CodeForces Problem

March 09, 2024

B. Reverse Sort

Constriants

Time Limit 2 seconds **Memory Limit** 256 MB

Problem Statement

Ashish has a binary string s of length n that he wants to sort in non-decreasing order.

He can perform the following operation: Choose a subsequence of any length such that its elements are in non-increasing order. Formally, choose any k such that $1 \leq k \leq n$ and any sequence of k indices $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ such that $s_{i_1} \geq s_{i_2} \geq \ldots \geq s_{i_k}$. Reverse this subsequence in-place. Formally, swap s_{i_1} with s_{i_k}, swap s_{i_2} with $s_{i_{k-1}}$, ... and swap $s_{i_{\lfloor k/2 \rfloor}}$ with $s_{i_{\lceil k/2 \rceil + 1}}$ (Here $\lfloor x \rfloor$ denotes the largest integer not exceeding x, and $\lceil x \rceil$ denotes the smallest integer not less than x)

Find the minimum number of operations required to sort the string in non-decreasing order. It can be proven that it is always possible to sort the given binary string in at most n operations.

Input Description

The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \leq n \leq 1000$) — the length of the binary string s.

The second line of each test case contains a binary string s of length n containing only 0s and 1s.

It is guaranteed that the sum of n over all test cases does not exceed 1000.

Output Description :

For each test case output the following: The minimum number of operations m in the first line ($0 \leq m \leq n$). Each of the following m lines should be of the form: $k\ i_1\ i_2\ ...\ i_k$, where k is the length and $i_1 < i_2 < ... < i_k$ are the indices of the chosen subsequence. For them the conditions from the statement must hold.

Examples

Input
3 7 0011111 5 10100 6 001000
Output
0 1 4 1 3 4 5 1 3 3 5 6

Note

In the first test case, the binary string is already sorted in non-decreasing order.
In the second test case, we can perform the following operation: $k = 4$: choose the indices $\{1, 3, 4, 5\}$
$\underline{1}\ 0\ \underline{1}\ \underline{0}\ \underline{0} \rightarrow \underline{0}\ 0\ \underline{0}\ \underline{1}\ \underline{1}$
In the third test case, we can perform the following operation: $k = 3$: choose the indices $\{3, 5, 6\}$
$0\ 0\ \underline{1}\ 0\ \underline{0}\ \underline{0} \rightarrow 0\ 0\ \underline{0}\ 0\ \underline{0}\ \underline{1}$