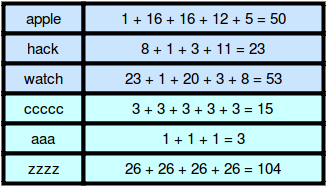
# Find Weight of Strings

The weight of a string is the sum of the weights of all the string's characters. For example:



**Super Reduced String**

**Sample Input 0**

aaabccddd

**Sample Output 0**

abd

**Explanation 0**

Steve performs the following sequence of operations to get the final string:

aaabccddd → abccddd → abddd → abd

**Sample Input 1**

aa

**Sample Output 1**

Empty String

**Explanation 1**

aa → Empty String

**Find All Valid Name and Phone Number**

**Print all valid phone number and name,**

**Constraints:**

* **Name must have minimum length 4**
* **name first character must be upper case**
* **name should no contain any numerical no**
* **phno length must equal to 10**

**Sample Input 0**

5

sottuneelam,9780123450

s,987987

sunflower4,8978979877

987987

mtr globjamoon,9879845646

**Sample Output 0**

Sottuneelam,9780123450

Mtr globjamoon,9879845646

**Explanation:**

first line contain size of array

from second line has input of name and phno separated by comma

sottuneelam length is greater than 4,9780123450 length is equal to 10, it is valid phno no

s,987987 name and phno no length is <4,<10 hence it is invalid

sunflower4 is onvalid name because it have numerical value 4, 8978979877is valid phno hence it is eliminated

987987 has only the phno which is invalid and doesnot contain name

mtr globjamoon length is greater than 4,9879845646 length is equal to 10, it is valid name and phno

# CamelCase

Alice wrote a sequence of words in [CamelCase](https://en.wikipedia.org/wiki/CamelCase) as a string of letters, , having the following properties:

* It is a concatenation of one or more *words* consisting of English letters.
* All letters in the first word are *lowercase*.
* For each of the subsequent words, the first letter is *uppercase* and rest of the letters are *lowercase*.

Given , print the number of words in  on a new line.

For example, . There are  words in the string.

**Sample Input**

saveChangesInTheEditor

**Sample Output**

5

**Explanation**

String  contains five words:

1. save
2. Changes
3. In
4. The
5. Editor

Thus, we print  on a new line.

# Caesar Cipher

Julius Caesar protected his confidential information by encrypting it using a cipher. [Caesar's cipher](https://en.wikipedia.org/wiki/Caesar_cipher) shifts each letter by a number of letters. If the shift takes you past the end of the alphabet, just rotate back to the front of the alphabet. In the case of a rotation by 3, w, x, y and z would map to z, a, b and c.

Original alphabet: abcdefghijklmnopqrstuvwxyz

Alphabet rotated +3: defghijklmnopqrstuvwxyzabc

**Sample Input**

11

middle-Outz

2

**Sample Output**

okffng-Qwvb

**Explanation**

Original alphabet: abcdefghijklmnopqrstuvwxyz

Alphabet rotated +2: cdefghijklmnopqrstuvwxyzab

m -> o

i -> k

d -> f

d -> f

l -> n

e -> g

- -

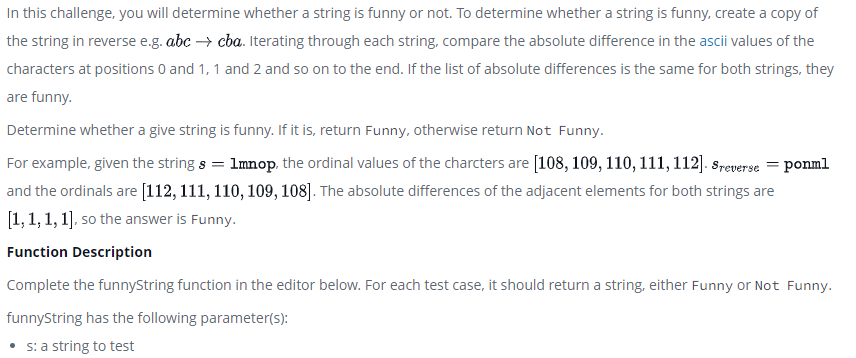
O -> Q

u -> w

t -> v

z -> b

# Funny String



**Mars Exploration**

Letters in some of the SOS messages are altered by cosmic radiation during transmission. Given the signal received by Earth as a string,*S* , determine how many letters of Sami's SOS have been changed by radiation.

For example, Earth receives SOSTOT. Sami's original message was SOSSOS. Two of the message characters were changed in transit.

**Sample Input 0**

SOSSPSSQSSOR

**Sample Output 0**

3

**Explanation 0**

S = **SOSSPSSQSSOR**, and signal length |S|=12 . Sami sent 4 SOS messages (i.e.: 12/3=4).

Expected signal: SOSSOSSOSSOS

Recieved signal: SOSSPSSQSSOR

Difference: X X X

We print the number of changed letters.

**Sample Input 1**

SOSSOT

**Sample Output 1**

1

**Explanation 1**

S = **SOSSOT**, and signal length |S|=6. Sami sent 4 SOS messages (i.e.: 6/3=2).

Expected Signal: SOSSOS

Received Signal: SOSSOT

Difference: X

We print the number of changed letters, which is .

**Sample Input 2**

SOSSOSSOS

**Sample Output 2**

0

**Explanation 2**

Since no character is altered, we print 0.

# Pangrams

Roy wanted to increase his typing speed for programming contests. His friend suggested that he type the sentence "The quick brown fox jumps over the lazy dog" repeatedly. This sentence is known as a pangram because it contains every letter of the alphabet.  
**Sample Input 0**

We promptly judged antique ivory buckles for the next prize

**Sample Output 0**

pangram

**Sample Explanation 0**

All of the letters of the alphabet are present in the string.

**Sample Input 1**

We promptly judged antique ivory buckles for the prize

**Sample Output 1**

not pangram

**Sample Explanation 0**

The string lacks an x.

**Good String or Not**

Given a string **S** of length **N**, you have to tell whether it is good or not. A good string is one where the distance between **every** two adjacent character is exactly 1. Here distance is defined by minimum distance between two character when alphabets from '**a**' to **'z'**are put in cyclic manner. For example distance between 'a' to 'c' is 2 and distance between 'a' to 'y' is also 2. The task is to print **"YES"** or **"NO"**(without quotes) depending on whether the given string is Good or not.

**Input:**  
First line of the input contains**T** denoting the number of test cases.Then **T** lines follow. Each line contains a string **S**.

**Output:**  
Print  the answer for each testcase in a separate line.

**Constraints:**

1≤**T**≤50  
1≤**|S|**≤50  
  
**Note: S** contains only lowercase alphabetic characters

**Input:**  
3  
aaa  
cbc  
bcd

**Output:**  
NO  
YES  
YES

# How Many Substrings?

input:

aabaa

 The substrings of abaa are a, b, ab, ba, aa, aba, baa, and abaa, so we print  8 on a new line.

# Strong Password

Louise joined a social networking site to stay in touch with her friends. The signup page required her to input a *name* and a *password*. However, the password must be *strong*. The website considers a password to be *strong* if it satisfies the following criteria:

* Its length is at least .
* It contains at least one digit.
* It contains at least one lowercase English character.
* It contains at least one uppercase English character.
* It contains at least one special character. The special characters are: !@#$%^&\*()-+

She typed a random string of length n in the password field but wasn't sure if it was strong. Given the string she typed, can you find the minimum number of characters she must add to make her password strong?

*Note*: Here's the set of types of characters in a form you can paste in your solution:

numbers = "0123456789"

lower\_case = "abcdefghijklmnopqrstuvwxyz"

upper\_case = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

special\_characters = "!@#$%^&\*()-+"

**Input Format**

The first line contains an integer  denoting the length of the string.

The second line contains a string consisting of  characters, the password typed by Louise. Each character is either a lowercase/uppercase English alphabet, a digit, or a special character.

**Constraints**

**Output Format**

Print a single line containing a single integer denoting the answer to the problem.

**Sample Input 0**

3

Ab1

**Sample Output 0**

3

**Explanation 0**

She can make the password strong by adding  characters, for example, $hk, turning the password into Ab1$hk which is strong.

 characters aren't enough since the length must be at least .

**Sample Input 1**

11

#HackerRank

**Sample Output 1**

1

**Explanation 1**

The password isn't strong, but she can make it strong by adding a single digit.