

import java.util.stream.\*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;

class Result {

/\*

\* Complete the 'getMaxDeletions' function below.

\*

\* The function is expected to return an INTEGER.

\* The function accepts STRING s as parameter.

\*/

public static int getMaxDeletions(String s) {

// Write your code here

}

}

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

String s = bufferedReader.readLine();

int result = Result.getMaxDeletions(s);

bufferedWriter.write(String.valueOf(result));

bufferedWriter.newLine();

bufferedReader.close();

bufferedWriter.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*