2015/11/04

E541-1

程式能力檢定題庫
(CPE 版)

分級 1

494　Kindergarten Counting Game

分級：1　　　　　　**分類：字元與字串**

Everybody sit down in a circle. Ok. Listen to me carefully.

``Woooooo, you scwewy wabbit!''

Now, could someone tell me how many words I just said?

大家圍成一個圓圈坐好。好，要仔細聽我說喔。
``Woooooo, you scwewy wabbit!''
現在誰可以跟我說，我剛剛說了幾個字呢？

Input

Input to your program will consist of a series of lines, each line containing multiple words (at least one). A ``word'' is defined as a consecutive sequence of letters (upper and/or lower case).

輸入是由一系列的句子組成，每一句話都混雜著單字（最少一個），單字定義為由連續的字母（大小寫都可）所組成的。

Output

Your program should output a word count for each line of input. Each word count should be printed on a separate line.

你的程式要幫忙算一算每筆輸入有幾個字。每行只會有一個輸出值印出。

Sample Input

```
Meep Meep!
I tot I taw a putty tat.
I did! I did! I did taw a putty tat.
Shsssssssssh ... I am hunting wabbits. Heh Heh Heh Heh ...
```

Sample Output

```
2
7
10
9
```

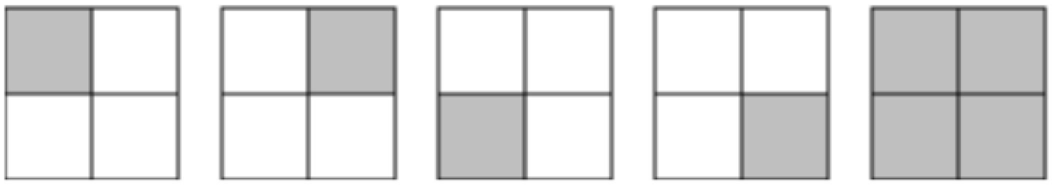
12149　Feynman

分級：1　　　　　　　　**分類：數學計算**

Richard Phillips Feynman was a well known American physicist and a recipient of the Nobel Prize in Physics. He worked in theoretical physics and also pioneered the field of quantum computing. He visited South America for ten months, giving lectures and enjoying life in the tropics. He is also known for his books "Surely You're Joking, Mr. Feynman!" and "What Do You Care What Other People Think?", which include some of his adventures below the equator.

His life-long addiction was solving and making puzzles, locks, and cyphers. Recently, an old farmer in South America, who was a host to the young physicist in 1949, found some papers and notes that is believed to have belonged to Feynman. Among notes about mesons and electromagnetism, there was a napkin where he wrote a simple puzzle: "how many different squares are there in a grid of N ×N squares?".

In the same napkin there was a drawing which is reproduced below, showing that, for N=2, the answer is 5.



費曼 (Richard Phillips Feynman) 是一個有名的美國物理學家及諾貝爾物理獎得主。他主攻理論物理並倡導量子電腦。他曾訪問南美十個月，在那兒演講並享受熱帶生活。他的成名作「別鬧了，費曼先生」及「你管別人怎麼想」中也包含了他在赤道以南的經歷。

他終生的嗜好是解決和建立謎題、鎖、及密碼。最近，一位曾在 1949 年接待這位年輕物理學家的南美老農夫找到一些屬於費曼的筆記。在這些有關介子及電磁學的筆記中，夾有一張餐巾紙，上寫有個簡單的謎題：「在一個 N × N 的方格中含有幾個不同的正方形？」

在下面同樣大小的紙上，重現了那餐巾紙上的圖，顯示了 N=2 時答案為 5。

Input

The input contains several test cases. Each test case is composed of a single line, containing only one integer N, representing the number of squares in each side of the grid (1 ≤ N ≤ 100).

The end of input is indicated by a line containing only one zero.

輸入有若干筆測資，每筆一行，內含著一個整數 N，代表方格的邊長 (1 ≤ N ≤ 100)。

輸入的結束以一行含有一個零來表示之。

Output

For each test case in the input, your program must print a single line, containing the number of different squares for the corresponding input.

對於每筆測資，你的程式須將輸出於一行，並說明該筆測資一共內含幾個不同的正方形。

Sample Input

```
2
1
8
0
```

Sample Output

```
5
1
204
```

12468 Zapping

分級：1 **分類：模擬**

I'm a big fan of watching TV. However, I don't like to watch a single channel; I'm constantly zapping between different channels.

My dog tried to eat my remote controller and unfortunately he partially destroyed it. The numeric buttons I used to press to quickly change channels are not working anymore. Now, I only have available two buttons to change channels: one to go up to the next channel (the ∆ button) and one to go down to the previous channel (the ▽ button). This is very annoying because if I'm watching channel 3 and want to change to channel 9 I have to press the ∆ button 6 times!

My TV has 100 channels conveniently numbered 0 through 99. They are cyclic, in the sense that if I'm on channel 99 and press ∆ I'll go to channel 0. Similarly, if I'm on channel 0 and press ▽ I'll change to channel 99.

I would like a program that, given the channel I'm currently watching and the channel I would like to change to, tells me the minimum number of button presses I need to reach that channel.

我是個超級電視迷，但是不喜歡固定看一個頻道，我經常切換在不同頻道間。

但不幸的是，我的狗咬壞了部分的遙控器鈕，我以前快速換台的數字鍵都不能用了，現在只剩兩個按鈕可以換頻道：一個往上切一個頻道 (∆ 按鈕)，一個往下切一個頻道 (▽ 按鈕)。這樣真的很煩，因為如果我要從頻道 3 換到頻道 9 我得按 6 次 ∆ 按鈕！

我的電視有 100 個頻道，號碼為 0 到 99。它們是循環的，也就是說，我從 99 台再按一下 ∆ 就會回到第 0 台。同理，我從第 0 台按一下 ▽ 就會回到 99 台。

我想要寫個程式，讓我輸入現在正在看的頻道和我想切換的頻道，它便告訴我最少需要按幾次按鈕才能到達。

Input

The input contains several test cases (at most 200).

Each test case is described by two integers a and b in a single line. a is the channel I'm currently watching and b is the channel I would like to go to (0 ≤ a, b ≤ 99).

The last line of the input contains two ` -1 's and should not be processed.

輸入含有多筆測資 (最多 200 筆)。

每筆測資皆描述於單行，含有兩個整數 a 和 b。a 是我現在看的頻道，而 b 則是我想要切換的頻道 (0 ≤ a, b ≤ 99)。

最後一行有兩個 -1，代表輸入結束。

Output

For each test case, output a single integer on a single line－the minimum number of button presses needed to reach the new channel (Remember, the only two buttons I have available are △ and ▽).

對於每筆測資，輸出一個整數於一行，也就是我最少要按幾次按鈕才能切到新頻道(記住，我只有 △ 和 ▽ 兩個按鈕可用)。

Sample Input

```
3 9
0 99
12 27
-1 -1
```

Sample Output

```
6
1
15
```

12602 Nice Licence Plates

分級：1 **分類：字元與字串**

Alberta licence plates currently have a format of ABC-0123 (three letters followed by four digits).

We say that the licence plate is ""nice"" if the absolute difference between the value of the first part and the value of the second part is at most 100.

The value of the first part is calculated as the value of base-26 number (where digits are in [A .. Z]). For instance, if the first part is "" ABC "", its value is 28 (0*26^2 + 1*26^1 + 2*26^0). So, the plate "" ABC-0123 "" is nice, because |28 -123| ≤ 100.

Given the list of licence plate numbers, your program should determine if the plate is nice or not.

亞伯達省的車牌目前有 ABC-0123 的樣式規格（三個字母後面接四個數字）。

而我們所謂“好的”車牌，第一部分的值與第二部分的值之間差最多為 100。

第一部分值是基於 26 英文碼(〔A.. Z])來計算數值。舉例來說，若第一部分為“ABC”時，它的數值便是 28（0 *26^ 2+ 1 *26^ 1+ 2×26^ 0），且|28-123|<=100，所以說車牌“ABC-0123”即是好的。

給定車牌號碼的清單，你的程式應確定這些牌號的好壞。

Input

First line of the input contains an integer N (1 ≤ N ≤ 100), the number of licence plate numbers. Then follow N lines, each containing a licence plate in the format ` LLL - DDDD '.

輸入的第一行包含一個整數 N（1≤ N ≤100），代表下列車牌號碼的數目。接下來的 N 行，每行包含一個車牌，格式皆為“LLL - DDDD”。

Output

For each licence plate print on a line ` nice ' or ` not nice ' (without quotes) depending on the plate number being nice as described in the problem statement.

對於每個輸入車牌，須根據題目說明中所述的車牌號規則判斷，並單行印出其為`nice' 或 `not nice'（不需帶引號）。

Sample Input

```
2
ABC-0123
AAA-9999
```

Sample Output

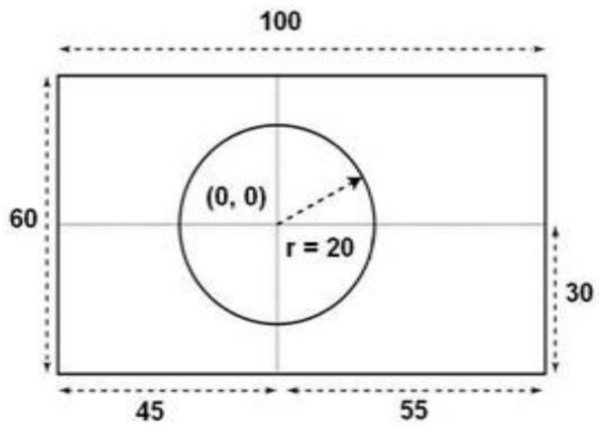
```
nice
not nice
```

12611　Beautiful Flag

分級：1　　　　　　　　　　　　**分類：幾何**

Teering is a little boy. He is trying to draw the national flag of Bangladesh. Being smart he knows he has to maintain the correct ratio and measurement while drawing the flag. You know the rules of drawing the national flag, don't you? If not, no worries, Teering is here to help you:

The national flag of Bangladesh consist of a green rectangle with a red circle inside it. The ratio of the length and width of the rectangle is 100 : 60 (i.e. if the length is 100 units then the width will be 60 units). The radius of the circle is 20% of the length (i.e. if the length is 100 units then the radius of the circle will be 20 units). To get the center of the circle you need to draw a horizontal line dividing the width in equal portion and draw a vertical line dividing the length in 45 : 55 ratio (i.e. if the length of the rectangle is 100 then 45 units will be in left and 55 units will be on the right side of the line). The crossing of the two lines will be the center of the circle. Here is an illustrated picture for better understanding.



Now Teering has started to draw a flag. He has already drawn the circle of radius R centered at the origin in a 2D co-ordinate system. Now he needs to determine the corner of the rectangle so that he can join them to complete the flag. Can you help him?

Teering 是一個小男孩。他正試圖描繪孟加拉多的國旗，聰明的他知道繪製國旗必須保持其正確的比率。你也知道繪製國旗的規則吧，不是嗎？如果不知道，無須擔心 Teering 在這，他會幫助你的：

孟加拉多國旗由一個綠色的長方形裡面一個紅色的圓圈所構成，矩形的長寬比為

100：60（即如果該長度為 100 單位，則寬將為 60 單位）。該圓半徑為長度的 20%（即如果長度為 100 單位，則圓的半徑將為 20 單位）。要得到圓的中心，你需要繪製一條水平線來分割成相等的寬，再繪製一條垂直線將其長度分割成 45：55 的比例（即如果矩形長度為 100 單位，則線上的左半部會有 45 單位，右半部會有 55 單位），這兩條線的交叉點就會是圓的中心點。下面有張圖片讓你比較好來理解。

現在 Teering 要開始來繪製國旗了，他已經畫好了以 R 為半徑，中心點在二維座標原點上的圓形，現在他需要確定矩形的角落位置，好讓他加入它們來完成國旗繪製。你能幫幫他嗎？

Input

The first line of input will contain an integer T (T < 101) denoting the number of test cases. Each of the following T lines will contain an integer R (R < 1001) each denoting the radius of the circle.

輸入的第一行包含一個整數Ｔ（Ｔ＜100）表示測試用例的數量。在以下Ｔ行都包含著一個整數Ｒ（Ｒ＜1001）表示一個圓的半徑。

Output

For each input output five lines. The first line will contain the case number. The following four lines will denote the upper left, upper right, lower right and lower left coordinates of the rectangle for the flag respectively. You have to print x coordinate and y coordinate separated by space in each line. You may assume that input is given in such that the corners will always be in integer coordinates. See sample input output for details.

對於每個輸入都輸出 5 行。第一行會有測試值的編號，下面四行將依序表示左上、右上、右下和左下的國旗矩形之座標。你每行必須印出以空白為間隔的 x 座標與 y 座標。妳可以假設輸入所給出的角落座標為整數座標。詳情請參見樣本的輸入輸出。

Sample Input

```
2
20
100
```

Sample Output

Case1:

30 -45

55 30

55 -30

-45 -30

Case2:

150 -225

275 150

275 -150

-255 -150

12416　Excessive Space Remover

分級：1　　　　　　　　　　**分類：模擬**

How do you remove consecutive spaces in a simple editor like notepad in Microsoft Windows? One way is to repeatedly "replace all" two consecutive spaces with one space (we call it an action). In this problem, you're to simulate this process and report the number of such "replace all" actions.

For example, if you want to remove consecutive spaces in A very big   joke.", you need two actions:

"A very big   joke." -> "A very big joke." -> "A very big joke."

在像 Windows 記事本這類簡單的編輯器中，你要如何移除連續的空白呢？可以重覆地以一個空白「全部取代」兩個連續的空白 (我們稱之為一個動作)。本題中你要模擬這個程序並回報需要幾個「全部取代」的動作。

比如說，如果你要移除「A very big   joke.」中的連續空白，你需要兩個動作：

"A very big   joke." -> "A very big joke." -> "A very big joke."

Input

The input contains multiple test cases, one in a separate line. Each line contains letters, digits, punctuations and spaces (possibly leading spaces, but no trailing spaces). There will be no TAB character in the input. The size of input does not exceed 1MB.

Explanation

If you can't see clearly, here is the sample input, after replacing spaces with underscores:

A*very**big****joke.
*********Goodbye!

輸入含有多筆測資，每筆一行。每行含有字母、數字、標點符號及空白 (可能有前導空白，但不會有後置空白)。輸入中不會有 TAB 字元。輸入檔大小不會超過 1MB。

提示：
如果看不清楚，範例輸入中的空白以星號取代後如下：

A*very**big****joke.
*********Goodbye!

Output

For each line, print the number of actions that are required to remove all the consecutive spaces.

對於每一行，印出移除所有連續空白所需的動作次數。

Sample Input

```
A very  big    joke.
         Goodbye!
```

Sample Output

```
2
4
```

490 Rotating Sentences

分級：1 **分類：字元與字串**

In ''Rotating Sentences'', you are asked to rotate a series of input sentences 90 degrees clockwise. So instead of displaying the input sentences from left to right and top to bottom, your program will display them from top to bottom and right to left.

在這個''Rotating Sentences''問題中，你必須將數列文字往順時針方向旋轉 90 度。也就是說將原本由左到右，由上到下的句子輸出成由上到下，由右到左。

Input

As input to your program, you will be given a maximum of 100 sentences, each not exceeding 100 characters long. Legal characters include: newline, space, any punctuation characters, digits, and lower case or upper case English letters. (NOTE: Tabs are not legal characters.)

給你的輸入最多不會超過 100 列，每列最多不會超過 100 個字元。合法的字元包括：換行，空白，所有的標點符號，數字，以及大小寫字母。（注意：Tabs 並不算是合法字元。）

Output

The output of the program should have the last sentence printed out vertically in the leftmost column; the first sentence of the input would subsequently end up at the rightmost column.

最後一列輸入必須垂直輸出在最左邊一行，輸入的第一列必須垂直輸出在最右邊一行。

Sample Input

```
Rene Decartes once said,
"I think, therefore I am."
```

Sample Output

```
"R
Ie
 n
te
```

h
iD
ne
kc
,a
r
tt
he
es
r
eo
fn
oc
re
e
s
la
i
ad
m,
.
"

136 Ugly Numbers

分級：1 **分類：質數、因數與倍數**

Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence

1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...

shows the first 11 ugly numbers. By convention, 1 is included.

Write a program to find and print the 1500'th ugly number.

Ugly Number 的定義為：該數之質因數必須為 2, 3 或 5。在此列舉一串數列：

1,2,3,4,5,6,8,9,10,12,15,...

這些就是前 11 個 Ugly Numbers。依照慣例，1 也算是在內。

請寫一個程式求出第 1500 個的醜數。

Input

There is no input to this program.

此題沒有輸入值。

Output

Output should consist of a single line as shown below, with <number> replaced by the number computed.

輸出值需為單行輸出，如輸出範例所示，並用計算出來的值替代<number>。

Sample Input

Sample Output

```
The 1500'th ugly number is <number>.
```

12626 I ❤ Pizza

分級：1 **分類：字元與字串**

In our particular computerized kitchen, ingredients are named by capital letters: A, B, C, D... Thus, to make a pizza MARGARITA we need as many ingredients as their letters, i.e. one M, three A, two R, one G, one I, and one T.

For example, if we have the ingredients:

AAAAAAMMRRTITIGGRRRRRRRR

Then we can make 2 pizzas MARGARITA, and still spare some R.

Given a set of ingredients, you have to say how many pizzas MARGARITA can be made. Note that there may be leftover ingredients, and also there may be unnecessary ingredients, such as B.

在我們特別的智慧廚房裡，食材會依大寫英文字母來命名。因此，要做一個披薩 MARGARITA，我們需要跟那些字母數一樣的食材量，換言之就是要一個M、三個A、兩個R、一個G、一個I與一個T。

舉個例子，如果我們有這些食材：

AAAAAAMMRRTITIGGRRRRRRRR

然後我們就可以做兩塊 MARGARITA 披薩，仍還剩餘一些的R。

你必須將材料分組，說明可以做多少塊的 MARGARITA 披薩。注意可能會有剩下的食材，他們也有可能是不必用到的食材，像是B。

Input

The first line contains a natural number, N, which indicates the number of test cases.

Each test case is given in one line. This line contains a series of capital letters from A to Z, which can be messy and may be repeated. At most one line can have 600 characters.

第一行包含一個自然數N，表示測試值的數量。

每一行都有一個測試值，包含著一系列範圍從A到Z的大寫英文字母，它可以是複雜且重複的。在一行中最多可以有 600 字母。

Output

For each test case, you must indicate how many pizzas MARGARITA can be made with the letters available, taking into account that there may be spare letters.

對於每個測試值，你必須根據這些字母來表示出可以做多少的MARGARITA披薩，也需考慮到裡面可能有剩餘的食材。

Sample Input

```
5
MARGARITA
AAAAAAMMRRTITIGGRRRRRRRR
AMARGITA
BOLOGNESACAPRICHOSATOMATERA
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Sample Output

```
1
2
0
1
0
```

12289　One-Two-Three

分級：1　　　　　　　　　**分類：字元與字串**

Your little brother has just learnt to write one, two and three, in English. He has written a lot of those words in a paper, your task is to recognize them. Note that your little brother is only a child, so he may make small mistakes: for each word, there might be at most one wrong letter. The word length is always correct. It is guaranteed that each letter he wrote is in lower-case, and each word he wrote has a unique interpretation.

你年幼弟弟剛學會寫英文的一二三。他在一張紙上寫了很多的這幾個字，而你的工作便是來辨認它們。要注意的是你弟弟不過是個小孩子，因此他會犯些小錯誤：每個單字至多一個錯誤的字母，且單字長度一定是正確的。他所寫的一定是小寫字母，而每個單字只可能有一種解釋。

Input

The first line contains the number of words that your little brother has written. Each of the following lines contains a single word with all letters in lower-case. The words satisfy the constraints above: at most one letter might be wrong, but the word length is always correct. There will be at most 10 words in the input.

第一行包含你弟弟所寫單字數。接下來的每一行含有一個小寫字母組成的單字。單字必符合上述限制：至多一個錯誤的字母，但是單字長度永遠正確。輸入中最多有 10 個單字。

Output

For each test case, print the numerical value of the word.
對每筆測資，輸出單字的數值。

Sample Input

```
3
owe
too
theee
```

Sample Output

```
1
2
```

12250 Language Detection

分級：1 **分類：字元與字串**

English, Spanish, German, French, Italian and Russian are the 6 most prominent languages in the countries of European Union. All of these languages have different words to represent the English word "HELLO". For example in Spanish the word equivalent to "HELLO" is "HOLA". In German, French, Italian and Russian language the word that means (or similar to) "HELLO" is "HALLO", "BONJOUR", "CIAO" and "ZDRAVSTVUJTE" respectively. In this problem your task is pretty simple. You will be given one of the six words mentioned above or any other word and you will have to try and detect the language it is from.

英文、西班牙文、德文、法文、義大利文及俄文為歐盟國家中最盛行的 6 種語言，而這些語言都以不同的字來表示英文的「HELLO」。例如西班牙文中等同於英文「HELLO」的字是「HOLA」，而德文、法文、義大利文及俄文中意思為(或相近)「HELLO」的字依序為「HALLO」、「BONJOUR」、「CIAO」和「ZDRAVSTVUJTE」。你在本題中的任務非常簡單。給你以上的六個單字之一或是其他的單字，你需要辨識它是哪的語言。

Input

Input file contains around 2000 lines of inputs. Each line contains a string S. You can assume that all the letters of the string are uppercase English letters and the maximum length of the string is 14. Input is terminated by a line containing a single `#' character (without the quote). This line should not be processed.

輸入檔含有大約 2000 行的輸入。每行皆含有一字串 S。你可以假設所有的字母都是大寫英文字母，且字串的最大長度為 14。輸入以僅含有一個「#」的一行作為結束，且該行不用執行。

Output

For each line of input except the last one produce one line of output. This line contains the serial of output followed by a language name. If the input string is `HELLO' or `HOLA' or `HALLO' or `BONJOUR' or `CIAO' or `ZDRAVSTVUJTE' then you should report the language it belongs to. If the input string is something other than these 6 strings print the string `UNKNOWN' (without the quotes) instead. All characters in the output strings are uppercase as well. Look at the output for sample input for formatting details.

除了最後一行以外，每組輸入資料都要有一行的輸出結果，此含有輸出的序號及語言名稱。如果輸入的字串是「HELLO」、「HOLA」、「HALLO」、「BONJOUR」、「CIAO」或「ZDRAVSTVUJTE」時，你要回報它是哪一種語言。如果輸入字串是這 6 個以外的字串則印出字串「UNKNOWN」。所有的輸出字串也都是大寫。詳細的格式細節請參見範例樣式。

Sample Input

```
HELLO
HOLA
HALLO
BONJOUR
CIAO
ZDRAVSTVUJTE
#
```

Sample Output

```
Case 1: ENGLISH
Case 2: SPANISH
Case 3: GERMAN
Case 4: FRENCH
Case 5: ITALIAN
Case 6: RUSSIAN
```

10055　Hashmat the Brave Warrior

分級：1　　　　　　　　**分類：數學計算**

Hashmat is a brave warrior who with his group of young soldiers moves from one place to another to fight against his opponents. Before Fighting he just calculates one thing, the difference between his soldier number and the opponent's soldier number. From this difference he decides whether to fight or not. Hashmat's soldier number is never greater than his opponent.

Hashmat 是一個勇敢的將領，他帶著年輕的士兵從這個城市移動到另一個城市與敵人對抗。在打仗之前他會計算己方與敵方士兵的數目差距，來決定是要開打或不開打。Hashmat 的士兵數絕不會比敵人的士兵數大。

Input

The input contains two numbers in every line. These two numbers in each line denotes the number soldiers in Hashmat's army and his opponent's army or vice versa. The input numbers are not greater than 2^32 . Input is terminated by `End of File'.

每組測試資料有 2 個整數於 1 列，代表 Hashmat 及敵人的士兵數，順序不定，這些數不會超過 2^32。輸入以 EOF 結束。

Output

For each line of input, print the difference of number of soldiers between Hashmat's army and his opponent's army. Each output should be in separate line.

對每組測試資料請輸出 Hashmat 與敵人士兵數目的差（正數）。每筆測資皆輸出一行。

Sample Input

```
10 12
10 14
100 200
```

Sample Output

```
2
4
```

11827　Maximum GCD

分級：1　　　　　　　　**分類：數學計算**

Given the N integers, you have to find the maximum GCD (greatest common divisor) of every possible pair of these integers.

給你 n 個正整數,你需要去找他們所有之中最大的一對 GCD 值 (greatest common divisor)

Input

The first line of input is an integer N(1 < N < 100) that determines the number of test cases. The following N lines are the N test cases. Each test case contains M (1 < M < 100) positive integers that you have to find the maximum of GCD.

輸入第一行有一整數 N(1 < N < 100)表示有幾組的測試資料，接下來的 N 行包含 M 個正整數(1 < M < 100)讓你去找其中的最大的一對 GCD 值。

Output

For each test case show the maximum GCD of every possible pair.

對於每組資料請輸出最大的一對 GCD 值。

Sample Input

```
3
10 20 30 40
7 5 12
125 15 25
```

Sample Output

```
20
1
25
```

12439　February 29

分級：1　　　　　　　　　**分類：數學計算**

It is 2012, and it's a leap year. So there is a "February 29" in this year, which is called leap day. Interesting thing is the infant who will born in this February 29, will get his/her birthday again in 2016, which is another leap year. So February 29 only exists in leap years. Does leap year comes in every 4 years? Years that are divisible by 4 are leap years, but years that are divisible by 100 are not leap years, unless they are divisible by 400 in which case they are leap years.

In this problem, you will be given two different date. You have to find the number of leap days in between them.

今年 2012 年，也是閏年，所以說今年就有「二月 29 日」，而這天也稱為「閏日」。有趣的是，今年二月 29 日出生的嬰兒要等到下一個閏年，也就是 2016 年才過生日。閏年是每 4 年一次嗎？能被 4 整除的年份雖說是閏年，但是若被 100 整除的年份卻不是閏年，除非它又可以被 400 整除，它才是閏年。

在本題中會給你兩個不同的日期，你要找出它們兩者之間有幾個「閏日」。

Input

The first line of input will contain T (≤ 500) denoting the number of cases.

Each of the test cases will have two lines. First line represents the first date and second line represents the second date. Note that, the second date will not represent a date which arrives earlier than the first date. The dates will be in this format —'month day, year'. See sample input for exact format. You are guaranteed that dates will be valid and the year will be in between 2 ∗ 10^3 to 2 ∗ 10^9. For your convenience, the month list and the number of days per months are given below. You can assume that all the given dates will be a valid date.

第一行有一個 T (≤ 500) 表示測資筆數。

每筆測資皆有兩行。第一行為第一個日期，第二行則為第二個。注意，第二個日期決不會比第一個早。而日期格式為 -「month day, year」。詳細情形請參見範例輸出。保證所有日期均為正確日期，而年份則介於 2 * 10^3 和 2 * 10^9 之間。為了方便，月份名稱及每個月的天數詳細如下所示。

Output

For each case, print the case number and the number of leap days in between two given dates (inclusive).

Note:

The names of the months are {“January”, “February”, “March”, “April”, “May”, “June”, “July”,“August”, “September”, “October”, “November” and “December”}. And the numbers of days for the months are {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30 and 31} respectively in a non-leap year. In a leap year, number of days for February is 29 days; others are same as shown in previous line.

對於每筆測資，印出測資編號及兩個日期之間（含）有幾個「閏日」。

筆記：

每月的英文名稱為“January”、“February”, “March”　、“April”　、“May”　、“June”　、“July”　、“August”　、“September”　、 “October”　、 “November” 與“December”。非閏年的各月天數依序為 31、28、 31、30、31、30、31、31、30、31、30、31 天，而在閏年，除了二月有 29 天，其他月皆與非閏年相同。

Sample Input

```
4
January 12, 2012
March 19, 2012
August 12, 2899
August 12, 2901
August 12, 2000
August 12, 2005
February 29, 2004
February 29, 2012
```

Sample Output

```
Case 1: 1
Case 2: 0
Case 3: 1
Case 4: 3
```

11150　Cola

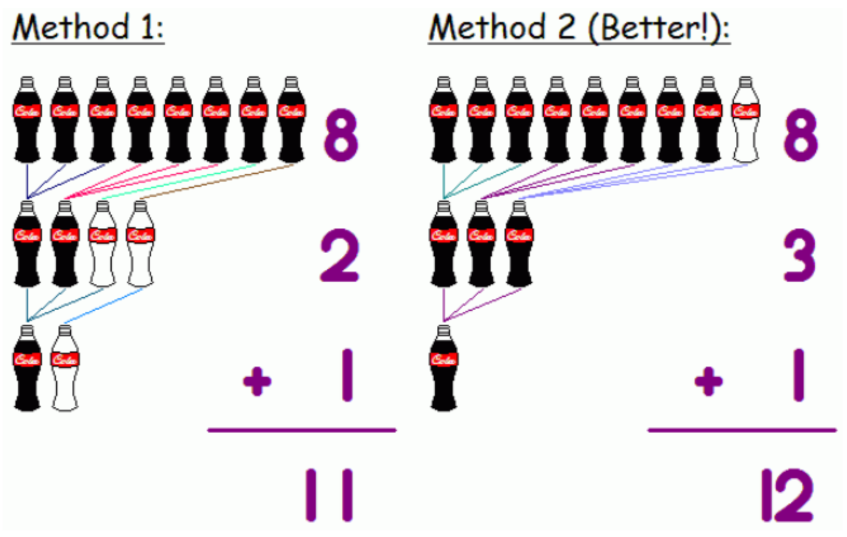
分級：1　　　　　　　　**分類：數學計算**

You see the following special offer by the convenience store:
"A bottle of Choco Cola for every 3 empty bottles returned"

Now you decide to buy some (say N) bottles of cola from the store. You would like to know how you can get the most cola from them.

The figure below shows the case where N = 8. Method 1 is the standard way: after finishing your 8 bottles of cola, you have 8 empty bottles. Take 6 of them and you get 2 new bottles of cola. Now after drinking them you have 4 empty bottles, so you take 3 of them to get yet another new cola. Finally, you have only 2 bottles in hand, so you cannot get new cola any more. Hence, you have enjoyed 8 + 2 + 1 = 11 bottles of cola.

You can actually do better! In Method 2, you first borrow an empty bottle from your friend (?! Or the storekeeper??), then you can enjoy 8 + 3 + 1 = 12 bottles of cola! Of course, you will have to return your remaining empty bottle back to your friend.



你在便利商店看到一個這樣的特別優惠: "3 瓶空可樂罐可以換一瓶可樂！"

現在，你決定從便利商店買一些可樂(N 瓶)，而你也想知道你最多可以喝到幾瓶可樂。

如圖所示，當 N = 8 時候的情況：方法一是標準的做法，在你喝了 8 瓶可樂之後，你有 8 個空罐子，把其中 6 瓶空罐子換成 2 瓶新的，喝完之後你就有 4 瓶空罐子，然後你又可以拿 3 瓶換 1 瓶新的，最後，你只有 2 個空罐子，所以你再也不能再換新的可樂了。所以你總共可以喝到 8 + 2 + 1 = 11 瓶可樂。

可是其實你有一個更好的方法！在第二種方法裡面，你可以先和你的朋友(或者是店員？)借一個空罐子，然後你就可以喝到 8 + 3 + 1 = 12 瓶可樂！當然，你最後要把 1 瓶空罐子還給你的朋友。

Input

Input consists of several lines, each containing an integer N (1 ≤ N ≤ 200).

輸入有好幾組資料，一組一行，包含有一個正整數 N (1 ≤ N ≤ 200)。

Output

For each case, your program should output the maximum number of bottles of cola you can enjoy. You may borrow empty bottles from others, but if you do that, make sure that you have enough bottles afterwards to return to them.

Note: Drinking too much cola is bad for your health, so... don't try this at home!! :-)

對於每組輸入，你的程式須輸出你最多可以喝到幾瓶可樂。你可以和別人借一些空罐子，但是請記得還給他一樣數量的空罐子。

請注意：喝太多可樂對身體不好，所以不要在家嘗試喔！:-)

Sample Input

```
8
```

Sample Output

```
12
```

10474 Where is the Marble?

分級：1 **分類：排序**

Raju and Meena love to play with Marbles. They have got a lot of marbles with numbers written on them. At the beginning, Raju would place the marbles one after another in ascending order of the numbers written on them. Then Meena would ask Raju to find the first marble with a certain number. She would count 1...2...3. Raju gets one point for correct answer, and Meena gets the point if Raju fails. After some fixed number of trials the game ends and the player with maximum points wins. Today it's your chance to play as Raju. Being the smart kid, you'd be taking the favor of a computer. But don't underestimate Meena, she had written a program to keep track how much time you're taking to give all the answers. So now you have to write a program, which will help you in your role as Raju.

Raju 和 Meena 喜歡一起玩彈珠，而他們手上有著很多寫著數字的彈珠。Raju 按照彈珠上面的號碼由小到大排成一列，然後 Meena 會要求 Raju 找出某個號碼的第一顆彈珠所在的位置。她會算 1...2...3...，如果 Raju 答對了，他就得 1 分，否則 Meena 得 1 分，直到遊戲結束，並以最高分的玩家獲勝。今天你有機會扮演 Raju 的角色，而作為聰明的孩子，你會寫一個程式來讓電腦計算。不過可別小看 Meena，她則寫了一個程式來檢查你花多少時間來回答所有的問題。

Input

There can be multiple test cases. Total no of test cases is less than 65. Each test case consists begins with 2 integers: N the number of marbles and Q the number of queries Meena would make. The next N lines would contain the numbers written on the N marbles. These marble numbers will not come in any particular order. Following Q lines will have Q queries. Be assured, none of the input numbers are greater than 10000 and none of them are negative.

Input is terminated by a test case where N = 0 and Q = 0.

此題有多筆測資（最多 65 筆），每筆測資的第一行有兩個數字：N 代表有多少的彈珠，Q 代表 Meena 想找到的彈珠值。接下來會有 N 列，每列各有一個彈珠的值（沒有按照順序出現）。再來有 Q 列皆含一個欲搜尋的值。可以確定的，不會有任何的值是負數，且不含會超過 100 00。

Output

For each test case output the serial number of the case. For each of the queries, print one line of output. The format of this line will depend upon whether or not the query number is written upon any of the marbles. The two different formats are described below:

・ \`x found at y ', if the first marble with number x was found at position y . Positions are numbered 1 , 2 , ⋯ , N .

・ \`x not found ', if the marble with number x is not present.

Look at the output for sample input for details.

對每組測試資料請先輸出一列，說明這是第幾組測試資料。而對每組 Meena 所問的每個問題也請輸出於一列，輸出格式如下所示的其中之一：

・ x found at y ： 如果第一個號碼為 x 的彈珠在位置 y 被發現（位置從 1 開始算）

・ x not found ： 如果找不到號碼為 x 的彈珠

輸出格式請參考 Sample Output。

Sample Input

```
4 1
2
3
5
1
5
5 2
1
3
3
3
1
2
3
0 0
```

Sample Output

```
CASE# 1:
5 found at 4
CASE# 2:
2 not found
3 found at 3
```

100 The 3n + 1 problem

分級：1 **分類：模擬**

Consider the following algorithm:

```
1.      input n
2.      print n
3.      if n = 1 then STOP
4.          if n is odd then   n ← 3*n+1
5.          else   n ← n/2
6.      GOTO 2
```

Given the input 22, the following sequence of numbers will be printed 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers *n* such that 0 < *n* < 1,000,000 (and, in fact, for many more numbers than this.)

Given an input *n*, it is possible to determine the number of numbers printed (including the 1). For a given *n* this is called the *cycle-length* of *n*. In the example above, the cycle length of 22 is 16.

For any two numbers *i* and *j* you are to determine the maximum cycle length over all numbers between *i* and *j*.

考慮以下的演算法：

```
1.      輸入 n
2.      印出 n
```

```
3.          如果  n = 1  結束
4.          如果  n  是奇數  那麼  n=3*n+1
5.          否則  n=n/2
6.          GOTO 2
```

例如輸入 22，會印出數列： 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

據推測此演算法對任何整數而言會終止 (當列印出 1 的時候)。雖然此演算法很簡單，但以上的推測是否真實卻無法知道。然而對所有的 n (0 < n < 1,000,000) 來說，以上的推測已經被驗證是正確的。

給一個輸入 n，透過以上的演算法我們可以得到一個數列（1 作為結尾），此數列的長度就稱為 n 的 cycle-length。像是上面所提，22 的 cycle length 便為 16。

現在有 2 個任意整數 i、j，你要知道介於 i、j（包含 i、j）之間的數所產生的數列中最大的 cycle length 是多少。

Input

The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.

輸入可能包含了好幾組測試資料，每組一列有一對整數 i、j，且所有數字皆會大於 0 小於 1,000,000。

Output

For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

請按照輸入順序，對測資的每一對 i、j 做輸出於一列。你應該要先印出 i、j，再印出介於 i、j（包含）之間的數所產生的數列中最大的 cycle length，印出順序不可變動，且此三數最少須以一個空白為間格。

Sample Input

```
1 10
```

```
100 200
201 210
900 1000
```

Sample Output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

913　Joana and the Odd Numbers

分級：1　　　　　　　　　　　**分類：模擬**

Joana loves playing with odd numbers. In the other day, she started writing, in each line, an odd number of odd numbers. It looked as follows:

```
1
3 5 7
9 11 13 15 17
19 21 23 25 27 29 31
…
```

On a certain line Joana wrote 55 odd numbers. Can you discover the sum of the last three numbers written in that line? Can you do this more generally for a given quantity of odd numbers? Given the number N of odd numbers in a certain line, your task is to determine the sum of the last three numbers of that line.

Joana 喜歡玩關於奇數的遊戲。有一天，她開始寫，每列都是奇數，如下表。

```
 1
 3  5  7
 9 11 13 15 17
19 21 23 25 27 29 31
…
```

在某一列 Joana 寫下了 55 個奇數數字，你可以看出該列最後 3 個數字的和嗎？給你一個數字 N，代表某一列有 N 個奇數數字，你的任務是把該列最後三個數加起來。

Input

The input is a sequence of lines, one odd number N(1 < N < 1000000000) per line.

輸入含有多組測試資料。每組測試資料一列，有一個數字 N，表示某一列會有 N 個奇數數字（1 < N < 1000000000）。

Output

For each input line write the sum of the last three odd numbers written by Joana in that line with N numbers. This sum is guaranteed to be less than 2^63.

對每組測試資料，輸出該列的最後三個數字的和。本問題中保證三個數字的和一定小於 2^63。

Sample Input

```
3
5
7
```

Sample Output

```
15
45
87
```

10783　Odd Sum

分級：1　　　　　　　　　　**分類：數學計算**

Given a range [a, b], you are to find the summation of all the odd integers in this range. For example, the summation of all the odd integers in the range [3, 9] is 3 + 5 + 7 + 9 = 24.

給你一個範圍[a, b]，請你找出 a 與 b 之間所有奇數的和。例如：範圍[3, 9]中所有奇數的和就是 3 + 5 + 7 + 9 = 24。

Input

There can be at multiple test cases. The first line of input gives you the number of test cases, T (1 ≤ T ≤ 100). Then T test cases follow. Each test case consists of 2 integers a and b (0 ≤ a ≤ b ≤ 100) in two separate lines.

此題含有多筆的測資。輸入第一列會給你一個整數 T（1 ≤ T ≤ 100），代表接下來有多少筆的測資。而每筆測資為兩列，包含兩個數 a 與 b（0 ≤ a ≤ b ≤ 100）。

Output

For each test case you are to print one line of output - the serial number of the test case followed by the summation of the odd integers in the range [a, b].

每組測試資料輸出一列，內容為在範圍[a, b]間所有奇數的和。

Sample Input

```
2
1
5
3
5
```

Sample Output

```
Case 1: 9
Case 2: 8
```

分級 2

12019   A - Doom's Day Algorithm

分級：2   **分類：數學計算**

No. Doom’s day algorithm is not a method to compute which day the world will end. It is an algorithm created by the mathematician John Horton Conway, to calculate which day of the week (Monday, Tuesday, etc.) corresponds to a certain date.

This algorithm is based in the idea of the doomsday, a certain day of the week which always occurs in the same dates. For example, 4/4 (the 4th of April), 6/6 (the 6th of June), 8/8 (the 8th of August), 10/10 (the 10th of October) and 12/12 (the 12th of December) are dates which always occur in doomsday. All years have their own doomsday.

In year 2011, doomsday is Monday. So all of 4/4, 6/6, 8/8, 10/10 and 12/12 are Mondays. Using that information, you can easily compute any other date. For example, the 13th of December 2011 will be Tuesday, the 14th of December 2011 will be Wednesday, etc.

Other days which occur on doomsday are 5/9, 9/5, 7/11 and 11/7. Also, in leap years, we have the following doomsdays: 1/11 (the 11th of January) and 2/22 (the 22nd of February), and in non-leap years 1/10 and 2/21.

Given a date of year 2011, you have to compute which day of the week it occurs.

No. Doom’s day 演算法不是一個計算世界末日時間的方法。它是由數學家 John Horton Conway 創建，用來計算某一特定日期對應於星期幾（星期一，星期二，等等）的算法。

這個演算法是基於末日來發想出的，這總是會發生在同一週的某一日期。例如，日期在 4/4（4 月的第 4 天），6/6（6 月的第 6 天），8/8（8 月的第 8 天），10/10（10 月的第 10 天）和 12/12（12 月的第 12 天），Doomsday 總是發生在那幾天，所有年份都有自己的 Doomsday。

2011 年，Doomsday 是在星期一。因此，所有的 4/4，6/6，8/8，10/10 和 12/12 都為星期一。利用這些資訊，你可以很容易地計算其他任何的日期。例如，2011 年 12 月 13 日會是星期二，而 2011 年 12 月 14 日是星期三，等等。

會發生 Doomsday 的還有其他日子：5/9，9/5，7/11 和 11/7。此外在閏年，還有以下的日期：1/11（1 月的第 11 天）和 2/22（2 月的第 22 天）。而非在閏年有 1/10 和 2/21。

給你西元 2011 年的某月某日，請你判斷 Doomsday 會在星期幾發生。

Input

The input can contain different test cases. The first line of the input indicates the number of test cases.

For each test case, there is a line with two numbers: M D. M represents the month (from 1 to 12) and D represents the day (from 1 to 31). The date will always be valid.

輸入含有多筆不同的資料，其第一行表示測資的筆數。

每一筆測資為一行，內含 2 個整數：M D。M 代表月份（1 至 12 月），D 代表天數（1 至 31 天），且日期都為有效。

Output

For each test case, you have to output the day of the week where that date occurs in 2011. The days of the week will be: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

對於每筆測資，請你判斷該日期在 2011 年是星期幾。星期一到星期日分別為 Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday。

Sample Input

```
8
1 6
2 28
4 5
5 26
8 1
11 1
12 25
12 31
```

Sample Output

Thursday
Monday
Tuesday
Thursday
Monday
Tuesday
Sunday
Saturday

12545 Bits Equalizer

分級：2 **分類：字元與字串**

You are given two non-empty strings S and T of equal lengths. S contains the characters `0', `1' and `?', whereas T contains `0' and `1' only. Your task is to convert S into T in minimum number of moves. In each move, you can

1. change a `0' in S to `1'
2. change a `?' in S to `0' or `1'
3. swap any two characters in S

As an example, suppose S = "01??00" and T = "001010". We can transform S into T in 3 moves:

- Initially S = "01??00"
- - Move 1: change S[2] to `1'. S becomes "011?00"
- - Move 2: change S[3] to `0'. S becomes "011000"
- - Move 3: swap S[1] with S[4]. S becomes "001010"
- S is now equal to T

您將得到兩個相同長度的非空串 S 和 T。 S 包含字符'0'，'1'和'？'，而 T 包含'0'和'1'而已。你的任務是將 S 以最低的次數移動成 T。在每一次的移動，你可以：

1. 將 S 中的一個`0’改為`1’
2. 將 S 中的一個`?’改為`0’或`1’
3. 將 S 中的任意二數交換位置

作個例子，假設 S =“01??00”和 T=“001010”。我們可以 3 步驟內將 S 轉換成 T：

- 一開始 S = "01??00"
- -步驟 1: 將 S[2] 改成`1'。S 便轉換成 "011?00"
- -步驟 2: 將 S[3] 改成`0'。S 便轉換成 "011000"
- -步驟 3: 將 S[1]和 S[4]交換位置。S 便轉換成 "001010"
- 現在 S 與 T 就會相等了

Input

The first line of input is an integer C (C ≤ 200) that indicates the number of test cases. Each case consists of two lines. The first line is the string S consisting of `0', `1' and `?'. The second line is the string T consisting of `0' and `1'. The lengths of the strings won't be larger than 100.

輸入的第一行中有一個整數 C（C ≤ 200），代表測試用資料的數量。每個資料皆包括兩行，第一行是以'0'，'1'和'？'所組成的字串 S。第二行是'0'和'1'組成的字串 T。字串長度皆不會大於 100。

Output

For each case, output the case number first followed by the minimum number of moves required to convert S into T. If the transition is impossible, output `-1' instead.

對於每筆資料，先輸出此第幾筆的資料，再接著輸出移動 S 成 T 所需的最少次數。若轉換是不可能成功的，須輸出`-1'來表示之。

Sample Input

```
3
01??00
001010
01
10
110001
000000
```

Sample Output

```
Case 1: 3
Case 2: 1
Case 3: -1
```

12455　Bars

分級：2　　　　　　　　　　　　**分類：**

We have some metallic bars, theirs length known, and, if necessary, we want to solder some of them in order to obtain another one being exactly a given length long. No bar can be cut up. Is it possible?

我們有一些已知長度的金屬棒，且必要時，我們可以把其中幾根金屬棒焊接成更長的一根，以便找出所需的特定長度的金屬棒。但金屬棒不得切割，這可能嗎？

Input

The first line of the input contains an integer, t, 0 ≤ t ≤ 50, indicating the number of test cases. For each test case, three lines appear, the first one contains a number n, 0 ≤ n ≤ 1000, representing the length of the bar we want to obtain. The second line contains a number p, 1 ≤ p ≤ 20, representing the number of bars we have. The third line of each test case contains p numbers, representing the length of the p bars.

輸入的第一行含有一個整數 t（0 ≤ t ≤ 50），表示測資的數量。而每筆測資有三行，第一行有一個數字 n（0 ≤ n ≤ 1000），代表我們所想要的長度。第二行有一個數字 p（1 ≤ p ≤ 20），表示我們所擁有的金屬棒的數量，而每筆測資的第三行有 p 個數字，表示 p 根金屬棒的長度。

Output

For each test case the output should contain a single line, consists of the string YES or the string NO, depending on whether solution is possible or not.

每筆測資輸出一行，依是否可能成功來輸出「YES」或「NO」字串。

Sample Input

```
4
25
4
10 12 5 7
925
10
45 15 120 500 235 58 6 12 175 70
120
```

```
5
25 25 25 25 25
0
2
13 567
```

Sample Output

```
NO
YES
NO
YES
```

12160　Unlock the Lock

分級：2　　　　　　　　　　　　**分類：**

Mr. Ferdaus has created a special type of 4-digit lock named "FeruLock" shown in the picture on the left. It always shows a 4-digit value and has a specific unlock code (An integer value). The lock is unlocked only when the unlock code is displayed. This unlock code can be made to appear quickly with the help of some of the special buttons available with that lock. Each

button has a number associated with it. When any of these buttons is pressed, the number associated with that button is added with the displayed value and so a new number is displayed. The lock always uses least significant 4 digits after addition. After creating such a lock, he has found that, it is also very difficult for him to unlock the Ferulock. As a very good friend of Ferdaus, your task is to create a program that will help him to unlock the Ferulock by pressing these buttons minimum number of times.

Ferdaus 先生發明了一種四位數字的特殊新型鎖 "FeruLock"，如圖所示，這種鎖總是可以在其表面上看到四個數字，並且有一個特定的解鎖碼（一個整數）。當鎖面上的四個數字剛好等於解鎖碼的時候，這個鎖就會被打開。此解鎖碼可以經由一些特殊按鈕的幫助下來快速地找到，而每個按鈕也具有跟它相關的數字在上面，當這個按鈕被按下去時，按鈕上的數字就會被加到目前鎖面上顯示的數字裡，於是鎖面上就會得到一組新的數字。而這個鎖永遠都只會顯示加法後的最末位四個數字，當 Ferdaus 發明了這個鎖之後，他發現要開這個 Ferulock 鎖是一件非常困難的事情。由於身為 Ferdaus 的好朋友，所以你的任務就是寫一個程式來幫助他來打開這個 Ferulock 鎖，而且必須要讓按下按鈕的次數最少。

Input

There will be at most 100 test cases. For each test case, there will be 3 numbers: L , U and R where L (0000 ≤ L ≤ 9999) represents the current lock code, U (0000 ≤ U ≤ 9999) represents the unlock code and R (1 ≤ R ≤ 10) represents the number of available buttons. After that, there are R numbers (0 ≤ RVi ≤ 9999) in a line representing the value of buttons. The values of L , U , RVi will always be denoted by a four digit number (even if it is by padding with leading zeroes). Input will be terminated when L = U = R = 0.

輸入最多有 100 組測試資料，對每一組測試資料會有三個數字 L,U,R。首先 L(0000 ≤ L ≤ 9999)代表現在鎖面上的數字，U(0000 ≤ U ≤ 9999)代表解鎖碼，而 R(1 ≤ R ≤ 10)代表特殊按扭的數量。接下來會有 R 個數字(0 ≤ RVi ≤ 9999)，代表這些按扭上面的數值。L, U, RVi 總是會以四位數字來表示(包含前面補 0 湊到四位數)，而輸入以 L=U=R=0 時結束。

Output

For each test case, there will be one line of output which represents the serial of output followed by the minimum number of button press required to unlock the lock. If it is not possible to unlock the lock, then print a line ` Permanently Locked ' instead (without quotes).

對每一組測試資料輸出單一列數字，說明要打開這個鎖最少需要按幾次按鈕。如果這個鎖無法被打開，則輸出 Permanently Locked。

Sample Input

```
0000 9999 1
1000
0000 9999 1
0001
5234 1212 3
1023 0101 0001
0 0 0
```

Sample Output

```
Case 1: Permanently Locked
Case 2: 9999
Case 3: 48
```

195 Anagram

分級：2 **分類：組合**

You are to write a program that has to generate all possible words from a given set of letters.

Example: Given the word "abc", your program should - by exploring all different combination of the three letters - output the words "abc", "acb", "bac", "bca", "cab" and "cba".

In the word taken from the input file, some letters may appear more than once. For a given word, your program should not produce the same word more than once, and the words should be output in alphabetically ascending order.

給你一些字母，請你寫一程式算出這些字元產生的所有可能組合。

例如：給你"abc"，你的程式應該要產生這三字母的各種組合："abc"、"acb"、"bac", "bca"、"cab"和"cba"

輸入的字元可能會有重複的，但輸出請不要有重複的字串出現。字串輸出的次序請依字元次序遞增。（字元次序：AaBbCcDd.....YyZz）

Input

The input file consists of several words. The first line contains a number giving the number of words to follow. Each following line contains one word. A word consists of uppercase or lowercase letters from A to Z. Uppercase and lowercase letters are to be considered different.

輸入會有多組的測試資料，其第 1 列有一個整數 n，便代表接下來有 n 組測資。每列測試資料由從 A 到 Z 的大寫或小寫英文字元組成。大小寫請視為不同的字元。

Output

For each word in the input file, the output file should contain all different words that can be generated with the letters of the given word. The words generated from the same input word should be output in alphabetically ascending order. An upper case letter goes before the corresponding lower case letter.

對於每一筆測試資料，請輸出所有可能的不同組合。每種組合一列。輸出的次序請依字元次序遞增印出，大寫字母須在對應的小寫字母前。

Sample Input

```
3
aAb
abc
acba
```

Sample Output

```
Aab
Aba
aAb
abA
bAa
baA
abc
acb
bac
bca
cab
cba
aabc
aacb
abac
abca
acab
acba
baac
baca
bcaa
caab
caba
cbaa
```

495 Fibonacci Freeze

分級：2 **分類：大數運算**

The Fibonacci numbers (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...) are defined by the recurrence:

F0 = 0

F1 = 1

Fi = Fi-1 + Fi-2 for all i ≥ 2

Write a program to calculate the Fibonacci Numbers.

Fibonacci 數列(0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...)的定義是：

F0 = 0

F1 = 1

Fi = Fi-1 + Fi-2 for all i ≥ 2

請寫一支程式計算某一項 Fibonacci 數。

Input

The input to your program would be a sequence of numbers smaller or equal than 5000, each on a separate line, specifying which Fibonacci number to calculate.

每組測試資料一列，各有一個整數 n（0 ≤ n ≤ 5000），代表要求的第幾個 Fabonacci 數。

Output

Your program should output the Fibonacci number for each input value, one per line.

你的程式應對每組測試資料請輸出第 n 個 Fibonacci 數，一個輸出佔一行。

Sample Input

```
5
7
11
200
```

Sample Output

```
The Fibonacci number for 5 is 5
The Fibonacci number for 7 is 13
The Fibonacci number for 11 is 89
The Fibonacci number for 200 is 280571172992510140037611932413038677189525
```

11995 I Can Guess the Data Structure!

分級：2 **分類：模擬**

There is a bag-like data structure, supporting two operations:

(1 x)：Throw an element x into the bag.
(2 y)：Take out an element from the bag.

Given a sequence of operations with return values, you're going to guess the data structure. It is a stack (Last-In, First-Out), a queue (First-In, First-Out), a priority-queue (Always take out larger elements first) or something else that you can hardly imagine!

這裡有個像袋子一樣的資料結構，它提供兩種操作方式：

(1 x)：表示將元素值 x 丟進袋子中。

(2 y)：表示將元素值 y 從袋中取出。

給定一連串的操作步驟，請你猜測該資料結構為何，可能為 1. 堆疊(stack，後進先出)，2. 佇列(queue，先進先出)，3. 優先佇列(priority-queue，總是先取最大的數值)，或是其他未知的資料結構。

Input

There are several test cases. Each test case begins with a line containing a single integer n (1 ≤ n ≤ 1000). Each of the next N lines is either a type-1 command, or an integer 2 followed by an integer x. That means after executing a type-2 command, we get an element x without error. The value of x is always a positive integer not larger than 100. The input is terminated by end-of-le (EOF).

輸入有多組測試資料，每組資料的第一列有一個整數 n (1 ≤ n ≤ 1000)，接下來 n 列可能會有兩種操作，步驟的格式請參考上面的說明。在執行第 2 種操作後可以拿到 x 這個元素且不會發生錯誤，x 皆為正整數且不會超過 100。輸入以 EOF 結束。

Output

For each test case, output one of the following:

stack：It's definitely a stack.
queue：It's definitely a queue.
priority queue：It's definitely a priority queue.
impossible：It can't be a stack, a queue or a priority queue.
not sure：It can be more than one of the three data structures mentioned above.

對每組測試資料輸出其資料結構為何，可能的答案如下：

stack：表示該結構為 stack。
queue：表示該結構為 queue。
priority queue：表示該結構為 priority queue。
impossible：非以上三種。
not sure：超過一種可能的答案。

Sample Input

```
6
1 1
1 2
1 3
2 1
2 2
2 3
6
1 1
1 2
1 3
2 3
2 2
2 1
2
1 1
2 2
4
1 2
1 1
2 1
2 2
```

```
7
1 2
1 5
1 1
1 3
2 5
1 4
2 4
```

Sample Output

```
queue
not sure
impossible
stack
priority queue
```

10013 Super long sums

分級：2 **分類：大數運算**

The creators of a new programming language D++ have found out that whatever limit for SuperLongInt type they make, sometimes programmers need to operate even larger numbers. A limit of 1000 digits is so small… You have to find the sum of two numbers with maximal size of 1.000.000 digits.

有個新語言 D++ 的開發者發現，不管他們將 SuperLongInt 型態的整數上限訂到多大，程式設計師仍然會有更大的數字需要處理。1000 位數的限制實在太小…現在你就要來算出兩個最長為 1,000,000 位數的數字的和。

Input

The first line of a input file is an integer N, then a blank line followed by N input blocks. The first line of an each input block contains a single number M (1 ≤ M ≤ 1000000) the length of the integers (in order to make their lengths equal, some leading zeroes can be added). It is followed by these integers written in columns. That is, the next M lines contain two digits each, divided by a space. Each of the two given integers is not less than 1, and the length of their sum does not exceed M.

There is a blank line between input blocks.

輸入的第一行為一個整數 N，接下來在一個空行之後會有 N 筆的測試資料。而每一組測試資料的開頭有一整數 M(1 ≤ M ≤ 1000000)，表示要相加的兩個數的長度（為了令兩個數字的長度相等，需要時會加入前導 0）。接下來的 M 列每一列會有兩個字元，以空白鍵作區格，且都不會小於 1，要相加的兩個數會以直行的方式表示，它們的和也不會超過 M 位數。

每筆測試資料間會有一個空行。

Output

Each output block should contain exactly M digits in a single line representing the sum of these two integers.

There is a blank line between output blocks.

每筆測試要輸出含有剛好 M 位數的一行，代表所輸入的兩個整數的和。

每筆輸出之間要有一行空行。

Sample Input

```
2

4
0 4
4 2
6 8
3 7

3
3 0
7 9
2 8
```

Sample Output

```
4750

470
```

10015　Joseph's Cousin

分級：2　　　　　　　　　　　　　**分類：模擬**

The Joseph’s problem is notoriously known. For those who are not familiar with the problem, among n people numbered 1,2…n, standing in circle every m’th is going to be executed and only the life of the last remaining person will be saved. Joseph was smart enough to choose the position of the last remaining person, thus saving his life to give the message about the incident.

Although many good programmers have been saved since Joseph spread out this information, Joseph’s Cousin introduced a new variant of the malignant game. This insane character is known for its barbarian ideas and wishes to clean up the world from silly programmers. We had to infiltrate some the agents of the ACM in order to know the process in this new mortal game.

In order to save yourself from this evil practice, you must develop a tool capable of predicting which person will be saved.

The Destructive Process

The persons are eliminated in a very peculiar order; m is a dynamical variable, which each time takes a different value corresponding to the prime numbers’ succession (2,3,5,7…). So in order to kill the ith person, Joseph’s cousin counts up to the ith prime.

約瑟夫問題(Josephus-Problem)是非常著名的題目，不過對於這個問題不太熟悉的人我們再說明一次：有 n 個編號從 1~n 的人照順序圍成一個圓圈，每數到第 m 個人時他就必須被處決，直到最後只剩下一個人。Joseph 是個很聰明的人，他總是能挑到最後存留的位置，所以這件事才被披露出來。

雖然有許多程式設計師在 Joseph 透露這個消息之後保住了他們的性命，Joseph 的表弟提出了這個邪惡遊戲的一個新變種。這個瘋狂的傢伙以野蠻的思想出名，他想把這個世界上的愚蠢程式設計師通通消滅掉。我們已經滲透到 ACM 的某些機構裡，以得知這個新的致命遊戲是如何進行的。

為了從這個邪惡的遊戲中保住你自己的性命，你必須設計一個工具讓你有辦法找出可以得救的人所站的位置。

遊戲過程：

要被消滅的人必須經過一個非常特殊的順序，m 是一個會改變的變數，而 m 改變的規則是按照質數的順序(2,3,5,7⋯...)來變化，所以為了處決第 i 個人時，Joseph 的表弟就會算第 i 個質數這麼多次。

Input

It consists of separate lines containing n [1..3501], and finishes with a 0.

輸入有多列的測資，每一列都會有一個單獨的 n(1~3501)，當輸入為 0 時就終止。

Output

The output will consist in separate lines containing the position of the person which life will be saved.

請對每筆測資輸出於單獨一列，來印出輸出最後存活者的位置。

Sample Input

```
6
0
```

Sample Output

```
4
```

10002　Center of Masses

分級：2　　　　　　　　　　　　**分類：幾何**

Find out the center of masses of a convex polygon.

找出一個凸多邊形的重心。

Input

A series of convex polygons, defined as a number n (n ≤ 100) stating the number of points of the polygon, followed by n different pairs of integers (in no particular order), denoting the x and y coordinates of each point. The input is finished by a fake "polygon" with m (m < 3) points, which should not be processed.

測資為多組凸多邊形的資料。每組定義一個數字 n（n ≤ 100）代表這多邊形的頂點，接下來會有 n 對不同的整數，為 n 個頂點的 x 與 y 座標（並沒有照順序）。如果測資的多邊形為不合理的，也就是給定的 n < 3 ，那代表檔案結束，且不須執行這組測資。

Output

For each polygon, a single line with the coordinates x and y of the center of masses of that polygon, rounded to three decimal digits.

對於每個凸多邊形，以單行輸出其重心的 x 和 y 座標，並請以小數點後取 3 位的數值來表示之。

Sample Input

```
4 0 1 1 1 0 0 1 0
3 1 2 1 0 0 0
7
-4 -4
-6 -3
-4 -10
-7 -12
-9 -8
-3 -6
-8 -3
1
```

Sample Output

```
0.500 0.500
0.667 0.667
-6.102 -7.089
```

993　Product of digits

分級：2　　　　　　　　**分類：質數、因數與倍數**

For a given non-negative integer number N, find the minimal natural Q such that the product of all digits of Q is equal N.

給你一個大於等於 0 的整數 N，請你找到最小的自然數 Q，使得在 Q 中所有數字（digit）的乘積等於 N。

Input

The first line of input contains one positive integer number, which is the number of data sets.

Each subsequent line contains one data set which consists of one non-negative integer number N (0 ≤ N ≤ 10^9).

輸入的第一列有一個整數代表共有多少組測試資料。

每組測試資料一列有 1 個非負的整數 N（0 ≤ N ≤ 10^9）。

Output

For each data set, write one line containing the corresponding natural number Q or `-1' if Q does not exist.

對每組測試資料，請輸出自然數 Q 於一列。如果 Q 不存在，請輸出-1。

Sample Input

```
5
1
10
123456789
216
26
```

Sample Output

```
1
25
```

-1

389

-1

分級　3

147　Dollars

分級：3　　　　　　　　　　**分類：組合**

New Zealand currency consists of $100, $50, $20, $10, and $5 notes and $2, $1, 50c, 20c, 10c and 5c coins. Write a program that will determine, for any given amount, in how many ways that amount may be made up. Changing the order of listing does not increase the count. Thus 20c may be made up in 4 ways: 1×20c, 2×10c, 10c+2×5c, and 4×5c.

紐西蘭的貨幣包含了 $100, $50, $20, $10, $5 的紙鈔和 $2, $1, 50c, 20c, 10c, 5c 的硬幣。給你某金額的數字，請你寫一個程式回答：使用這些面額的紙鈔或硬幣，有多少種不同的方法可以組合成這個金額。例如 20c 可以有 4 個方法可以得到：（改變金額的順序不會增加方法數，例如 2 * 5c + 1 * 10c 和下面第 3 種方法視為同一種）

1 * 20c

2 * 10c

1* 10c + 2 * 5c

4 * 5c

Input

Input will consist of a series of real numbers no greater than $300.00 each on a separate line. Each amount will be valid, that is will be a multiple of 5c. The file will be terminated by a line containing zero (0.00).

輸入含有多組測試資料。每組測試資料一列，含有 1 個金額（不大於$300.00）。這個金額一定是合法的，也就是一定是 5c 的倍數。當輸入為 0.00 時代表輸入結束。

Output

Output will consist of a line for each of the amounts in the input, each line consisting of the amount of money (with two decimal places and right justified in a field of width 6), followed by the number of ways in which that amount may be made up, right justified in a field of width 17.

每組測試資料輸出一列，包含輸入的金額（小數點 2 位，總長度 6 位，靠右對齊）以及有多少種不同的方法可以組合成這個金額（總長度 17 位，靠右對齊）。

Sample Input

```
0.20
2.00
0.00
```

Sample Output

```
  0.20                4
  2.00              293
```