

Problem H. Sasha And Swag Strings

Input file: *standard input*
Output file: *standard output*
Time limit: 15 seconds
Memory limit: 768 mebibytes

Hey you, swagger and [a bad person]! You [bothered] everybody. Shove your suffix tree in your [censored].

Well-wisher

Sasha enjoys strings and problems on string theory. Especially he loves suffix structures. As Sasha loves **TopForces** community he is going to write an entry named “*On suffix trees, part 37*”. Surely, he wants everybody to understand what the article says, so he decided to decorate it with some nice examples. After a recent contest on **TopForces**, Sasha achieved the rank “International swagger”, and now he wants to match it. To achieve this, Sasha wants to choose the swaggiest strings among some given set.

Of course, Sasha’s concept of string swagness is based on suffix structures. To be precise, on suffix trees. To calculate the swagness of a string, Sasha performs the following actions:

1. He constructs the compressed suffix tree of the string.

Let us recall that a suffix tree is the trie built from all suffixes of the string. A compressed suffix tree is the same trie in which consecutive edges are merged. See the notes section for an example.

2. On each edge of the compressed suffix tree, he counts the number of distinct non-empty substrings of the string which is written on the edge.

The swagness of the string is defined as the sum of the calculated values among all edges. Unfortunately, Sasha is not so good with the suffix trees as he wants you to think, and can’t handle the problem on his own. Help him!

Input

The first line contains an integer n ($1 \leq n \leq 10^5$).

Each of the next n lines contains a string. The i -th line contains a non-empty string s_i . Each string consists entirely of lowercase Latin letters. The total length of all strings s_i does not exceed $5 \cdot 10^5$.

Output

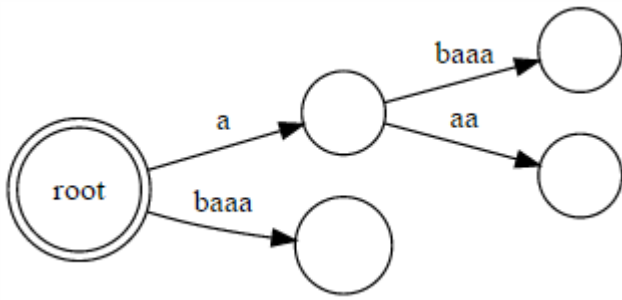
Print n integers, one per line. The i -th number must be the swagness of the string s_i .

Examples

standard input	standard output
3 umqra umnik merkurev	35 35 101
1 abaaa	17

Note

Compressed suffix tree for string *abaaa*:



Edges are a , aa , $baaa$ and $baaa$. They have 1, 2, 7 and 7 distinct non-empty substrings, respectively. So, the answer is $1 + 2 + 7 + 7 = 17$.