***Maximise Binary String***

C371\_Coding\_September2022

**Topic** : String

**Difficulty Level :** Easy

**Question / Problem Statement** :

Martha gave a string **S** of length **N** to Clark, each of whose characters is either “0”, “1”, or “\*”.

Now Clark’s task is to replace every “\*” present in **S** with either “0” or “1”, such that the (absolute) difference between the number of 1s and the number of 0s present in it of the resulting binary string is maximised.

Write a program to print the maximised absolute difference between the number of 1s and the number of 0s present in the resulting binary string.

**Note**

**N** is always positive.

**Function Description**

In the provided code snippet, implement the provided **maximiseString(...)** method using the variables to print the maximised absolute difference between the number of 1s and the number of 0s present in the resulting binary string. You can write your code in the space below the phrase **“WRITE YOUR LOGIC HERE”**.   
  
There will be multiple test cases running so the Input and Output should match exactly as provided.  
The base Output variable **result** is set to a default value of **-404** which can be modified. Additionally, you can add or remove these output variables.

**Input Format**

First line contains an integer **N**.

Second line contains a string **S**.

**Sample Input**

4 –denotes N.

\*101 –denotes S.

**Constraints**

1 <= **N** <= 1000.

**S** always contains either “0”, “1”, or “\*”.

**Output Format**

Output should return the maximised absolute difference between the number of 1s and the number of 0s present in the resulting binary string.

**Sample Output**

2

**Explanation**

For string S = “\*101”, if we replace the “\*” with “0”, the string becomes “0101” and the absolute difference between the number of 1s and the number of 0s is 2 - 2 = 0.

If we replace the “\*” with “1”, the string becomes “1101” and the absolute difference between the number of 1s and the number of 0s is 3 - 1 = 2, which is maximum.

**Solution Steps**

1. We need to find the maximum absolute difference between the number of 1s and the number of 0s after replacing every “\*” present in string either with 1s or 0s.

2. Replacement of “\*” depends on the count of 1s and 0s as we need maximum absolute difference, so we need to replace every “\*” whose count is greater.

3. If 1s count is greater than 0s count then replace every “\*” with “1” and similarly replace every “\*” with “0” if 0s count is greater than 1s as we need maximum absolute difference and if both 1s and 0s count are the same then replace every “\*” with either “0” or “1”.

4. Finally return the maximum absolute difference.

**Running Solution in C++** :

#include <bits/stdc++.h>

using namespace std;

int main(){

//Declaring variable N and S.

int N;

string S;

//taking input N.

cin>>N;

//taking input S.

cin>>S;

//Declaring count of 1s, 0s and \* variable

int cnt\_one=0,cnt\_zero=0,cnt\_star=0;

//iterating the string.

for(int idx=0;idx<N;idx++){

//finding zero count.

if(S[idx]=='0')

cnt\_zero++;

//finding one count.

else if(S[idx]=='1')

cnt\_one++;

//finding star count.

else

cnt\_star++;

}

//Declaring max absolute difference variable.

int mxm\_abs\_diff;

//if 1s count and 0s count are the same, replace every star either with 0 or 1.

if(cnt\_one==cnt\_zero)

mxm\_abs\_diff=cnt\_star;

//if 1s count is greater than 0s count replace every star with 1 for maximised difference.

else if(cnt\_one>cnt\_zero)

mxm\_abs\_diff=(cnt\_one+cnt\_star)-cnt\_zero;

//if 0s count is greater than 1s count replace every star with 0 for maximised difference.

else

mxm\_abs\_diff=(cnt\_zero+cnt\_star)-cnt\_one;

cout<<mxm\_abs\_diff<<"\n";

return 0;

}

Input:

6

\*\*1000

Output:

4

**Test Cases [ Qty: 12 ]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case No** | **Input** | **Output** | **Score** |
| 1 | 4  \*101 | 2 | 0 |
| 2 | 6  \*\*1000 | 4 | 0 |
| 3 | 1  \* | 1 | 1 |
| 4 | 25  1110011010\*\*\*\*1\*0\*0110\*10 | 9 | 1 |
| 5 | 1000  01\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*101\*\*1010\*1 | 400 | 1 |
| 6 | 15  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | 15 | 1 |
| 7 | 100  1010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010 | 0 | 1 |
| 8 | 50  1111010110\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | 44 | 1 |
| 9 | 32  11111111111111111111111111111111 | 32 | 1 |
| 10 | 40  0000000000000000000000000000000000000000 | 40 | 1 |
| 11 | 75  101010111111111011111111101111111110111111111011111111101111111110111111111 | 57 | 1 |
| 12 | 8  101010\*\* | 2 | 1 |

Plagiarism found – No

Clarity of the problem statement - Yes

Clarity of the example in the problem statement - Yes

Clarity of sample test cases - Yes

Clarity of test cases (Dual output) – Yes

Clarity of explanations - Yes

Provided Solution running – Yes

EEOC complaint (using abusive words/Indian Names/) - No

Similar Question in System - No

Difficulty Level – Easy

Question w.r.t strings concepts- Yes

Final Comment: **Accepted**