Gary is an avid hiker. He tracks his hikes meticulously, paying close attention to small details like topography. During his last hike he took exactly ***n*** steps. For every step he took, he noted if it was an *uphill*, ***U***, or a *downhill*, ***D*** step. Gary's hikes start and end at sea level and each step up or down represents a ***1*** unit change in altitude. We define the following terms:

A *mountain* is a sequence of consecutive steps *above* sea level, starting with a step *up* from sea level and ending with a step *down* to sea level.

A *valley* is a sequence of consecutive steps *below* sea level, starting with a step *down* from sea level and ending with a step *up* to sea level.

Given Gary's sequence of *up* and *down* steps during his last hike, find and print the number of *valleys* he walked through.

For example, if Gary's path is ***s = [DDUUUUDD]***, he first enters a valley ***2*** units deep. Then he climbs out an up onto a mountain ***2*** units high. Finally, he returns to sea level and ends his hike.

Function Description

Write down the function *countingValleys*  with any language you like. It must return an integer that denotes the number of valleys Gary traversed.

countingValleys has the following parameter(s):

*n*: the number of steps Gary takes

*s*: a string describing his path

Input Format

The first line contains an integer ***n***, the number of steps in Gary's hike.  
The second line contains a single string ***s***, of ***n*** characters that describe his path.

Constraints



Output Format

Print a single integer that denotes the number of valleys Gary walked through during his hike.

Sample Input

8

UDDDUDUU

Sample Output

1

Explanation

If we represent \_ as sea level, a step up as /, and a step down as \, Gary's hike can be drawn as:

\_/\ \_

\ /

\/\/

He enters and leaves one valley.