**★★☆☆☆**

**題組：基礎48題**

**題號：Q12414 - Calculating Yuan Fen**

**整理者：陳紫淇**

**學號：ADT105142**

**使用語言:JAVA**

**解題日期：2018年8月18日**



**題目:**

Yuanfen (http://en.wikipedia.org/wiki/Yuanfen) is a Chinese term that is hard to understand for people in other countries. Roughly speaking, yuanfen means the pre-determined “binding force” that links two people (usually two lovers) together. Although it is a blind faith, many people, especially girls like to calculate it. Unfortunately, my girlfriend is one of them. One day, she asked me, “Sweetie, shall we find out our yuanfen?” Oh, I really hate that question, but I cannot reject it... Luckily, I’m a programmer, so the only thing I need to do is to find a seemingly good algorithm and write a yuanfen calculator. After several hours’ searching in the web, I decided to implement the following popular yuanfen algorithm:

Step 1: Pick up the name abbreviations of the couple and concatenate them. For example, if the couple named Jiang Yun Fan and Tang Yu Rou, the concatenation of abbreviations is JYFTYR.

Step 2: Replace each letter with a number string. For some predefined positive integer ST, replace A with ST, and B with ST + 1, C with ST + 2, . . . , Z with ST + 25. For example, if ST = 81, A should be replaced with 81, B should be replaced with 82, . . . , Z will be replaced by 106. In the case above, JYFTYR will be replaced by 901058610010598.

Step 3: Repeat the following: add up each pair of consecutive digits, and write down the last digit of each sum. It’s not difficult to see that each time we perform this action, the number of digits is decreased by 1. When the number string is exactly 100, or has no more than 2 digits, the process ends. The current number is the yuanfen between the couple. In the case above, the process is as follows:

901058610010598

91153471011547

0268718112691

284589923850

02937815135

2120596648

332545202

65799722

1268694

384453

12898

3077

374

01

So if ST = 81, Jiang Yun Fan and Tang Yu Rou’s yuanfen is only 1! Too bad! I know my girlfriend very well. I know that even the result is as high as 99, she’ll still be unhappy. Could you find the value of ST such that the yuanfen between my sweetheart and I is 100?

**Input**

There will be at most 50 test cases. Each case contains a string of at least four and at most ten capital letters.

**Output**

For each test case, print the smallest positive integer ST (note that ST should not be zero). If it does not exist or larger than 10000, print a string ‘:(’ (without quotes). Disclaimer Don’t be sad if the result of you and your sweetie is larger than 10000. That’s no big deal.

**Sample Input**

JYFTYR

ABCDEF

YTHHLS

YTHLML

LYXM

JYFLY

CBTZX

LXYZLE

LXYLYR

QWERTY**Sample Output**

148

634

:(

910

96

4284

631

850

149

2277

**問題描述：**

[緣份](http://en.wikipedia.org/wiki/Yuanfen)是一個外國人難以理解的中文名詞。大致說來，緣份是一種冥冥中將兩人 (通常是情人) 結合的力量。僅管是種迷信，很多人——特別是女生——喜歡去計算它。

不幸地，我的女友也是這樣。有天，她問我：「甜心，可以算一下我們的緣份嗎？」唉，我真的很討厭這問題，但我無法拒絕。還好，我是個程式設計師，所以我只要找到一個看來不錯的演算法並寫成一個緣份計算器就可以了。在網路上搜尋了幾個小時後，我決定採用以下的緣份演算法：

第一步：取出姓名的縮寫並接在一起。例如，如果這對戀人叫 Jiang Yun Fan 和 Tang Yu Rou，他們的縮寫就是 JYFTYR。

第二步：將每個字母以數字字串取代。用 ST 來取代 A，ST+1 來取代 B，ST+2 來取代 C，......，ST+25 來取代 Z，其中 ST 為一個已知的正整數。 例如，如果 ST=81，A 就以 81 來取代，B 就以 82 來取代，......，Z 則以 106 來取代。上面的例子，JYFTYR 則以 901058610010598 來取代。

第三步：重覆以下動作：將相鄰的兩位數相加，並寫下和的個位數。不難發現這個動作每做一次，這個數字字串就會少一位數。當這個數字變成 100 或是不超過兩位數時，便停止這個程序。所得的數字便是兩人的緣份。以上面的例子來說，處理的過程如下：

901058610010598

91153471011547

0268718112691

284589923850

02937815135

2120596648

332545202

65799722

1268694

384453

12898

3077

374

01

所以如果 ST=81，Jiang Yun Fan 和 Tang Yu Rou 的綠份便只有 1。

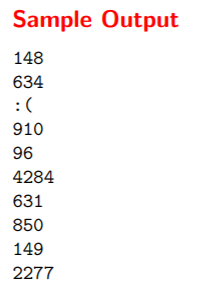
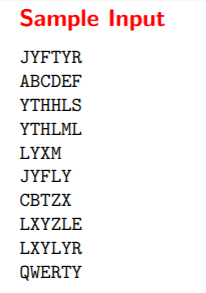
慘了，我很了解我的女友，我知道就算結果是 99 她仍然會不高興。你可以找到一個 ST 使得我和女友間的緣份會是 100 嗎？

**Input**

最多 50 筆測。每筆測資有一個含有最少四個最多十個大寫字母的字串。

**Output**

對於每筆測資，印出最小的正整數 ST (ST 不為零)。如果它不存在或是大於 10000，印出 **:(**  。



**解法:**

1. 將字母轉成ASCII，減去65+ST，做成題目要求的數字
2. 將數字拆成一個個傳入陣列儲存
3. 將陣列中相鄰的兩數相加，再拿其結果更新陣列(注意:要把最後一個數字刪除)
4. 重複3.直到陣列大小為3，判斷是否為100，若是則印出ST並跳出迴圈，否則將ST+1重複1.~4.
5. 若ST=10000仍無法做出100，印出:(

**解法範例：**

|  |
| --- |
| import java.util.Scanner;  import java.util.ArrayList;  public class Main  {  public static void main(String[] args)  {  Scanner sc = new Scanner(System.in);  String input;  int ST = 0;  ArrayList<Integer> myList = new ArrayList<>();  while (sc.hasNextLine())  {  input = sc.nextLine();  for (ST = 0; ST <= 10000; ST++)  {  myList.clear();  for (int i = 0; i < input.length(); i++)  {  int a0 = ((int) input.charAt(i)) - 65 + ST; // 將字串轉成數字(依照ST)  String a1 = String.valueOf(a0);  for (int j = 1; j < a1.length(); j++) // 將一個個數字存入陣列  myList.add(Integer.valueOf(a1.substring(j - 1, j)));  myList.add(Integer.valueOf(a1.substring(a1.length() - 1))); // 最後一個字  }  while (myList.size() > 3) // 將相鄰的兩位數相加，並寫下和的個位數  {  for (int k = 0; k <= myList.size() - 2; k++)  {  int b = myList.get(k) + myList.get(k + 1);  if (b >= 10)  myList.set(k, b % 10);  else  myList.set(k, b);  }  myList.remove(myList.size() - 1); // 刪除最後一個數字  }  if (myList.get(0) == 1 && myList.get(1) == 0 && myList.get(2) == 0) // 能做出100  {  System.out.println(ST);  break;  }  if (ST == 10000)  System.out.println(":(");  }  }  }  } |