It's Not Easy Being Green

DiPS CodeJam 24

Prompt

'It's not easy being green,' was the frequent refrain of Kermit the frog.
'It seems you blend in
With so many other ordinary things,'
he says. Let's find him some friends, shall we? Each input is a string of length n made up of numbers and uppercase letters. Your task is to find the number of 6-character substrings which, when read as a hex colour code, are shades of green. Substrings may overlap with each other.
To determine whether a substring is a shade of green or not, check for the following conditions to be true:

- the amount of blue in the colour is less than 80% of the amount of green
- the amount of red in the colour is less than 64% of the amount of green

Given a string of length n, how many substrings can you find that satisfy the conditions?

Input Format

- The first line of the input contains the integer n.
- The second line of the input contains a string of length n containing numbers and uppercase letters.

Output Format

The first and only line of your output must contain a single integer m, denoting the the number of substrings that satisfy the conditions.

Constraints

- $10^4 \leq n \leq 10^6$

Solution

- Start a counter at 0 to count the number of substrings that satisfy the condition
- Iterate through every 6-letter substring
- Check if the substring is in hexadecimal
- If the substring is hexadecimal and satisfies the condition, increment the counter

Sample Program

```
def solve(s):
  m = 0
  for i in range(len(s)-5):
    substring = s[i:i+6]
    if not all(c in string.hexdigits for c in substring):
      break

    red, green, blue = [int(substring[j:j+2], 16) for j in range(0, 6, 2)]

    if blue < 0.8*green and red < 0.64*green:
      m+=1

  return m
```