1077. Kuchiguse (20)

时间限制

100 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

HOU, Qiming

The Japanese language is notorious for its sentence ending particles. Personal preference of such particles can be considered as a reflection of the speaker's personality. Such a preference is called "Kuchiguse" and is often exaggerated artistically in Anime and Manga. For example, the artificial sentence ending particle "nyan~" is often used as a stereotype for characters with a cat-like personality:

 Itai nyan~ (It hurts, nyan~)

 Ninjin wa iyada nyan~ (I hate carrots, nyan~)

Now given a few lines spoken by the same character, can you find her Kuchiguse?

**Input Specification:**

Each input file contains one test case. For each case, the first line is an integer N (2<=N<=100). Following are N file lines of 0~256 (inclusive) characters in length, each representing a character's spoken line. The spoken lines are case sensitive.

**Output Specification:**

For each test case, print in one line the kuchiguse of the character, i.e., the longest common suffix of all N lines. If there is no such suffix, write "nai".

**Sample Input 1:**

3

Itai nyan~

Ninjin wa iyadanyan~

uhhh nyan~

**Sample Output 1:**

nyan~

**Sample Input 2:**

3

Itai!

Ninjinnwaiyada T\_T

T\_T

**Sample Output 2:**

nai

[提交代码](https://www.patest.cn/contests/pat-a-practise/1077)

水题……对于第一个字符串，从后往前获取子串，用子串去匹配其他字符串从后到前等长的子串，若不匹配则不为目标，则目标为上一个匹配出的结果，若之前一直没有匹配，则没有目标；

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#define MAX 505

using namespace std;

vector<string>all\_list;

int length=-1;

int compare(string target)

{

for (int i = 1; i < all\_list.size(); i++)

{

if (all\_list[i].substr(all\_list[i].size() - target.size()) != target)

return 0;

}

length = target.size();

return 1;

}

int main()

{

int num;

cin >> num;

getchar();

all\_list.resize(num);

for (int i = 0; i < num; i++)

getline(cin, all\_list[i]);

for (int i = all\_list[0].size()-1; i >= 0; i--)

{

if (compare(all\_list[0].substr(i)))

continue;

break;

}

if (length != -1)

cout << all\_list[0].substr(all\_list[0].size() - length);

else

cout << "nai";

}