



# Rename

locked

Problem

Submissions

Leaderboard

Discussions

Д-р М стартира собствен курс по програмиране, където показва различни бързи алгоритми. Той много държи да знае имената на своите курсисти и е решен да запомни всичките. За да оптимизира времето да им каже целите имена той прилага следната практика: всеки курсист има ново, съкратено име в курса, с което М ще се обръща към него. Имайки нареден списък с курсисти, името на даден курсист е най-краткия префикс, който не е префикс на друг курсист, вече срещнат в списъка. Вашата задача е да генерирате списъка с новите имена на курсистите.

Ако цялото име на курсист вече е срещнато, то новото му име е *старо\_име число* (разделени с интервал), показващо кой подред е със същото име.

## Input Format

Първия ред на входа е размера на списъка с курсисти N.

Следват N реда с първоначалните имена на курсистите.

## Constraints

$$1 \leq N \leq 10^5$$

Всяко име е до 10 символа

Всяко име е съставено от символи a-z

## Output Format

N реда, където на всеки ред е съкратеното име на съответния курсист

## Sample Input 0

```
3
alexander
alina
georgi
```

## Sample Output 0

```
a
ali
g
```

## Explanation 0

1. Тривиален случай - най-краткия префикс е 'a'
2. И двата префикса - 'a', 'al' са префикси на вече срещнат курсист, следващия най-кратък е 'ali'

## Sample Input 1

```
4
antoan
anton
antonina
anton
```

## Sample Output 1

```
a
anton
antoni
anton 2
```

## Explanation 1

1. Тривиален случай
2. Всичките префикси на 'anton' - 'a', 'an', 'ant', 'anto' са префикси на предишен курсист - съкратеното име е 'anton'
3. Аналогично на 2
4. Цялото име 'anton' вече сме го срещнали, съкратеното име е 'anton 2'

## Sample Input 2

```
3
ivan
i
i
```

## Sample Output 2

```
i
i
i 2
```

## Explanation 2

По горните изисквания, не е изключено двама студенти да имат едно и също съкратено име. Например при този вход, двама се казват i. Цялото име на вторият курсист е префикс на предишен, но е първият с това име, затова няма число, което да следва.

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



Submissions: 93

Max Score: 100

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)Current Buffer (saved locally, editable)  

C++14



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 #include <set>
7 #include <map>
8 using namespace std;
9
10 set<string> prefixes;
11 map<string, int> names;
12
13 int main() {
14     ios_base::sync_with_stdio(false);
15     cin.tie(NULL);
16
17     int n;
18     string name;
19
20     cin >> n;
21
```

```
22  for (int i = 0; i < n; i++) {
23      cin >> name;
24      string pref = "";
25
26      bool printed = false;
27      for (int j = 0; j < name.size(); j++) {
28          pref += name[j];
29
30          if (prefixes.count(pref) == 0 && !printed) {
31              //printf("%s\n", pref);
32              cout << pref << '\n';
33              printed = true;
34          }
35          prefixes.insert(pref);
36      }
37
38      if (names.count(name) == 0) {
39          //printf("%s\n", name);
40          if(!printed)
41              cout << name << '\n';
42          names[name] = 1;
43      }
44      else{
45          //printf("%s %d\n", name, names[name]);
46          names[name]++;
47          cout << name << ' ' << names[name] << '\n';
48      }
49  }
50  return 0;
51 }
52
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code