**Question 1:**

In this challenge, you will be given a string. You must remove characters until the string is made up of any two alternating characters. When you choose a character to remove, all instances of that character must be removed. Your goal is to create the longest string possible that contains just two alternating letters.

As an example, consider the string abaacdabd. If you delete the character a, you will be left with the string bcdbd. Now, removing the character c leaves you with a valid string bdbd having a length of 4. Removing either b or d at any point would not result in a valid string.

Given a string , convert it to the longest possible string made up only of alternating characters. Print the length of string on a new line. If no string can be formed, print instead.

Function Description

Complete the alternate function in the editor below. It should return an integer that denotes the longest string that can be formed, or if it cannot be done.

alternate has the following parameter(s):

s: a string

Input Format

The first line contains a single integer denoting the length of .

The second line contains string .

Constraints

Output Format

Print a single integer denoting the maximum length of for the given ; if it is not possible to form string , print instead.

Sample Input

10

beabeefeab

Sample Output

5

Explanation

The characters present in are a, b, e, and f. This means that must consist of two of those characters and we must delete two others. Our choices for characters to leave are [a,b], [a,e], [a, f], [b, e], [b, f] and [e, f].

If we delete e and f, the resulting string is babab. This is a valid as there are only two distinct characters (a and b), and they are alternating within the string.

If we delete a and f, the resulting string is bebeeeb. This is not a valid string because there are consecutive e's present. Removing them would leave consecutive b's, so this fails to produce a valid string .

Other cases are solved similarly.

babab is the longest string we can create.

**Answer**

from itertools import combinations

n = int(input())

s = input()

unique = set(s)

delCount = len(unique) - 2

ans = 0

if(len(unique) > 1):

for i in combinations(unique, delCount):

temp = s

count = 0

for j in i:

temp = temp.replace(j, '')

if(all(temp[k]!=temp[k+1] for k in range(len(temp)-1))):

if(ans < len(temp)):

ans = len(temp)

print(ans)

else:

print(ans)

**Question 2**

Given an array of strings A[], determine if the strings can be chained together to form a circle. A

string X can be chained together with another string Y if the last character of X is same as first

character of Y. If every string of the array can be chained, it will form a circle.

For eg for the array arr[] = {"for", "crack", "rig", "kaf"} the answer will be Yes as the given strings can be chained as "for", "rig", "crack" and "kaf"

Input

The first line of input contains an integer T denoting the number of test cases. Then T test cases

follow.

The first line of each test case contains a positive integer N, denoting the size of the array.

The second line of each test case contains a N space seprated strings, denoting the elements of the

array A[].

Output

If chain can be formed, then print 1, else print 0.

Constraints

1 <= T <= 100

0 <= N <= 30

0 <= A[i] <= 10

Examples

Input

2

3

abc bcd cdf

4

ab bc cd da

Output

0

1

Explanation:

Testcase 1: Only strings "abc" and "cdf" follows the pattern chaining strings together but not "bcd" as none of other two strings begin with the character 'd'. Thus formation of circle of strings fails.

Testcase 2: Every string is given such that, first and last character follows chaining pattern to form circle of strings.

**Answer**

for i in range(int(input())):

n=int(input())

l=list(map(str,input().split()))

l=set(l)

x=[]

for i in l:

for j in l:

if i[-1]==j[0]:

x.append(i)

x.append(j)

if len(set(l))==len(set(x)):

print(1)

else:

print(0)