

# 重要的競賽指南!!

請仔細的閱讀以下的指南。它包含了將如何打包和提交你的答案給裁判的重要資訊。如果對於這份指南有任何問題，請在競賽開始前提問。



## 程式輸入/輸出:

需要互動的程式(task)應該要即時在螢幕上(STDOUT)要求使用者輸入，並且從鍵盤/螢幕讀取輸入(STDIN).

對於不需要互動的程式(task)，你會有二種選擇來讀取輸入資料。在一些題目當中，名為**probXX.in**的檔案將會被提供來作為樣本輸入，'XX'代表問題的號碼。

你的解答可能會需要有系統地從檔案 **probXX.in** 讀取輸入。也就是說使用程式語言中的File I/O架構。

大部分的問題會接受直接由鍵盤來輸入(STDIN) 來取代檔案運作。對於需要很多輸入的程式，這樣就變得很冗長乏味。然而，一個簡單的方式是在執行的時候借由導入檔案的內容資料到你的程式中。例如：一個名為 **prob01.in** 的檔案可以用下面的語法直接從STDIN輸入到你的程式當中：

```
%> java prob01 < prob01.in
%> java -jar js.jar prob01.js < prob01.in
%> python prob01.py < prob01.in
%> prob01.exe < prob01.in
```

在這個例題裡，你正執行著 **prob01** 以及把 **prob01.in**檔案中的資料傳送到你程式中的STDIN。你的程式行為就會如同你從鍵盤鍵入你的輸入一樣。

所有程式的輸出應該都要傳送到螢幕上(STDOUT)。

## 提交你的程式

**Interpreted Programs (JAVA, JavaScript, Python)**。你的程式一定要用 **probXX.java** / **probXX.js** / **probXX.py**來命名，'XX'代表問題的號碼。請僅提交原始碼(.java、.js 或 .py)。Java的主要類別一定要命名為 **probXX**。請注意大小寫。所有主要和支援類別都應該包含在預設的(或匿名的)套件裡。

**Native Programs (C, C++... etc)**。你的程式應該用 **probXX.exe**來命名，'XX'代表問題的號碼。

**強烈地建議你在開始競賽之前，  
先提交問題#0 (在下一頁上) 以確保你的編譯環境和裁判的是相容的。**



## 引言

這個問題主要的用意是在於允許每一個小組提交一個測試的程式，來確認由他們的電腦產生的程式可以在我們的裁判系統中被評定。強烈建議每一個小組在開始競賽之前先提交這個問題 — 嘿，這可是送分題喔！

你對這個測試程式的工作是要寫一個改良版的“Hello World!”程式。簡單地在螢幕上列印出 “First Year of CodeWars in Taiwan!”。

## 樣本輸出

```
First Year of CodeWars in Taiwan!
```





## 引言

一位在當地市場的自產自銷服裝的女士。依據材料成本她賣一件襯衫可賺8美元，在市場的攤位租金是每天95美元。寫一個程式來計算利潤總額以P表示，她在一天之內賣出的襯衫的數量以N表示。

$$P = 8 * N - 95$$

## 輸入

輸入是一天內售出襯衫的數量N。如果是業績不好，這個數字可能是零！

31

## 輸出

該程式必須列印當天的銷售利潤。根據N的值，此值可以是正的或負的。

153



## 引言

今天你可以成為一個火箭科學家！模型火箭是便宜且易於建造的，並能飛到幾千英尺的高度。

如果你想計算你的模型火箭能飛到多少海拔高度，在某些情況下，你將可能需要知道你的火箭的質量。您可以用磅秤來計算火箭的重量，但您可能發現，您的磅秤是以盎司為單位來顯示。這一切都很好，但火箭發動機的推力通常是以公制單位"牛頓"來表示。如果是這樣，在計算火箭飛行最高高度之前，您可能需要把盎司轉換成公克。

轉換可以很容易的透過下面公式來達成：

$$1 \text{ 盎司} = 28.3495 \text{ 公克}$$

## 輸入

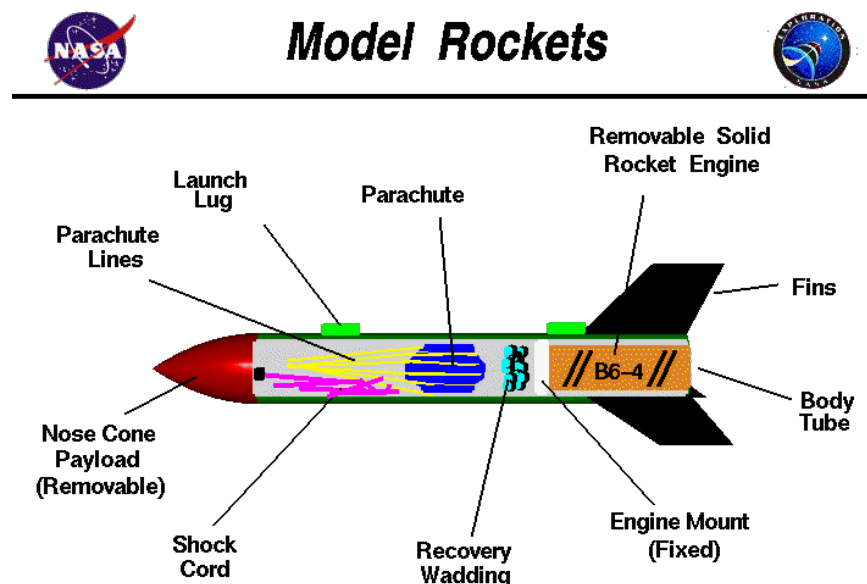
輸入是火箭的重量，單位為盎司。

11.3

## 輸出

該程式必須列印火箭的質量，以公克計，如下所示。程式的輸出值可容許與期望值有±1的誤差。

320.3496



Notice: This programming problem is not endorsed by NASA, the National Association of Rocketry, or any other rocket-related organization.



## 引言

你知道嗎？如果你站在月球上，你在月球上的重量只有在地球上的 $\frac{1}{6}$ 重。根據星球的質量和半徑，在太陽系中每個星球（月亮，行星，冥王星等）的地心引力都不一樣。

寫一個程式來計算一個人在該星球表面上的重量。

## 輸入

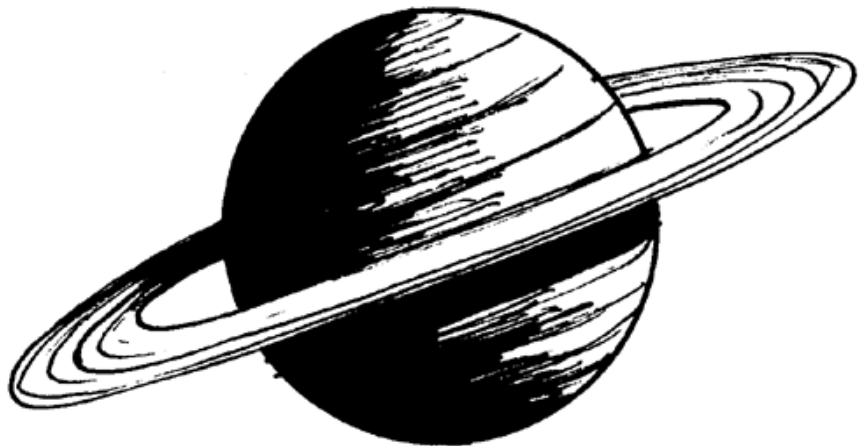
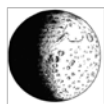
每一行依序輸入包含一個人的名字，他們的重量（以磅計，在地球上），用一個英文單字來代表一個星球的名稱，和轉換因子（地心引力）。最後一行要輸入“END”以及三個零。

```
Fred 179.0 Luna 0.1654  
Layla 131 Mars 0.376  
Pat 145.2 Neptune 1.14  
Rajavel 156.4 Ganymede 0.146  
END 0 0 0
```

## 輸出

程式必須使用以下的格式將每個轉換的重量結果列印出來。程式的輸出值可容許與期望值有 $\pm 1$ 磅的誤差。

```
On Luna, Fred would weigh 29.6066 pounds.  
On Mars, Layla would weigh 49.256 pounds.  
On Neptune, Pat would weigh 165.528 pounds.  
On Ganymede, Rajavel would weigh 22.8344 pounds.
```





## 引言

彈道學是一種討論航程、行為和拋物體效應的結構科學。你可以藉由以下公式計算出一隻憤怒鳥(或高爾夫球或其他物體)將如何飛行，如果它被從固定的平面定點丟出(發射、彈射、吐出):

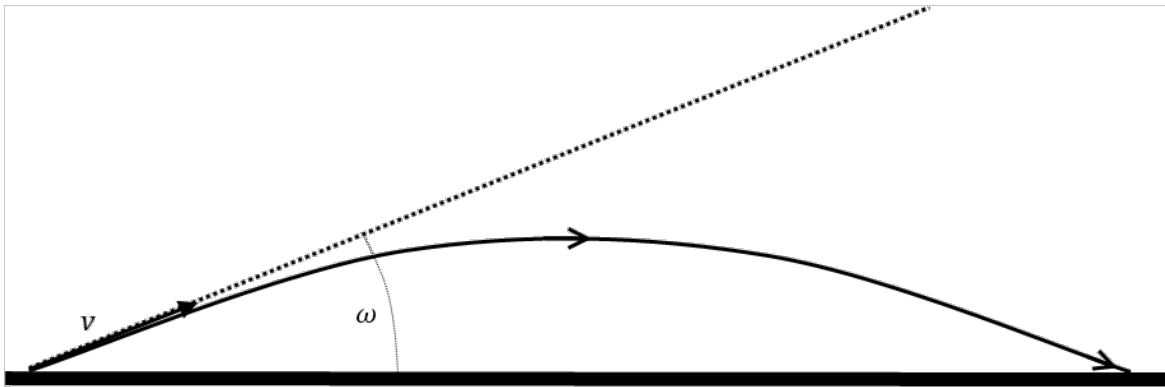
$$d = v^2 \times \sin(2\omega) / g$$

這裡 $v$ 是初速(丟出去的速度)， $\omega$ 是地面夾角， $g$ 是重力加速度，地表的重力加速度約為 $9.80665 \text{ m/s}^2$ 。

對於這個程式，我們將假設不考慮空氣阻力和地表曲度，且將以公尺、秒和度為單位。你的語言中的`sin()`副程式也許會用弧度而不是用角度。如果如此，你可以用以下的弧度轉換公式:

$$\text{radians} = \pi \times \text{degrees} / 180$$

這裡 $\pi$ 是約為 $3.14159265$



## 輸入

輸入會是兩個浮點數，且以分行區隔，第一個是拋物體的初速以公尺每秒表示，第二個是以度為單位的拋射角。

```
63.9
```

```
65
```

## 輸出

這程式必須列印出以公尺表示拋物體飛行到落地的距離。答案必須要精準到期望值的 $\pm 1$ 公尺內。

```
318.9591
```



## 引言

您可能還記得，斐波那契數列(費氏級數)是一個由0和1開始的數列，每個數字是前兩個數字的總和，也就是  $F[n] = F[n-1] + F[n-2]$ ，設  $F[0] = 0$  和  $F[1] = 1$ 。

除了起始數列是0、1、1和每個數字是前三個數字的總和之外，斐波那契數列和斐波那契數列是類似的。亦即  $T[n] = T[n-1] + T[n-2] + T[n-3]$  設  $T[0] = 0$ ,  $T[1] = 1$  和  $T[2] = 1$ 。

前11個斐波那契數列依序是0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149.

## 輸入

輸入的每一行是一個整數。最大的輸入值是30。輸入的最後一行是-1。

```
3
9
11
0
-1
```

## 輸出

對於每一個非負數的輸入，該程式必須使用整數作為輸入斐波那契數列的階數和列印該階斐波那契的數值。

```
2
81
274
0
```



## 引言

三角形面積公式如下所示:

$$\text{面積} = \frac{a \times b \times \sin w}{2}$$

$a$  與  $b$  分別為兩個邊長， $w$  則為其夾角。其中夾角  $w$  可以由下列兩個步驟求得。  
首先藉由第三個邊長  $c$  與以下的公式計算出  $\cos w$ :

$$\cos w = \frac{a^2 + b^2 - c^2}{2 \times a \times b}$$

再利用下列程式語言所支援的 `arc-cosine` 函式計算出夾角  $w$ 。

Java: `Math.acos()`  
C++: `acos()`  
Python: `math.acos()`  
JavaScript: `Math.acos()`

計算兩點距離的公式如下所示

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

## 輸入

每一行輸入將包含三個相異且不重疊的  $(x, y)$  座標。輸入六個零代表輸入終止。

```
3.1415 2.7777 -3.9123 0.2133 0.4324 -11.111
-8.675309 1.41421 9.999 0.0001 9.999 1.41421
0.7071 7.732 2.718 -1.005 -6.931 0.866025
0.6125 0.03125 99.999 0.9125 99.999 -0.56875
0 0 0 0 0 0
```

## 輸出

程式每一行將會輸出一個三角形的面積。程式的輸出值可容許與期望值有  $\pm 1$  的誤差。舉例，假如期望值為 13.2038，答案落於 12.2038 與 14.2038 將可視為正確答案。

```
45.5104
13.2038
40.2704
73.6081
```

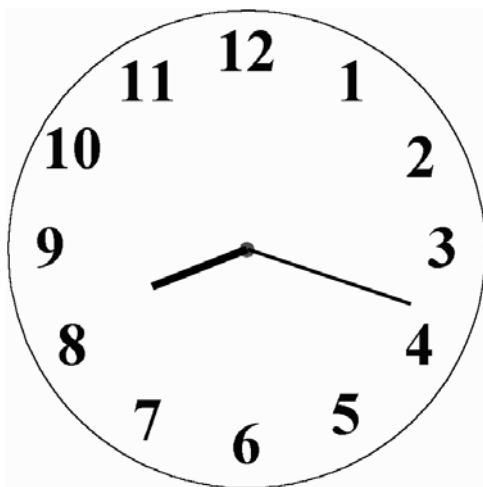




## 引言

這個程式將會根據傳統指針的時鐘算出時間。時針在每個小時之間緩慢地移動，並且不會用跳躍方式移動至下一小時。

如果時針和分針分別在數字6的兩邊而且和6有相同距離時，程式應該用所提供的分針所指的分鐘數來決定相對應的時間。



## 輸入

每一行輸入的值是一個0~59的整數。用-1當做所有輸入的結束。

```
18
5
46
0
-1
```

## 輸出

程式將對每一行的輸入印出相對應的時間，請使用 小時:分鐘為格式。個位數的小時數不需要補0當起始，但是個位數的分鐘必須要補0。

```
8:18
11:05
2:46
12:00
```



## 引言

“uncopyrightable” 有著一個有趣的特性，十五字母之中，每個字母都僅出現一次。在英文中，找不到其他更長且組成字母都絕對相異的單字了(每個字母都只僅出現一次)。

十五個字母的古希臘單字“皮膚”，“dermatoglyphics” 也是有相同特性(在古希臘文中，derma=“skin”(皮膚)，glyph=“carving”(雕刻))，這是屬於一種指紋科學研究。

請寫出一個程式可以判斷組成該單字的字母是否都是相異的。

## 輸入

請以一行一個單字的方式輸入，末行使用句號(.)做為輸入結束。

```
UNCOPYRIGHTABLE  
FLIPPER  
EXECUTABLE  
UNPROFITABLE  
QUESTIONABLY  
WINDOW  
TAMBOURINE  
.
```

## 輸出

程式隨後將印出先前輸入的每一個字以及正確資訊，“USES DISTINCT LETTERS” 代表該單字使用相異字母，或者“DOES NOT USE DISTINCT LETTERS”代表沒有相異字母。

```
UNCOPYRIGHTABLE USES DISTINCT LETTERS  
FLIPPER DOES NOT USE DISTINCT LETTERS  
EXECUTABLE DOES NOT USE DISTINCT LETTERS  
UNPROFITABLE USES DISTINCT LETTERS  
QUESTIONABLY USES DISTINCT LETTERS  
WINDOW DOES NOT USE DISTINCT LETTERS  
TAMBOURINE USES DISTINCT LETTERS
```



## 引言

這個程式是一個有趣的數學題目！

有一個函數  $f(n)$ ，它的輸入值是一個正整數  $n$ ，傳回值是從 0 到  $n$  (包含  $n$ ) 的十進位連續整數數列中 1 的個數。以輸入值  $n = 13$  為例  $f(13) = 6$ ，計算方法為 1, 10, 11 (視為兩個由 1 組成的數字), 12 和 13，得到 1 的個數為 6。此外  $f(1) = 1$ 。

你的任務為寫出此函數  $f(n)$  的程式。

## 輸入

每一行代表一個正整數輸入值，最大值為 9999。最後一行填入 -1 代表此次輸入結束。

```
13
1
999
23
1111
9997
511
-1
```

## 輸出

依序列出每一個輸入函數的正整數值的傳回值。

```
6
1
300
13
448
4000
204
```

## 引言

有五個學童分別以兩兩一對測量體重，五個學童可以配對出十組不同的體重值，提供這十組體重值，請你找出每位學童的體重為何。

你可以假設所有學童的體重均為整數。

提示: 每個學童都被秤重四次。.



## 輸入

每一行輸入十個整數值，最後一行以十個整數 0 代表輸入結束。

```
114 129 118 125 123 122 121 124 120 116
109 118 114 116 125 107 111 112 121 123
121 114 100 110 100 121 114 111 104 125
0 0 0 0 0 0 0 0 0 0
```

## 輸出

此程式必須在每一行中由小到大依序列出每位學童體重，且每位學童的體重均為整數值。

```
56 58 60 64 65
52 55 57 59 66
45 55 55 59 66
```



## 引言

"An argument is a connected series of statements intended to establish a proposition."

"No it isn't."

"Yes it is! It's not just contradiction."

"Look, if I argue with you, I must take up a contrary position."

"Yes, but that's not just saying 'No it isn't'."

"Yes it is."

"No it isn't."

寫一個程式來產生反義句。

## 輸入

第一行輸入表示接著下列輸入句子的數目。每一行的輸入會包含一個單獨的句子，這個句子會用到動詞"is"以及句號結束。

7

This is not an argument.

An argument is an intellectual process.

It is fair if you do not go.

The Ferris wheel is not working.

A butterfly is beautiful, but litter is not.

A lady discerns that which is not elegant from that which is.

A lemur is a monkey and a grivet is a monkey but a chimp is not.

## 輸出

這個程式一定要列印出每個"is"句子轉化成的反義句。也就是說，如果原本的句字在"is"之後有"not"這個字，那麼輸出就不應該有。如果原本的句字在"is"後面沒有"not"，輸出就應該在"is"後面馬上加上"not"。這樣子敘述有清楚嗎？

This is an argument.

An argument is not an intellectual process.

It is not fair if you do not go.

The Ferris wheel is working.

A butterfly is not beautiful, but litter is.

A lady discerns that which is elegant from that which is not.

A lemur is not a monkey and a grivet is not a monkey but a chimp is.

## 引言

我們來結合兩個有趣的點子: 迴文與摩斯密碼

A ● -	J ● - - -	S ● ● ●
B - ● ● ●	K - ● -	T -
C - ● - ●	L ● - ● ●	U ● ● -
D - ● ●	M - -	V ● ● ● -
E ●	N - ●	W ● - -
F ● ● - ●	O - - -	X - ● ● -
G - - ●	P ● - - ●	Y - ● - -
H ● ● ● ●	Q - - ● -	Z - - ● ●
I ● ●	R ● - ●	

Morse code



請記住，略掉空白或標點符號的迴文是由前或由後判讀都是相同的一個字串。所以單字"ATE"的摩斯密碼是"● - - ●"，這是一個迴文。同理，"AT ME"也是迴文。

## 輸入

輸入數個大寫英文單字，請以一行一個單字的方式輸入，末行使用句號(.)做為輸入結束。

```
ELEGIZED
QUIRKILY
MERCURY
FACE A WINE
HAPPY DAY
FEVER REBEL
SOPRANOS
EMIT OLD UFO TIME
PROTEIN POWDER
ANNEXING
ENJOIN
.
```

## 輸出

程式必須把單字轉換為摩斯密碼並判斷出其摩斯密碼點與破折號是否為迴文。程式必須對每一行的輸入列印出正確的訊息。"is a MCP" 是摩斯密碼迴文，or "is \*not\* a MCP"代表不是摩斯密碼迴文；請用\* 字元來輔助裁判判斷摩斯密碼迴文!

```
ELEGIZED is a MCP
QUIRKILY is a MCP
MERCURY is *not* a MCP
FACE A WINE is a MCP
HAPPY DAY is *not* a MCP
FEVER REBEL is a MCP
SOPRANOS is a MCP
EMIT OLD UFO TIME is a MCP
PROTEIN POWDER is *not* a MCP
ANNEXING is a MCP
ENJOIN is a MCP
```



## 引言

高中學校裡有用來宣傳或者公布管理部門訊息的跑馬燈。跑馬燈的高度有5像素，所以顯示的字母大小剛好符合那個空間。學校的報社有一個網路攝影機對準著跑馬燈，他們想要把上面的結果轉變成文字，然後他們就可以複製任何上面重要的佈告。

## 輸入

輸入是一個高度為5行的大標題，由大寫英文、數字和空白鍵所組成。

```
## # ## ### # # ### ### ## ### ### ## ### ### # ## # # ##
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
# # # # # ### # # ### # # # # # # # # # # # # # # # # #
# # # # # # ### # # # # # # # # # # # # # # # # # # # #
## # ## ### ### # # # # # # # # ### ### ### ### # # # ##
```

## 輸出

這個程式一定要用 ASCII string 列印出大標題。

CODEWARS 2013 ROCKS

在大標題上會被看到的文字：

```
### ## ### ### # # ### ### ### ### ###
# # # # # # # # # # # # # # # #
# # # ### ### ### ### ### # ### ###
# # # # # # # # # # # # # # #
### ### ### ### # ### ### # ### ###

### ### ## ## ### ### ## # # ### # # # # # #
# # # # # # # # # # # # # # # # # #
### ## # # # ### ### # # ### # # # # # #
# # # # # # # # # # # # # # # # # #
# # ### ## ## ### # ## # # ### ### # # ### # #

# ### ### ### ## ### # # # # # # # # # #
# # # # # # # # # # # # # # # # #
### # # ### ### ## # # # # # # # # # #
# # # # # # # # # # # # # # # # #
# # # # # # # # # # # ### # # # # #
```

為了方便，你可以在所提供的 **BannerCharaters.txt** 檔案中找到上面的文字。



## 引言

若有一個鍵盤被用來當作門的密碼鎖，這鍵盤看起來就像一個典型的電話數字鍵盤：

```
1 2 3
4 5 6
7 8 9
* 0 #
```

用戶輸入一個4位數字，並按下 # 字鍵開門。

## 輸入

輸入值為紅外掃描儀上記錄一組溫度讀數的二維矩陣，每數值分別是在鍵盤上十二個按鍵的表面溫度。溫度有所差別的原因為使用者按按鍵，讓熱力傳到按鍵上所造成的差別。

### *Example input 1*

```
72.1 75.0 75.1
72.0 72.1 72.2
72.0 72.2 77.9
72.1 72.2 75.2
```

### *Example input 2*

```
77.2 77.1 76.9
77.0 83.5 77.0
77.1 77.3 77.1
77.0 79.4 79.1
```

該程式將使用該數據來辨識其中哪四個數字被用來當密碼。對於每一組數據集會因為環境溫度和溫度差而有所不同。此外，少量的誤差（輕微的溫度變化）應該可以預期的。

## 輸出

該程式必須列印出四個數字及其所有可能的排列，並用遞增方式列出且不重複。

### *Example output 1*

```
2399
2939
2993
3299
3929
3992
9239
9293
9329
9392
9923
9932
```

### *Example output 2*

```
0555
5055
5505
5550
```





## 引言

某知名貨運公司在德州中部有一個配送中心。包裹經由輸送帶送到貨物裝卸區。在貨物被裝上卡車之前，裝卸區有一個空間可用來堆放兩堆包裹。您將編寫一個程式來配置堆疊。

當一個包裹到輸送帶的終點時，包裹必須放置在跟包裹尺寸大小接近的堆疊頂端表面。然而，如果包裹尺寸超過堆疊頂部的長度或寬度，包裹則不得被放置在該堆疊上。每個堆疊的面積為120公分\*120公分，包裹不能過大，以符合這個空間。為了符合堆疊，包裹可以向左旋轉或向右旋轉（但不能由其側邊立起）。此外，如果堆疊的高度超過200公分，包裹不可以放置在堆疊上。

如果一個包裹不能放置在堆疊上，且其他的堆疊空間是空的，那麼該包裹將成為堆疊的底部。如果該包裹可放置於現有的堆疊上，則該包裹不得放置在新的堆疊上。如果一個包裹不能被放在任何堆疊，那麼最高的堆疊被裝載到卡車上，該包裹將成為一個新堆疊的底端。所有的包裹都被處理後，任何剩餘的堆疊則以最高到最矮的排序裝入卡車。

## 輸入

輸入的第一行中以 **N** 表示需要被處理的包裹數目。之後的每一行，以公分為單位來標示每個包裹。每個包裹都以一個字母來表示，由三個整數以公分為單位依序描述：長度、寬度和高度。

### Example 1

```
13
A 59 109 49
B 84 76 58
C 91 54 42
D 43 75 38
E 35 45 25
F 76 89 40
G 26 29 16
H 102 84 46
I 75 75 75
J 71 71 51
K 101 82 42
L 78 88 58
M 77 87 57
```

### Example 2

```
14
A 101 83 67
B 82 63 72
C 58 79 70
D 81 55 39
E 80 101 65
F 57 85 51
G 75 46 68
H 97 69 63
I 65 91 64
J 56 82 72
K 65 71 69
L 44 97 63
M 91 43 65
N 41 84 72
```

## 輸出

程式必須在一行上列印出每個被裝入卡車的堆疊。堆疊必須依據被裝入卡車的順序列印。對於每個堆疊，程式必須列印出從底部到頂部的包裹標籤。

### Example 1

```
A C D E
B G
F I J
H K L
M
```

### Example 2

```
A B D
C G
H I K
L M N
E F J
```



## 引言

在撰寫程式時，有許多問題都可以用親子關係來解決。這一原理被用於在 HTML 文件的結構和階層界面的執行。這個方法是很有用的，因為程式可以將此規則應用到以繼承的概念為基礎的數據上。當然，相同的資料結構也可用於追蹤家族血緣，記錄和研究人的族譜，或選擇配種糧作。

就此題所述程式為例，兩個家長和一個或多個子女之間的關係用一個表達式表示：“A + B : C, D, E。”在這個例子中，A 和 B 是父母，而 C, D, E 是孩子。

有兩種方式詢問此程式。第一種詢問的類型是“A > D?”，它的意思是 A 是 D 的祖先嗎？第二種詢問的類型是“O ^ V”，它的意思是 O 和 V 有共同的祖先？

## 輸入

第一行輸入的是親子關係表達式的數量，跟隨其後的是每行一個表達式，每一個表達式均包含雙親和至少一個孩子。在表達式之後，緊接著的是詢問的數量，接著是一行一個詢問，字母跟符號彼此之間用一個空格分隔開來。

```
9
A + B : C , D , E .
F + G : H .
I + J : K , L .
C + H : M , N .
D + K : O .
L + E : P , Q , R , S .
N + Q : T , U , V .
O + S : W , X .
Y + H : Z .
7
K > O ?
F > W ?
B ^ U ?
O ^ V ?
A > Z ?
F > Z ?
X ^ Z ?
```

## 輸出

程式須列出每一個詢問的關係是否成立，成立的話，請記錄為 TRUE(真)，不成立為 FALSE(否)。請注意，每一個字母不會是它自己本身的雙親或祖先。

```
K > O ? TRUE
F > W ? FALSE
B ^ U ? FALSE
O ^ V ? TRUE
A > Z ? FALSE
F > Z ? TRUE
X ^ Z ? FALSE
```



## 引言

一個古怪的學者藉著大公司資助從事秘密研究已好一段時日了。最近這位學者消失且無人知其下落。警方與大學的官員認為這位古怪學者的研究紀錄筆記本也許可以找到可疑的線索，但是筆記本所記載的最近內容卻是難以理解。很明顯地這位學者以奇特的方式做研究記錄，每個單字不是刪去就是重組母音位置。請寫出一個程式可以轉換他的筆記內容成為標準的英文拼字。

為了可以完成這項任務，你的程式將需要讀取一個字典的檔案。檔名為 **wordlist.txt**，這檔案與其他用於程式競賽的輸入檔案範本放在一起，每一行皆是一個單字。

## 輸入

每一行輸入是一個由大寫字母組成的字串，某些母音字母 **A,E,I,O,U** 已經被移除，某些母音卻是被弄亂，所有剩下子音的字母是以正確的順序呈現。程式輸入不含標點符號。輸入結束請用七個大於符號。

```
TEH TSET SBJCTS AER GRWNG SSPCS NDA PSSMSTC
PRHPS EW SHULDO PEORMSI TEHM ECKA
SMTHNG AHS GOEN TRRBLY OWRNG
GLEEDOS AHS FOLEODD TEH NRCHMNT CNTEER IWTH DDLY NRTXN
>>>>>>>
```

## 輸出

程式必須破解每個輸入的字母並且印出正確的英文單字。假如兩個或兩個以上的英文單字符合條件，程式要能印出 **"?WORD1/WORD2/ETC?"**。假如沒有任何單字符合輸入字碼，程式必須印出原先輸入的字母並且前後用問號包圍住，例如 **"?PSRGCHX?"**。

```
THE TEST SUBJECTS ?ARE/EAR/ERA? GROWING SUSPICIOUS AND PESSIMISTIC
PERHAPS WE SHOULD PROMISE THEM CAKE
?SMOOTHING/SOMETHING? HAS GONE TERRIBLY WRONG
?GLEEDOS? HAS FLOODED THE ENRICHMENT ?CENTER/CENTRE? WITH ?DEADLY/ODDLY? NEUROTOXIN
```



## 引言

尋字遊戲很有趣！混和字母也很有趣！所以混和一堆字母然後來找尋字串是多麼地有趣呢！.....字串太有趣了！

讓我們從下列幾行的文字開始：

```
Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
```

下一步驟，讓我們挑個任意行數和重新地從上到下、從左到右寫入字母。在過程中我們會忽略所有非字母的字，並且把字母轉換成大寫。

```
TAVDEWNRUTEHEREGOLDOFID
WDEILODYLTTLAOARIDODNAC
OSRNLOSIDRBNNVLSAOOERO
RDGAODOCNAODEEOTNKWAAU
OIEYWAROOVTBTLNODENSSL
```

很酷吧？你現在可以找到 GOLD? TOE? HAND? RAW 這些字嗎？你可以寫一個程式來找到它們嗎？

## 輸入

第一行輸入會提供在猜字遊戲版面上的行數 (最大值為23)。在行數之後，會提供輸入文字的行數，緊接著為輸入文字。下一行則會提供搜尋字的數目，接著每一行則是搜尋的字。

```
5
4
Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
6
GOLD
NODE
TOE
MELT
RAW
HERE
```

## 輸出

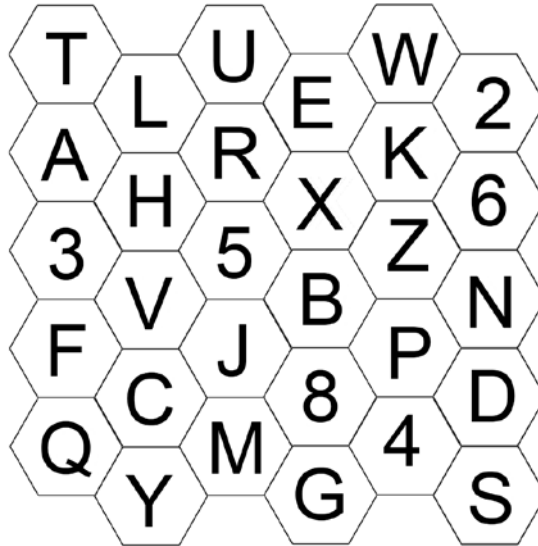
首先程式一定要列印出混和字串的版面。之後程式一定要列印出每個搜尋的文字，並在之後加上字串第一個字母(在混字版面上)的列和行。如果字串出現過一次以上，列印出所有成對的列和行並以逗號分隔每對。字串可能以水平、垂直或對角線，就像經典的字串搜尋猜謎。如果程式無法找到字串，程式一定要列印出” NOT FOUND”來取代行和列。版面的左上角是第0行和第0列。

```
TAVDEWNRUTEHEREGOLDOFID
WDEILODYLTTLAOARIDODNAC
OSRNLOSIDRBNNVLSAOOERO
RDGAODOCNAODEEOTNKWAAU
OIEYWAROOVTBTLNODENSSL
GOLD 0 15
NODE 4 14
TOE 3 15
MELT NOT FOUND
RAW 2 2, 4 6
HERE 0 11
```



## 引言

許多經典的桌上遊戲設計在六角型的格子上玩，二個六角之間的距離常常必須要被計算。寫一個程式來計算下面格子的距離。舉例來說，U 和 B 的距離是3 個六角邊，而從 A 到 M 的是4 個六角邊。



## 輸入

輸入第一行須為成對字母的數目，之後的每一行是用空白鍵分隔的一對字母。

```
5
U B
M A
3 D
R E
Q K
```

## 輸出

程式需要列印出每一對的字母以及它們之間邊數。

```
U B 3
M A 4
3 D 5
R E 1
Q K 5
```

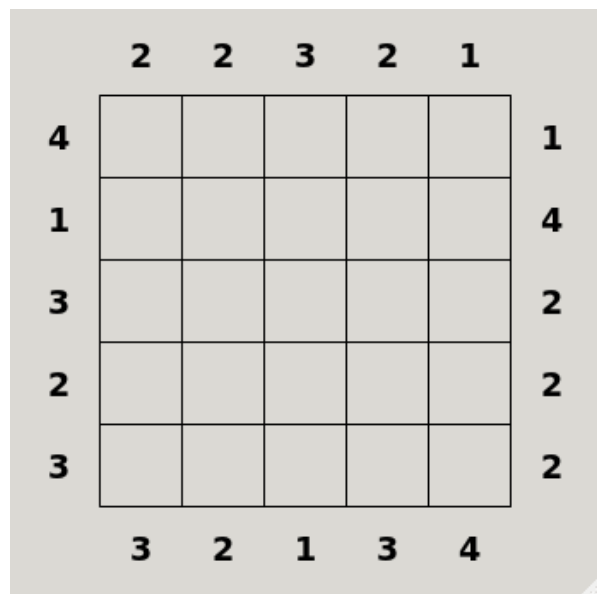


## 引言

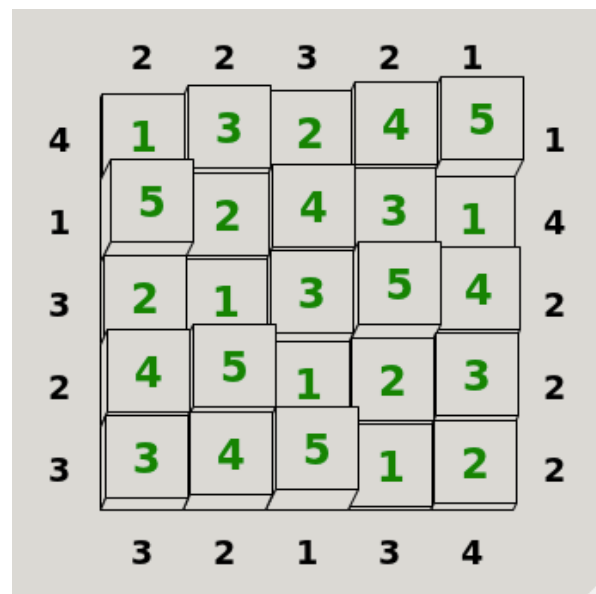
你有一個正方型的棋盤，在每一個棋盤中的方格你可以建立一個任意高度從1到棋盤大小的格子餅堆。在棋盤的邊緣有一些數字的線索，你的任務是要按照下面的方式在每一個方格中建立起格子餅堆：

- 每一堆格子餅堆的高度在每一行中只能出現一次且不能重複
- 每一堆格子餅堆的高度在每一列中只能出現一次且不能重複
- 每一個數字線索則是敘述著格子餅堆的高度排列應該如何放置。如果你從數字線索上的方向來看棋盤，假設較低格子餅堆會被較高的格子餅堆遮住。例如：在5X5的棋盤上，數字為'5'的線索就是表示這個方向上的五個格子餅堆必須以漸增順序排列(否則你將沒有辦法看到所有五個威化餅堆)，而數字為'1'的線索就表示最高的格子餅堆(就是那個標記為5的)必須放在第一個位置。

## 輸入



## 輸出



第一行輸入表示最上面每一列的數字線索，接下來五行的輸入表示棋盤左邊和右邊的每一列數字線索。最後一行輸入表示最下面每一列的數字線索。

2 2 3 2 1

4 1

1 4

3 2

2 2

3 2

3 2 1 3 4

你的程式需要印出5X5的棋盤且每一列當中用空白將每一格隔開。

1 3 2 4 5

5 2 4 3 1

2 1 3 5 4

4 5 1 2 3

3 4 5 1 2

## 引言

所謂的頭韻就是一連串單字或片語的第一音節為重複特殊發音，例如 "If any so hardy in this house holds himself"。重複子音發音於字中或字尾，有時候是被稱之為作"子音韻"這樣的專有名詞，但是"頭韻"用於這個我們這次要寫得競賽程式中，不論子音字母位在於何處皆可稱之。

傳統頭韻已被用於詩或文學上，現今它最常用於歌詞但是也見於雜誌文章的標題或廣告，大致上是很多方面的呈現。

寫一個程式分析頭韻。記住頭韻是關於發音而非字母本身，所以程式要保留下列規則：

- 第一個子音發音算作3倍，即使單字由母音開始
- 雙子音看作一個(所以 "mammal" 有兩個 "m" 發音)
- c 在 e,i 或 y 之前看作 s
- c 在 a,o 或 u 之前看作 k
- c 在單字節尾之前看作 k
- ck 看作為單一個 k
- q 看作 k
- x 看作 k 和 s 兩個
- ch 不同於 c 或 h
- sh 看作 s
- th 不同於 t 或 h
- tch 看作 ch
- gh 看作 h
- wh 看作 w

## 輸入

輸入包含數行英文字，每一行字用區隔的 ~ 符號字元，而且最後一行輸入要用單一個 ~ 符號。

```
Once upon a midnight dreary while I pondered weak and weary ~  
Mary sat musing on the lamp-flame at the table ~  
Carrie's cat clawed her couch, creating chaos ~  
The silken sad uncertain rustling of each purple curtain ~  
Hot hearted Beowulf was bent upon battle ~  
The daily diary of the American dream ~  
The famous families lived along eleven fancy lanes ~  
Charmed rich duchesses cherish diamonds ~  
~
```

## 輸出

程式必需根據以上的規則，列印出每一行的支配的子音發音以及發音次數。如果發音次數相同，列印出所有支配的子音發音。為了評審方便，請轉換所有字母為大寫的方式

```
W 9  
M 8  
K 15  
S 8  
B 9  
D 9  
L 13  
CH D 8
```

