**Q1)**

Chayan and Shubham are best friends . They love to play with strings. One day , Chayan challenged Shubham to find out minimum no. of characters to be removed from strings to make it palindrome(shuffling of characters can be done) . Shubham tried to solve the problem but he was unable to solve the problem. Can you help him out in solving this problem.

**INPUT:-**

First line contains no. of testcases **T .**

For each test case you are given a string **S** (can be of different length **N**).

**NOTE :-** String contains only lowercase letters.

**OUTPUT:-**

For each test case output an integer which resembles the no. of removed characters.

**CONSTRAINTS:**

1<=T<=100

1<=N<=106

Time Limit: 1 sec

**SAMPLE INPUT:-**

3

aabbcd

abcd

aaabb

**SAMPLE OUTPUT:-**

**1**

**3**

**0**

**Explanation:-**

test case 1: input string : aabbcd

Either c or d can be removed to make it palindrome.

**Q2)**

At a railway station, we have time-table of trains arrival and departure. We need to find the minimum number of platforms so that all the trains can be accommodated as per their schedule.

**Input:**

First line contains no. of testcases T.

For each test case,first line contain integer n denoting the no. of trains

Second line contains the arrival time of trains (separated by spaces)

Third line contains the departure time of trains

Consider the time to be 24 hours format

**Output:**

Minimum number of platforms so that all the trains are accommodated

n

a1 a2   … an

b1 b2   … bn

**Constraints :**

1<=T<=100

1<n<1000

0000 <= ai <=2400

0000 <= bi <=2400

Time Limit: 2 sec

**SAMPLE INPUT:**

1

4

0100 0300 0005 0050

0200 0400 0100 0055

**SAMPLE OUTPUT**:

2

**Q3)**

Write an efficient program to find the largest sum of contiguous subarray within an one-dimensional array of integers. A contiguous subarray of an array is defined as the sequence of elements that are in any continuous set of indices that are valid within an array.

**Input Constraints**:

**First line contains no. of testcases** 1<=T<=100

For each test case,first line : array size (**N**), where 1<= **N**<=104

Second line : **N** integers separated by spaces

where each number in array **A[i]** satisfies

-10000 <= **A[i]**<=10000

Time Limit: 2sec

**Output Constraints**:

Single integer **SUM** which is the largest sum of all possible contiguous subarrays.

**SAMPLE INPUT:**

1

3

6 -4 5

**SAMPLE OUTPUT:**

7

**Explanation:**

array A[]=6,-4, 5. Possible contiguous subarray combinations are {6}, {-4}, {5}, {6,-4}, {-4,5} and {6,-4,5}. Note that {6,5} is not a valid subarray as the indices of 5 and 4 are not continuous.  
The contiguous subarray  {6,-4,5} has the largest sum 7.

**Q4)**

Kohli love the concepts of numerology. He wants to find out those numbers (in a given particular range) whose digit sum is equal to **K** where digit sum will be calculated until a single digit is formed.

For example, **K**=9 and the inputted number = 999, then first the sum of digits is calculated which is 9+9+9=27. As 27 is not a single digit, its digit sum is calculated i.e. 2+7=9. 9 is single digit then no need to calculate further .Match this number with **K**

**Input:**

The first line contains **T** denoting the number of test cases.

Each test case contains an integer **L , R** (both notifing the range L-R and are inclusive) and **K**.

**Output:**

Count the numbers in particular range(L-R) whose digit sum is equal to k.

**SAMPLE INPUT:**

1

10 30 1

**SAMPLE OUTPUT:**

3

**Explanation :**

Test case 1: L=10, R=30 and K=1 then total numbers in given range=10, 11,12.......,30 in which 10 whose digit sum=1 ,19 whose digit sum=10 again calculate digit sum=1+0=1 similarly 28 . so total count=3 (10,19,28).

**Q5)**

After years of study, scientists at **ISRO** have discovered an alien language transmitted from a faraway planet. The alien language is very unique in that every word consists of exactly **L** lowercase letters. Also, there are exactly **W** words in this language.

Once the dictionary of all the words in the alien language was built, the next breakthrough was to discover that the aliens have been transmitting messages to Earth for the past decade. Unfortunately, these signals are weakened due to the distance between our two planets and some of the words may be misinterpreted. In order to help them decipher these messages, the scientists have asked you to devise an algorithm that will determine the number of possible interpretations for a given pattern.

A pattern consists of exactly **L** tokens. Each token is either a single lowercase letter (the scientists are very sure that this is the letter) or a group of unique lowercase letters surrounded by parenthesis ( and ). For example: (ab)d(dc) means the first letter is either a or b, the second letter is definitely d and the last letter is either d or c. Therefore, the pattern (ab)d(dc) can stand for either one of these 4 possibilities: add, adc, bdd, bdc.

**Input**

The first line of input contains 3 integers, **L**, **D** and **N** separated by a space. **D** lines follow, each containing one word of length **L**. These are the words that are known to exist in the alien language. **N** test cases then follow, each on its own line and each consisting of a pattern as described above. You may assume that all known words provided are unique.

**Output**

For each test case, output

Case #**X**: **K**

where **X** is the test case number, starting from 1, and **K** indicates how many words in the alien language match the pattern.

**CONSTRAINTS:-**

1 ≤ **L**≤ 15  
1 ≤ **D** ≤ 5000  
1 ≤ **N** ≤ 500

**SAMPLE INPUT:-**  
 3 5 4  
abc  
bca  
dac  
dbc  
cba  
(ab)(bc)(ca)  
abc  
(abc)(abc)(abc)  
(zyx)bc

**SAMPLE OUTPUT:-**  
Case #1: 2  
Case #2: 1  
Case #3: 3  
Case #4: 0