**Problem: Transforming Quxes**

**Scenario**

On a mysterious island, there are creatures known as Quxes which come in three colors: red (R), green (G), and blue (B). One power of the Qux is that if two of them are standing next to each other, they can transform into a single creature of the third color. Given N Quxes standing in a line, determine the smallest number of them remaining after any possible sequence of such transformations.

**Problem Statement**

Given an array of N Quxes standing in a line, return the smallest number of Quxes remaining after any possible sequence of transformations.

**Input Format**

* First line contains an integer N (the number of Quxes).
* Second line contains a string of length N, where each character is either 'R', 'G', or 'B'.

**Constraints**

* 1 <= N <= 100

**Output Format**

* An integer representing the smallest number of Quxes remaining.

**Example**

**Example 1:**

Input:

5

RGBGB

Output:

1

Explanation:

One possible sequence of transformations is as follows:

- ['R', 'G', 'B', 'G', 'B'] -> (R, G) -> B -> ['B', 'B', 'G', 'B']

- ['B', 'B', 'G', 'B'] -> (B, G) -> R -> ['B', 'R', 'B']

- ['B', 'R', 'B'] -> (R, B) -> G -> ['B', 'G']

- ['B', 'G'] -> (B, G) -> R -> ['R']

**Example 2:**

Input:

3

RRR

Output:

3

Explanation:

No transformations are possible as all Quxes are of the same color.

**Solution Approach**

The solution involves simulating the transformations. Since the problem allows for any sequence of transformations, we can utilize a greedy approach by iteratively combining adjacent Quxes of different colors until no further transformations are possible. The minimum number of Quxes remaining will be either 1 or a number greater than 1 if no transformations are possible.

Here is the Python code implementing this approach:

python

def min\_remaining\_quxes(quxes):

# Helper function to get the third color

def third\_color(c1, c2):

colors = {'R', 'G', 'B'}

colors.remove(c1)

colors.remove(c2)

return colors.pop()

# Convert the list to a mutable list of characters

quxes = list(quxes)

# Process the transformations

i = 0

while i < len(quxes) - 1:

if quxes[i] != quxes[i + 1]:

new\_color = third\_color(quxes[i], quxes[i + 1])

quxes = quxes[:i] + [new\_color] + quxes[i + 2:]

i = max(i - 1, 0)

else:

i += 1

return len(quxes)

# Reading input

n = int(input().strip())

quxes = input().strip()

# Computing the result

result = min\_remaining\_quxes(quxes)

# Printing the output

print(result)

This code reads the input values, processes the transformations, and prints the smallest number of Quxes remaining. The helper function third\_color determines the resulting color from a pair of different colors. The main function min\_remaining\_quxes processes the transformations by iterating through the list and applying the transformation rules until no more transformations can be performed.

**Extra Test Cases**

**Test Case 1:**

Input:

1

R

Output:

1

Explanation:

There is only one Qux, so no transformations are possible.

**Test Case 2:**

Input:

2

RG

Output:

1

Explanation:

One possible transformation is:

- ['R', 'G'] -> (R, G) -> B -> ['B']

**Test Case 3:**

Input:

4

RRGG

Output:

2

Explanation:

One possible transformation sequence is:

- ['R', 'R', 'G', 'G'] -> (R, G) -> B -> ['R', 'B', 'G']

No further transformations are possible.

**Test Case 4:**

Input:

6

RGBRGB

Output:

2

Explanation:

One possible transformation sequence is:

- ['R', 'G', 'B', 'R', 'G', 'B'] -> (R, G) -> B -> ['B', 'B', 'R', 'G', 'B']

- ['B', 'B', 'R', 'G', 'B'] -> (R, G) -> B -> ['B', 'B', 'B', 'B']

- ['B', 'B', 'B', 'B'] -> No more transformations possible.

**Test Case 5:**

Input:

5

GGGGG

Output:

5

Explanation:

All Quxes are of the same color, so no transformations are possible.

**Test Case 6:**

Input:

7

RGBGRBR

Output:

1

Explanation:

One possible transformation sequence is:

- ['R', 'G', 'B', 'G', 'R', 'B', 'R'] -> (R, G) -> B -> ['B', 'B', 'G', 'R', 'B', 'R']

- ['B', 'B', 'G', 'R', 'B', 'R'] -> (G, R) -> B -> ['B', 'B', 'B', 'B', 'R']

- ['B', 'B', 'B', 'B', 'R'] -> (B, R) -> G -> ['B', 'B', 'B', 'G']

- ['B', 'B', 'G'] -> (B, G) -> R -> ['B', 'R']

- ['B', 'R'] -> (B, R) -> G -> ['G']

**Test Case 7:**

Input:

8

RRRRGGGG

Output:

4

Explanation:

No transformations are possible as all Quxes are grouped by their color.

Top of Form

Bottom of Form