Call of Cthulhu

Dexter loves multitasking and whenever he and his sister chat, he encodes his sister's word W to a number. Encoding works like this:

Each distinct substring S in W has two attributes denotes as A and B. A represents the number O, middle index, for each repetition of S in W. Assume one of the repetition for some S starts at index l and ends at index r, inclusive. The number O is as follows.

$$l, r \in [1, 2, ..., n] \ \mathit{and} \ r \geq l$$

$$O = l + \lfloor (r - l)/2 \rfloor$$

B represents how many times each distinct substring S repeats in W.

Dexter creates an array D which contains every distinct S. While creating D, Dexter must obey the following two rules.

For any two substrings, S_i at index i and S_{i-1} at index i-1, S_i's attribute A must be greater or equal to S_{i-1}'s attribute A. If there exist two substring such that S_i's attribute A is equal to S_{i-1}'s attribute A, One with the greater attribute B must have higher index. If they have same attributes B order does not matter.

Dexter wants the output of array D, to do that he writes down B attribute of each S_i in D. If in the output the consecutive B attributes are same, they can be writen as only one B attribute. Print the shortest output possible.

Input Format

First line contains number N, length of the word W.
Second line contains one string word W. Word W contains only small letters of latin alphabet.

Constraints

$0 < N \leq 1000$

Output Format

Print the shortest output possible.Each B must be contiguous in the output.

Sample Input

3
cbc

Sample Output

12

Explanation

First index: 1, last index: 1 , Formula: $1 + \lfloor((1-1)/2)\rfloor$, middle index of this substring: 1 , substring: c , sum of middle indices until now: 1, number of repetition of current substring until now: 1

First index: 1 , last index: 2 , Formula: $1 + \lfloor((2-1)/2)\rfloor$, middle index of this substring: 1 , substring: cd , sum of middle indices until now: 1,number of repetition of current substring until now: 1

First index: 1 , last index: 3, Formula: $1 + \lfloor((3-1)/2)\rfloor$, middle index of this substring: 2 , substring: cdc , sum of middle indices until now: 2 , number of repetition of current substring until now: 1

First index: 2 , last index: 2 , Formula: $2 + \lfloor((2-2)/2)\rfloor$, middle index of this substring: 2 , substring: d , sum of middle indices until now: 2, number of repetition of current substring until now: 1

First index: 2 , last index: 3, Formula: $2 + \lfloor((3-2)/2)\rfloor$, middle index of this substring: 2 , substring: dc , sum of middle indices until now: 2, number of repetition of current substring until now: 1

First index: 3 ,last index: 3, Formula: $3 + \lfloor((3-3)/2)\rfloor$, middle index of this substring: 3, substring: c , sum of middle indices until now: 4,number of repetition of current substring until now: 2

Version of the output before zipping seperated with space: 1 1 1 1 2
Final version of the output, zipped and without whitespace: 12