



# Suffix Tries with Strings

locked

by pruthvishalcodi1

Problem

Submissions

Leaderboard

Discussions

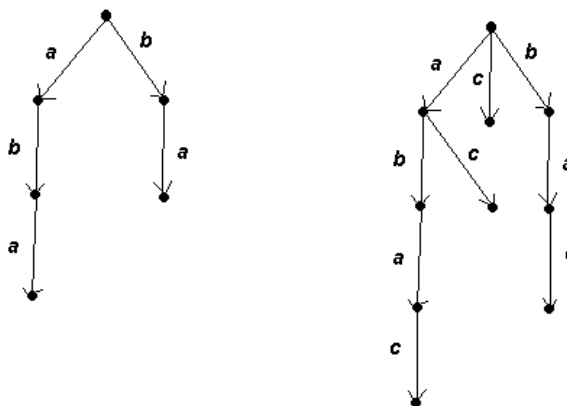
Editorial

Suffix trees are very powerful data structures. Using suffix trees it's often easy to solve the hardest computer science problems on strings. In this problem we'll consider a simplified version of suffix trees -- suffix tries (that is, tries formed by all suffixes of a given string).

Generally, the suffix tree is a tree whose edges are labeled with strings. Instead, we'll only consider suffix tries whose edges are labeled with single characters. This way every edge of a suffix tree may correspond to one or more edges connected in a chain-style fashion of a suffix trie.

Another trait of suffix trees is that each suffix of the string for which the suffix tree is built corresponds to exactly one path from the tree's root to a leaf. This is usually achieved by terminating each suffix of the string with a special character, say, \$. However, in suffix tries we won't do that. In particular, this implies that the number of leaves in a suffix trie can be smaller than the length of the original string.

In the picture the suffix tries for strings 'aba' and 'abac' are presented:



If we erase all letters from the edges of a suffix trie, it actually becomes a directed graph. Let's call two suffix tries equivalent if their corresponding directed graphs are isomorphic.

You're given a single string consisting of lowercase Latin letters. Your task is to find the number of different strings of the same length consisting of lowercase Latin letters as well such that their suffix tries are equivalent to the suffix trie of the given string. As this number can be large enough, output the remainder of its division by 42424242.

## Input Format

The first line contains a single number  $T$  -- the number of test cases (no more than 10). Each of the next  $T$  lines contains a single non-empty string of length no more than 100000 consisting of lowercase Latin letters  $a..z$ .

## Constraints

- $0 \leq T \leq 1000$
- $1 \leq \text{String\_length} \leq 10^3$

### Output Format

For each test case output just one line containing the sought number modulo 42424242.

### Sample Input 0

```
5
a
aa
abc
aba
helloworld
```

### Sample Output 0

```
26
26
15600
1300
6221124
```

### Explanation 0

In the first test case every string of length 1 satisfies the condition. In the second test case every string consisting of two equal letters is fine. In the third test case every string comprising three pairwise distinct letters adds 1 to the answer. In the fourth test case string 'bee' is one of the 1300 good strings.

[f](#) [t](#) [in](#)

Submissions: 3

Max Score: 18

Rate This Challenge:

☆☆☆☆☆

[More](#)

Admin Options

[Edit Challenge](#)

[View Submissions](#)

Current Buffer (saved locally, editable) [🔗](#) [🕒](#)

C++14



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8
9 int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
```

Line: 1 Col: 1

[Upload Code as File](#)

Test against custom input

Run Code

Submit Code

---

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)