



Practice

Compete

Jobs

Rank

Leaderboard



bor0s

Dashboard &gt; Algorithms &gt; Strings &gt; Beautiful Binary String

Badge Progress



Points: 1001.20 Rank: 26277

# Beautiful Binary String



by Shafaet

Problem

Submissions

Leaderboard

Discussions

Editorial

Alice has a **binary string**,  $B$ , of length  $n$ . She thinks a binary string is beautiful if and only if it doesn't contain the **substring** "010".

In one step, Alice can change a 0 to a 1 (or vice-versa). Count and print the minimum number of steps needed to make Alice see the string as beautiful.

## Input Format

The first line contains an integer,  $n$  (the length of binary string  $B$ ).  
The second line contains a single binary string,  $B$ , of length  $n$ .

## Constraints

- $1 \leq n \leq 100$
- Each character in  $B \in \{0, 1\}$ .

## Output Format

Print the minimum number of steps needed to make the string beautiful.

## Sample Input 0

```
7
0101010
```

## Sample Output 0

```
2
```

## Sample Input 1

```
5
01100
```

## Sample Output 1

```
0
```

## Sample Input 2

```
10
0100101010
```

## Sample Output 2

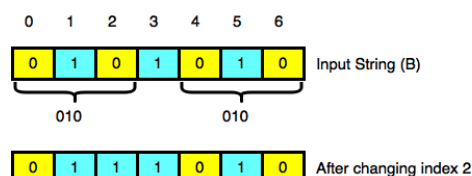
```
3
```

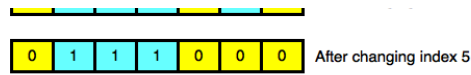
## Explanation

### Sample Case 0:

In this sample,  $B = "0101010"$

The figure below shows a way to get rid of each instance of "010":





Because we were able to make the string beautiful by changing **2** characters ( $B_2$  and  $B_6$ ), we print **2**.

**Sample Case 1:**

In this sample  $B = "01100"$

The substring **"010"** does not occur in  $B$ , so the string is already beautiful and we print **0**.

[f](#) [t](#) [in](#)

Submissions: **18500**

Max Score: 20

Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)

C++

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int minSteps(int n, string B){
5     int numsteps = 0;
6     std::string::size_type i;
7
8     while ((i = B.find("010")) != std::string::npos) {
9         //cout << B << " -> ";
10        B[i+2] = '1';
11        //cout << B << endl;
12        numsteps++;
13    }
14    return numsteps;
15    // Complete this function
16 }
17
18 int main() {
19     int n;
20     cin >> n;
21     string B;
22     cin >> B;
23     int result = minSteps(n, B);
24     cout << result << endl;
25     return 0;
26 }
27
```

Line: 1 Col: 1

[Upload Code as File](#) [Test against custom input](#)

[Run Code](#)

[Submit Code](#)

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)