# A. Prefixes

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nikolay got a string $s$ of **even** length $n$, which consists only of lowercase Latin letters 'a' and 'b'. Its positions are numbered from $1$ to $n$.

He wants to modify his string so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'. To achieve that, Nikolay can perform the following operation arbitrary number of times (possibly, zero): choose some position in his string and replace the letter on this position with the other letter (i.e. replace 'a' with 'b' or replace 'b' with 'a'). Nikolay can use no letters except 'a' and 'b'.

The prefix of string $s$ of length $l$ ($1 \le l \le n$) is a string $s[1..l]$.

For example, for the string $s$ ="abba" there are two prefixes of the even length. The first is $s[1 \dots 2]$ ="ab" and the second $s[1 \dots 4]$ ="abba". Both of them have the same number of 'a' and 'b'.

Your task is to calculate the minimum number of operations Nikolay has to perform with the string $s$ to modify it so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'.

## Input

The first line of the input contains one **even** integer $n$ $(2 \le n \le 2 \cdot 10^5)$ — the length of string $s$.

The second line of the input contains the string $s$ of length $n$, which consists only of lowercase Latin letters 'a' and 'b'.

## Output

In the first line print the minimum number of operations Nikolay has to perform with the string $s$ to modify it so that every its prefix of **even** length has an equal amount of letters 'a' and 'b'.

In the second line print the string Nikolay obtains after applying all the operations. If there are multiple answers, you can print any of them.

### Examples

| input | Copy |
|---|---|
| 4<br>bbbb | |

| output | Copy |
|---|---|
| 2<br>abba | |

| input | Copy |
|---|---|
| 6<br>ababab | |

| output | Copy |
|---|---|
| 0 | |

```
ababab
```

**input** Copy

```
2
aa
```

**output** Copy

```
1
ba
```

### Note

In the first example Nikolay has to perform two operations. For example, he can replace the first 'b' with 'a' and the last 'b' with 'a'.

In the second example Nikolay doesn't need to do anything because each prefix of an even length of the initial string already contains an equal amount of letters 'a' and 'b'.