#1061 : Beautiful String

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

描述

We say a string is beautiful if it has the equal amount of 3 or more continuous letters (in increasing order.)

Here are some example of valid beautiful strings: "abc", "cde", "aabbcc", "aaabbbccc".

Here are some example of invalid beautiful strings: "abd", "cba", "aabbc", "zab".

Given a string of alphabets containing only lowercase alphabets (a-z), output "YES" if the string contains a beautiful sub-string, otherwise output "NO".

输入

The first line contains an integer number between 1 and 10, indicating how many test cases are followed.

For each test case: First line is the number of letters in the string; Second line is the string. String length is less than 10MB.

输出

For each test case, output a single line "YES"/"NO" to tell if the string contains a beautiful sub-string.

提示

Huge input. Slow IO method such as Scanner in Java may get TLE.

**样例输入**

4

3

abc

4

aaab

6

abccde

3

abb

**样例输出**

YES

NO

YES

NO

嗯日常智障……

首先先读懂题意，该题目中，要判断字符串有没有这样一种子串，组成它的是三种相同数量的相邻字母组成的字符串。

然后以后不要被10MB之类的数据吓到了……拜托给你内存限制就有256mb了，区区10mb还能爆内存不成……

所以STL也是可以用的，没有必要用最原始的字符串处理。

主要题目解决就是分割相同字符子串，即把aabbcc分成aa,bb,cc这样，然后记录下每个字符串的内容，字符数量，存入STORE记录。

然后对于一个字符串，通过推理可得，三个符合连续标准的字符构成字符串，如果存在相等的可能的话，中间的字符得是三个字符串中最小的（或相等）以此为标准过一遍即可……。

#include <iostream>

#include<sstream>

#include<iomanip>

#include<string>

#include<vector>

#include<stack>

#include<algorithm>

#define hash 997

#define MAX 1000000000

#define ll long long

using namespace std;

struct substr

{

char content;

int num;

};

vector<substr>text;

string temp;

void count(string temp)

{

substr a;

a.content = temp[0];

a.num = temp.size();

text.push\_back(a);

}

int main()

{

int num;

int len;

cin >> num;

while (num--)

{

scanf("%d", &len);

getchar();

char former = getchar();

temp.push\_back(former);

for (int i = 1; i < len; i++)

{

char current = getchar();

if (current != former)

{

count(temp);

temp.clear();

temp.push\_back(current);

former = current;

}

else

temp.push\_back(current);

}

if (!temp.empty())

count(temp);

int flag = 0;

for (int i = 1; i < text.size() - 1; i++)

{

if ((text[i].content - 1) == text[i - 1].content && (text[i].content + 1) == text[i + 1].content)

{

if (text[i].num <= min(text[i - 1].num, text[i + 1].num))

{

flag = 1;

}

}

}

if (flag)

cout << "YES" << endl;

else

cout << "NO" << endl;

temp.clear();

text.clear();

}

}