

---

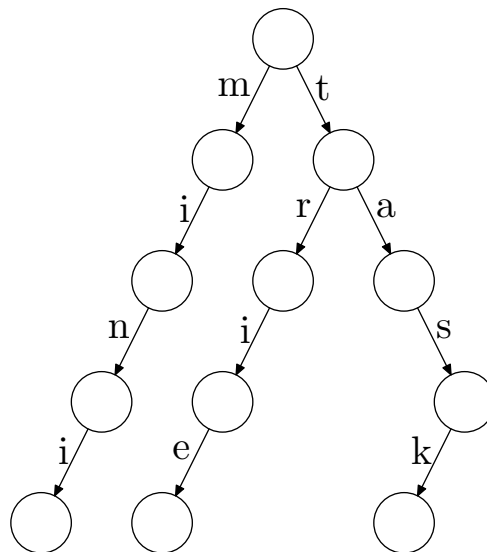
# Trie Minimization

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **1 second**  
Memory limit:         **512 megabytes**

Recently Innopolis University students have learnt data structures for searching strings, one of which was prefix tree or trie. Trie is a data structure that stores a set of strings as a rooted tree.

The tree has the following structure. Each edge of the tree is labeled with a single letter, there are no two edges labeled with the same letter that come out of each node. Each string can be read walking along some path from root to some vertex.

For instance, we can build a trie for strings “min”, “trie”, “task”, and “mini”, and it looks like this:



After the class professor gave the following homework: given a set of strings, your task is to replace some letters, so that the trie has the minimum number of nodes. What is the minimum number of replacements required for that? Students easily solved the task, can you do the same?

## Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 100\,000$ ) — the number of strings in the given set.

The following  $n$  lines contain strings from the set. Strings consist of lowercase English letters, the length of each string doesn't exceed 100 000.

The total length of all strings in the set doesn't exceed 1 000 000.

## Output

Print a single integer — the minimum number of replacements required to make a trie of minimum possible size.

## Scoring

Subtask	Score	Constraints
1	12	$n = 2$
2	14	$n = 3$
3	15	string lengths are 2
4	24	strings consist of letters 'a' and 'b'
5	35	no additional constraint

---

## Example

standard input	standard output
4 min trie task mini	8

## Explanation

In the given example one of the ways to solve the task is to make three replacements in “**trie**” and get “**mine**”, in “**task**” make four replacements and get “**mine**”, and in “**mini**” make one replacement and get “**mine**”. The total number of replacements done is 8.

This results in set: “**min**”, “**mine**”, “**mine**”, and “**mine**”.

The trie built for this set is on the picture, it consists of 5 nodes.

