Finest Final String

FWMC receives a string s of length n, consisting of only characters `A` and `B`, as a birthday gift. The identity of the person who keeps gifting FWMC strings as birthday gifts remains a mystery...

FWMC then starts playing around with the string. In each step, FWMC can perform the following compression operation on the string any number of times:

- Choose a position i such that $1 \leq i < n$ and $s_i = s_{i+1}$. Then delete two characters s_i and s_{i+1}, and insert one character of the opposite type at the deleted position (`B` if $s_i =$ `A`, and vice versa).

FWMC thinks that the smaller the lexicographical order[1] of a string, the **finer** it is. Your task is to help them find the **finest** string s' that can be obtained from string s, and one possible sequence of compression operations to obtain this string.

Input

Each test contains multiple test cases. The first line contains the number of test cases τ ($1 \leq \tau \leq 5\,000$). The description of the test cases follows.

- The first line contains an integer n ($1 \leq n \leq 5\,000$) — the length of the string s.
- The second line contains the string s. This string only contains characters `A` and `B`.

It is guaranteed that the sum of n over all test cases does not exceed $5\,000$.

Output

For each test case, print the following:

- The first line contains a string s' — the **finest** string that can be obtained from the string s.
- The second line contains an integer k ($0 \leq k < n$) — the number of compression operations.
- The third line contains k integers $p_1, p_2, \ldots, p_k$ representing k operations. The value p_i means that the i-th operation is to remove two characters at positions p_i and p_i+1, then insert a new character at position p_i. The value p_i must be less than the length of s before this operation.

If there are multiple solutions, you can output any of them.

[1] A string a is lexicographically smaller than string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$; or
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Sample Input 1	Sample Output 1
3 1 B 3 BAA 5 ABBBA	B 0 A 2 2 1 AA 3 3 3 2

Sample Explanation

In the first test case, it is impossible to perform any compression operation on the given string. Therefore, the **finest** string obtainable is 'B'.

In the second test case, one possible way to obtain the string 'A' (which can be shown to be the **finest** string obtainable) is as follows:

$$\mathtt{BAA} \rightarrow \mathtt{BB}$$
$$\mathtt{BB} \rightarrow \mathtt{A}$$

In the third test case, one possible way to obtain the string 'AA' (which can be shown to be the **finest** obtainable string) is as follows:

$$\mathtt{ABBBA} \rightarrow \mathtt{ABAA}$$
$$\mathtt{ABAA} \rightarrow \mathtt{ABB}$$
$$\mathtt{ABB} \rightarrow \mathtt{AA}$$