1. In this challenge, the user enters a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.

**NOTE:** String letters are case-sensitive.

**Input Format**

The first line of input contains the original string. The next line contains the substring.

**Output Format**

Output the integer number indicating the total number of occurrences of the substring in the original string.

**Sample Input**

ABCDCDC

CDC

**Sample Output**

2

==================================================

2. Kevin and Stuart want to play the '**The Minion Game**'.

**Game Rules**  
Both players are given the same string S  
Both players have to make substrings using the letters of the string S

Stuart has to make words starting with consonants.  
Kevin has to make words starting with vowels.  
The game ends when both players have made all possible substrings.

**Scoring**  
A player gets +1 point for each occurrence of the substring in the string S

**For Example**:  
String S= BANANA  
Kevin's vowel beginning word = ANA  
Here, ANA occurs twice in BANANA. Hence, Kevin will get 2 Points.

**Function Description**

Complete the minion\_game in the editor below.

minion\_game has the following parameters:

* string string: the string to analyze

**Prints**

* string: the winner's name and score, separated by a space on one line, or Draw if there is no winner

**Input Format**

A single line of input containing the string S

**Note**: The string S will contain only uppercase letters:[A-Z]

**Sample Input**

BANANA

**Sample Output**

Stuart 12

================================================================

3. Given the names and grades for each student in a class of N students, store them in a nested list and print the name(s) of any student(s) having the second lowest grade.

**Note:** If there are multiple students with the second lowest grade, order their names alphabetically and print each name on a new line.

**Input Format**

The first line contains an integer,N, the number of students.  
The 2N subsequent lines describe each student over 2 lines.  
- The first line contains a student's name.  
- The second line contains their grade.

**Output Format**

Print the name(s) of any student(s) having the second lowest grade in. If there are multiple students, order their names alphabetically and print each one on a new line.

**Sample Input 0**

5

Harry

37.21

Berry

37.21

Tina

37.2

Akriti

41

Harsh

39

**Sample Output 0**

Berry

Harry

There are 5 students in this class whose names and grades are assembled to build the following list:

python students = [['Harry', 37.21], ['Berry', 37.21], ['Tina', 37.2], ['Akriti', 41], ['Harsh', 39]]

The lowest grade of37.2 belongs to Tina. The second lowest grade of 37.21 belongs to both Harry and Berry, so we order their names alphabetically and print each name on a new line.

4. The provided code stub will read in a dictionary containing key/value pairs of name:[marks] for a list of students. Print the average of the marks array for the student name provided, showing 2 places after the decimal.

**Input Format**

The first line contains the integer n , the number of students' records. The next n lines contain the names and marks obtained by a student, each value separated by a space. The final line contains **query\_name**, the name of a student to query.

**Output Format**

Print one line: The average of the marks obtained by the particular student correct to 2 decimal places.

**Sample Input 0**

3

“Krishna” [67, 68, 69]

“Arjun” [70, 98, 63 ]

“Malika” [52, 56, 60]

Malika

**Sample Output 0**

56.00

5. When users post an update on social media,such as a URL, image, status update etc., other users in their network are able to view this new post on their news feed. Users can also see exactly when the post was published, i.e, how many hours, minutes or seconds ago.

Since sometimes posts are published and viewed in different time zones, this can be confusing. You are given two timestamps of one such post that a user can see on his newsfeed in the following format:

Day dd Mon yyyy hh:mm:ss +xxxx

Here +xxxx represents the time zone. Your task is to print the absolute difference (in seconds) between them.

**Input Format**

The first line contains T, the number of testcases.  
Each testcase contains 2 lines, representing time t1 and time t2 .

**Output Format**

Print the absolute difference t-t2 in seconds.

**Sample Input 0**

2

Sun 10 May 2015 13:54:36 -0700

Sun 10 May 2015 13:54:36 -0000

Sat 02 May 2015 19:54:36 +0530

Fri 01 May 2015 13:54:36 -0000

**Sample Output 0**

25200

88200

**Explanation**

In the first query, when we compare the time in UTC for both the time stamps, we see a difference of 7 hours. which is 7\* 3600 seconds or 25200 seconds.

Similarly, in the second query, time difference is 5 hours and 30 minutes for time zone adjusting for that we have a difference of 1 day and 30 minutes. Or 24\*3600+30 \* 60 = 88200

6. Have the function LongestWord(**text**) take the **text** parameter being passed and return the largest word in the string. If there are two or more words that are the same length, return the first word from the string with that length. Ignore punctuation and assume **text** will not be empty.

#### Examples I/p "I love dogs" Output: love

7. Have the function ThreeSums(**arrs**) take the array of integers stored in **arrs**, and determine if any three distinct numbers (excluding the first element) in the array can sum up to the first element in the array.

For example: if **arrs** is [8, 2, 1, 4, 10, 5, -1, -1] then there are actually three sets of triplets that sum to the number 8: [2, 1, 5], [4, 5, -1] and [10, -1, -1].

Your program should return the string **true** if 3 distinct elements sum to the first element, otherwise your program should return the string **false**. The input array will always contain at least 4 elements.

#### Examples

 [10, 2, 3, 1, 5, 3, 1, 4, -4, -3, -2]  
Output: true

[12, 3, 1, -5, -4, 7]  
Output: false

8. Have the function QuestionsMarks(**str**) take the **str** string parameter, which will contain single digit numbers, letters, and question marks, and check if there are exactly 3 question marks between every pair of two numbers that add up to 10.

If so, then your program should return the string true,

otherwise it should return the string false. If there aren't any two numbers that add up to 10 in the string, then your program should return false as well.  
  
For example: if **str** is "arrb6???4xxbl5???eee5" then your program should return true because there are exactly 3 question marks between 6 and 4, and 3 question marks between 5 and 5 at the end of the string.

#### Examples

 "aa6?9"  
Output: false

 "acc?7??sss?3rr1??????5"  
Output: true

9. Have the function FirstReverse(**str**) take the **str** parameter being passed and return the string in reversed order. For example: if the input string is "Hello World and Coders" then your program should return the string sredoC dna dlroW olleH.

. Examples

 "coderbyte"  
Output: etybredoc

"I Love Code"  
Output: edoC evoL I

10 . You and Fredrick are good friends. Yesterday, Fredrick received N credit cards from **ABCD Bank**. He wants to verify whether his credit card numbers are valid or not. You happen to be great at regex so he is asking for your help!

A valid credit card from **ABCD Bank** has the following characteristics:

► It must start with a 4, 5or 6.  
► It must contain exactly 16 digits.  
► It must only consist of digits (0-9).  
► It may have digits in groups of 4 , separated by one hyphen **"****-****"**.  
► It must NOT use any other separator like ' ' , '\_', etc.  
► It must NOT have 4 or more consecutive repeated digits.

**Examples**:

**Valid Credit Card Numbers**

4253625879615786

4424424424442444

5122-2368-7954-3214

**Invalid Credit Card Numbers**

42536258796157867 #17 digits in card number → Invalid

4424444424442444 #Consecutive digits are repeating 4 or more times → Invalid

5122-2368-7954 - 3214 #Separators other than '-' are used → Invalid

44244x4424442444 #Contains non digit characters → Invalid

0525362587961578 #Doesn't start with 4, 5 or 6 → Invalid

**Input Format**

The first line of input contains an integer N.  
The next N lines contain credit card numbers.

**Output Format**

Print 'Valid' if the credit card number is valid. Otherwise, print 'Invalid'. Do not print the quotes.

**Sample Input**

6

4123456789123456

5123-4567-8912-3456

61234-567-8912-3456

4123356789123456

5133-3367-8912-3456

5123 - 3567 - 8912 - 3456

**Sample Output**

Valid

Valid

Invalid

Valid

Invalid

Invalid

11. You are given a string S and width w.  
Your task is to wrap the string into a paragraph of width w.

**Function Description**

Complete the wrap function in the editor.

wrap has the following parameters:

* string string: a long string
* int max\_width: the width to wrap to

**Returns**

* string: a single string with newline characters ('\n') where the breaks should be

**Sample Input**

ABCDEFGHIJKLIMNOQRSTUVWXYZ

4

**Sample Output**

ABCD

EFGH

IJKL

IMNO

QRST

UVWX

YZ