

12626	I ♥ Pizza	4
12611	Beautiful Flag	6
12602	Nice Licence Plates.....	9
12545	Bits Equalizer.....	11
12503	Robot Instructions.....	13
12468	Zapping.....	15
12455	Bars	17
12439	February 29.....	19
12416	Excessive Space Remover	21
12405	Scarecrow.....	23
12372	Packing for Holiday	25
12289	One-Two-Three	27
12250	Language Detection	29
12160	Unlock the Lock.....	31
12149	Feynman.....	33
12136	Schedule of a Married Man	35
12019	A - Doom's Day Algorithm.....	38
11995	I Can Guess the Data Structure!.....	41
11991	Easy Problem from Rujia Liu?	44
11988	Broken Keyboard (a.k.a. Beiju Text)	46
11965	Extra Spaces	48
11953	Battleships.....	50
11946	Code Number.....	52
11942	Lumberjack Sequencing.....	55
11917	Do Your Own Homework!	57
11898	Killer Problem.....	60
11879	Multiple of 17	62
11850	Alaska	64
11849	CD.....	66
11847	Cut the Silver Bar	68
11830	Contract Revision	70
11827	Maximum GCD	72
11824	A Minimum Land Price.....	73
11764	Jumping Mario	75
11742	Social Constraints.....	77
11730	Number Transformation	79
11727	Cost Cutting.....	81
11716	Digital Fortress	83

11689	Soda Surpler	85
11677	Alarm Clock	87
11636	Hello World!	89
11608	No Problem	91
11597	Spanning Subtree	93
11586	Train Tracks.....	94
11579	Triangle Trouble	97
11577	Letter Frequency	99
11538	Chess Queen	100
11494	Queen.....	102
11479	Is this the easiest problem?	104
11461	Square Numbers	106
11455	Behold My Quadrangle	107
11396	Claw Decomposition	109
11388	GCD LCM	112
11342	Three-square.....	114
11321	Sort! Sort!! and Sort!!!.....	116
11222	Only I did it!	119
11188	Ternary	121
11094	Continents.....	122
11054	Wine trading in Gergovia	124
11044	Searching for Nessy.....	126
11039	Building designing.....	128
11000	Bee	130
10931	Parity.....	132
10922	2 the 9s.....	134
10908	Largest Square.....	136
10905	Children's Game	138
10814	Simplifying Fractions.....	140
10394	Twin Primes.....	141
10192	Vacation	143
10066	The Twin Towers	146
10055	Hashmat the Brave Warrior	148
10023	Square root	150
10004	Bicoloring	151
846	Steps.....	154
725	Division.....	156
612	DNA Sorting.....	158

572	Oil Deposits	160
495	Fibonacci Freeze.....	162
494	Kindergarten Counting Game	164
490	Rotating Sentences	166
406	Prime Cuts.....	168
401	Palindromes	170
397	Equation Elation	173
382	Perfection.....	176
374	Big Mod.....	178
305	Joseph	180
294	Divisors.....	182
264	Count on Cantor	184
256	Quirky Squares	186
245	Uncompress	188
195	Anagram.....	191
151	Power Crisis.....	194
147	Dollars	196
141	The Spot Game.....	198
136	Ugly Numbers	200
118	Mutant Flatworld Explorers	201
105	The Skyline Problem	204

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:

AAAAAAMMRRTITIGRRRRRRRR

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：

AAAAAAMMRRTITIGRRRRRRRR

然後我們就可以做兩塊 MARGARITA 披薩，仍還剩餘一些 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
AAAAAAMMRRTITIGRRRRRRRR
AMARGITA
BOLOGNESACAPRICHOSATOMATERA
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Sample Output

```
1
2
0
1
0
```

Analysis

將題目所提之字母，加以統計並計算字元量即可得解。

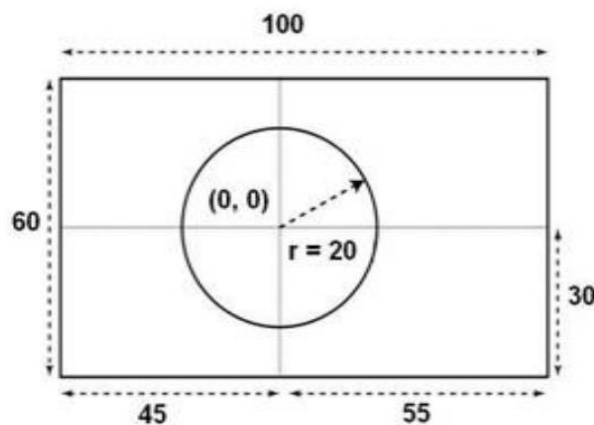
12611 Beautiful Flag

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國的國旗，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：

孟加拉國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。你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

Analysis

根據題目給定的比例方式，將長、寬求出，便可直接輸出四個角的點座標。

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 ($0 \cdot 26^2 + 1 \cdot 26^1 + 2 \cdot 26^0$). So, the plate " ABC-0123 " is nice, because $|28 - 123| \leq 100$.

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 $28(0 \cdot 26^2 + 1 \cdot 26^1 + 2 \cdot 26^0)$ ，且 $|28 - 123| \leq 100$ ，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 ($1 \leq N \leq 100$), 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 ($1 \leq N \leq 100$)，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

Analysis

此題因有規定的格式，所以只需懂得如何抓取字元的程式碼，再按照題意來計算即可。

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 ($C \leq 200$) 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 ($C \leq 200$)，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
```

Analysis

此題為簡單 greedy 題，選擇操作順序為 2->1->3，先將 S 轉換成與目標字串相同的字元，再將須轉換的 0 改成 1，最後檢查位置不同個數即可。

12503 Robot Instructions

分級：1

分類：模擬

You have a robot standing on the origin of x axis. The robot will be given some instructions. Your task is to predict its position after executing all the instructions.

- LEFT: move one unit left (decrease p by 1, where p is the position of the robot before moving)
- RIGHT: move one unit right (increase p by 1)
- SAME AS i: perform the same action as in the i-th instruction. It is guaranteed that i is a positive integer not greater than the number of instructions before this.

你有一台機器人站在 X 軸的原點上，該機器人將被賦予一些指令。你的任務是所有指令執行完後，預測它的位置。

- LEFT：向左移動一個單位（指機器人的位置 p 比移動前的位置減少 1 個單位）
- RIGHT：向右移動一個單位（位置 p 增加 1 個單位）
- SAME AS i：執行與第 i 個指令相同的動作。i 為正整數，且不大於在此之前的指令數量。

Input

The first line contains the number of test cases T ($T \leq 100$). Each test case begins with an integer n ($1 \leq n \leq 100$), the number of instructions. Each of the following n lines contains an instruction.

第一行包含數字 T ($T \leq 100$)，指測試用例的數量。每筆測試資料的開始有一個整數 n ($1 \leq N \leq 100$)，表示指令的數量，說明以下 n 行皆包含著一個指令。

Output

For each test case, print the final position of the robot. Note that after processing each test case, the robot should be reset to the origin.

對於每筆的測試資料，皆須印出機器人的最終位置。需要注意的是處理每筆資料後，機器人會重新回到原點。

Sample Input

2

3

LEFT

RIGHT

SAME AS 2

5

LEFT

SAME AS 1

SAME AS 2

SAME AS 1

SAME AS 4

Sample Output

1

-5

Analysis

簡單題，直接按照題目說明模擬即可。

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 \triangle 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 \triangle 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 \triangle 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 (\triangle 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 \triangle 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 \triangle 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 ($0 \leq a, b \leq 99$).

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 ($0 \leq a, b \leq 99$)。

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 \triangle 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 \triangle 和 ∇ 兩個按鈕可用)。

Sample Input

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

Sample Output

```
6
1
15
```

Analysis

按照題意將兩個方向算完，再找最小即可得解。

12455 Bars

分級：2

分類：

We have some metallic bars, their 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 , $0 \leq t \leq 50$, indicating the number of test cases. For each test case, three lines appear, the first one contains a number n , $0 \leq n \leq 1000$, representing the length of the bar we want to obtain. The second line contains a number p , $1 \leq p \leq 20$, 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 ($0 \leq t \leq 50$)，表示測資的數量。而每筆測資有三行，第一行有一個數字 n ($0 \leq n \leq 1000$)，代表我們所想要的長度。第二行有一個數字 p ($1 \leq p \leq 20$)，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

Analysis

此題規定值範圍較小，直接按照題目去窮舉即可。

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

Analysis

此題不能一年一年去判斷，效率太差。將第二個日期前發生過的閏年數，再減去第一個日期所發生數，即可得解。

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
```

Analysis

此題不需要真的對輸入去模擬『全部取代』的動作，『全部取代』的次數取決於輸入中最長的那一段連續空白，因此『全部取代』的次數就是最長的那一段連續空白字元的個數取 \log 的結果。

12405 Scarecrow

分級：1

分類：模擬

Taso owns a very long field. He plans to grow different types of crops in the upcoming growing season. The area, however, is full of crows and Taso fears that they might feed on most of the crops. For this reason, he has decided to place some scarecrows at different locations of the field.



The field can be modeled as a $1 \times N$ grid. Some parts of the field are infertile and that means you cannot grow any crops on them. A scarecrow, when placed on a spot, covers the cell to its immediate left and right along with the cell it is on.

Given the description of the field, what is the minimum number of scarecrows that needs to be placed so that all the useful section of the field is covered? Useful section refers to cells where crops can be grown.

Taso 有一塊非常長的條狀田地。他計劃在下一季種植不同的作物。但是這個地區有很多烏鴉，Taso 怕他們會吃掉大部分的作物。因此他決定在田地的不同位置放置稻草人。

田地可以用 $1 \times N$ 的格子模擬。有些部分是不毛之地無法種植作物。一個稻草人除了所在的格子外也保護其左側及右側緊鄰的格子。

已知田地的描述，要保護所有可用田地最少需要幾個稻草人？可用地意指作物可生長之地。

Input

Input starts with an integer T (≤ 100), denoting the number of test cases.

Each case starts with a line containing an integer N ($0 < N < 100$). The next line contains N characters that describe the field. A dot (.) indicates a crop-growing spot and a hash (#) indicates an infertile region.

輸入開始有一整數 T (≤ 100)，表示測資筆數。

每筆測資第一行有一整數 N ($0 < N < 100$)，下一行便有 N 個字元以描述田地。
點 (.) 表示良田，井號 (#) 表示不毛之地。

Output

For each case, output the case number first followed by the number of scarecrows that need to be placed.

對於每筆測資，先輸出測資編號，再輸出需放置的稻草人數量。

Sample Input

```
3
3
.#.
11
...##...##
2
##
```

Sample Output

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

Analysis

此題的田地為一維陣列，所以從最左邊開始遇到田地就放置，而一個稻草人能守住 3 格的地方（其與其+1 和其+2），因此類推便可求解。

12372 Packing for Holiday

分級：1

分類：模擬

Mr. Bean used to have a lot of problems packing his suitcase for holiday. So he is very careful for this coming holiday. He is more serious this time because he is going to meet his fiancée and he is also keeping frequent communication with you as a programmer friend to have suggestions. He gets confused when he buys a gift box for his fiancée because he can't decide whether it will fit in his suitcase or not. Sometimes a box doesn't fit in his suitcase in one orientation and after rotating the box to a different orientation it fits in the suitcase. This type of behavior makes him puzzled.

So to make things much simpler he bought another suitcase having same length, width and height, which is 20 inches. This measurement is taken from inside of the box. So a box which has length, width and height of 20 inches will just fit in this suitcase. He also decided to buy only rectangular shaped boxes and keep a measuring tape in his pocket. Whenever he chooses one gift box, which must be rectangular shaped, he quickly measures the length, width and height of the box. But still he can't decide whether it will fit in his suitcase or not. Now he needs your help. Please write a program for him which calculates whether a rectangular box fits in his suitcase or not provided the length, width and height of the box. Note that, sides of the box must be parallel to the sides of the suitcase.

憨豆先生曾經因為假期，而有很多收拾行李的問題，所以他很謹慎地對這即將到來的假日，尤其是這次他要去見他的未婚妻，而他一直都跟你這位程式設計師有著頻繁的聯繫以便你給他一些建議。他現在很困惑，因為當他為他的未婚妻買了禮物盒，他不能確定他的行李箱是否裝得下它。而讓他十分不解的是，有時箱子裝不下他的行李箱，但將它旋轉到不同的方向卻又裝得下。

因此，為了使事情變得簡單，他買了有相同的長度，寬度和高度的另一個 20 英寸行李箱。它的測量方法是從內部來測的，因此，一個有 20 英寸的長度、寬度和高度的箱子，將剛好適合這個行李箱。不過，他還是決定只買矩形的盒子和攜帶捲尺在他的口袋裡，所以每當他選擇一個長方形的禮品盒，他便能夠迅速測量盒子的長度，寬度和高度。可是他仍然不能決定它是否裝得下他的手提箱。現在，他需要你的幫助來寫一個程式，設置箱子的長度，寬度和高度，計算其是否適合他的手提箱。需要注意的是，盒子的每一面都要與手提箱的每一面平行。

Input

Input starts with an integer T ($T \leq 100$), which indicates the number of test cases.

Each of the next T line contains three integers L , W and H ($1 \leq L, W, H \leq 50$) denoting the length, width and height of a rectangular shaped box.

輸入開始有一整數 T ($T \leq 100$)，表示測資筆數。

每行測資皆有三整數 L 、 W 和 H ($1 \leq L, W, H \leq 50$)，分別表示長方形盒子的長寬高。

Output

For each test case, output a single line. If the box fits in the suitcase in any orientation having the sides of the box is parallel to the sides of the suitcase, this line will be “Case #: good”, otherwise it will be “Case #: bad”. In your output # will be replaced by the case number.

Please see the sample input and sample output for exact format.

對於每筆測資，需結果輸出於單行。如果盒子以任一方向(每一面平行與行李箱的面)裝得進去行李箱的話，輸出將是 “Case #: good”，反之則為 “Case #: bad”。在你的輸出中，#會取代成測資編號。

Sample Input

```
2
20 20 20
1 2 21
```

Sample Output

```
Case 1: good
Case 2: bad
```

Analysis

簡單題。直接判斷是不是三維都不超過 20 即可。

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
thee
```

Sample Output

1

2

3

Analysis

由於只會出現 1,2,3 這三種可能性，所以可以就長度、形狀…，直接設條件來判斷最像三個之中哪一個即可。

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
```

Analysis

直接按題目說明內建字串來比對即可。

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 ($0000 \leq L \leq 9999$) represents the current lock code, U ($0000 \leq U \leq 9999$) represents the unlock code and R ($1 \leq R \leq 10$) represents the number of available buttons. After that, there are R numbers ($0 \leq RV_i \leq 9999$) in a line representing the value of buttons. The values of L , U , RV_i 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 $L(0000 \leq L \leq 9999)$ 代表現在鎖面上的數字， $U(0000 \leq U \leq 9999)$ 代表解鎖碼，而 $R(1 \leq R \leq 10)$ 代表特殊按鈕的數量。接下來會有 R 個數字 $(0 \leq R_i \leq 9999)$ ，代表這些按鈕上面的數值。 L, U, R_i 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系列數字，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
```

Analysis

使用 BFS 搜尋：將各組數字運算結果，存到 `push` 到 `queue` 裡等待下一步展開，同時記錄按鈕的次數。因為只取末四位數，要記錄以避免重複出現的 BFS 展開。

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 $N \times N$ 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 $N \times N$ 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 ($1 \leq N \leq 100$).

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 ($1 \leq N \leq 100$)。

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
```

Analysis

將此題以 $N=1、2、3\cdots$ 展開，推導後便發現題目為 $1^2 + 2^2 + \cdots + n^2$ ，所以利用 $n(n+1)(2n+1)/6$ 之公式即可。

12136 Schedule of a Married Man

分級：1

分類：排序

Last year we set a problem on bachelor arithmetic which made some bachelors really unhappy. So to even things up, we are making a problem on the tough schedule of a married man.

Our dashing hero Danny has recently got married and that has created a lot of problems for him, at least that is what his friends think. So many broken promises, so many missed appointments and dinners. Err! Danny, now is losing tracks of even simplest of calculations, so you must help him to decide whether he can attend his



meeting or not. Danny is busy with his wife for a large portion of the day. This large portion is denoted by a starting time and an ending time. Then Danny has an important meeting in a day, he misses that if it overlaps or touches (For example, if Danny's time span with his wife finishes at 18:00 and the meeting starts at 18:00 then the two schedules conflict and Danny misses the meeting) the time scheduled for his wife. Given the time span Danny has allotted for his wife and the time span of the meeting ,you will have to find whether Danny misses that meeting or not.

去年，我們設置了一些讓單身漢很不愉快的數學問題。所以把事情扯平，我們另外以一個已婚男人的艱苦的計畫來做一個問題。

我們瀟灑的主人公 Danny 最近結婚了，並且對他來說產生了很多的問題，至少他的朋友這麼認為。許多破碎的承諾，許多錯過的約會和晚餐。更糟糕的是，Danny 現在正失去最簡單的估計能力，因此你一定要幫忙決定他是否能參加他的會議。Danny 一天中的很大一部分的時間是忙於他妻子的(這個一大部分包含其開始與結束時間)。某一天 Danny 有一個重要的會議，若它剛好重疊在與妻子相處的時間，他就只能錯過了（例如，如果 Danny 與妻子相處結束時間為 18:00，而會議開始時間為 18:00，因此這兩個時間衝突到，Danny 便會錯失了會議）。由於 Danny 的時間已經分配給他的妻子和會議，你將須找出 Danny 是否會錯過會議。

Input

First line of the input file contains an integer N ($0 < N < 2001$) which denotes how many sets of inputs are there. The input for each set is given in two lines. The description for each set is given below:

First line of each set contains two strings separated by a single space. These two strings denote the time span Danny is busy with his wife. The second line also contains two strings which denotes the time when Danny has to attend a meeting. All the strings that denote time are of the format hh:mm (two digit for hour and two digit for minute). For example “forty five past eight” (Morning) is denoted as ‘08:45’, ”forty five past 9” (night) is denoted as ‘21:45’. You can assume that all times are valid 24-hour clock time, starting time strictly precedes ending time and all times are within a single day.

輸入的第一行包含著一個整數 N ($0 < N < 2001$)，代表測資的數量。每筆測資為兩行，說明如下：

第一行包含二個以空白間格的字串，代表著 Danny 劃分給妻子的時間。第二行同樣包含二個字串，代表著 Danny 參加會議的時間。每個時間的格式為 hh:mm（兩數值表示小時，兩數值表示分鐘）。例如，早上八點過四十五分表示 08:45，而晚上九點過四十五分表示 21:45。已知時間為 24 小時制，開始時間一定早於結束時間，且皆在同一天。

Output

For each set of input produce one line of output. This line contains the serial of output followed by a string which denotes Danny’s decision. If Danny can attend the meeting then print ‘Hits Meeting’ and if Danny misses the meeting as it conflicts with the time allotted for his wife print ‘Mrs Meeting’ instead.

每筆測資須輸出於一行，並說明 Danny 的決定。如果 Danny 可以參加會議，就印 ‘Hits Meeting’，但若因時間衝突而無法參加，則印 ‘Mrs Meeting’。

Sample Input

```
3
17:47 22:40
06:18 17:04
10:44 17:05
01:11 01:27
```

03:36 19:02

14:33 15:24

Sample Output

Case 1: Hits Meeting

Case 2: Hits Meeting

Case 3: Mrs Meeting

Analysis

由於此題的時間都在同一天的 24 小時內，因此可以透過計算總時間來判斷是否有衝突，會比較容易些。

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

Analysis

由於此題的時間都在同一年，可以先找出 12 月 31 日是星期幾，再以陣列窮舉，並記錄所有日期的星期數，便可直接讀入月份和日期來查詢解答。

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 ($1 \leq n \leq 1000$). 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 ($1 \leq n \leq 1000$)，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

Analysis

使用三種結構去模擬輸入的結果，檢查哪個結構的結果與輸入相符即可。

11991 Easy Problem from Rujia Liu?

分級：2

分類：數學計算

Though Rujia Liu usually sets hard problems for contests (for example, regional contests like Xi'an 2006, Beijing 2007 and Wuhan 2009, or UVa OJ contests like Rujia Liu's Presents 1 and 2), he occasionally sets easy problem (for example, 'the Coco-Cola Store' in UVa OJ), to encourage more people to solve his problems :D

Given an array, your task is to find the k -th occurrence (from left to right) of an integer v . To make the problem more difficult (and interesting!), you'll have to answer m such queries.

雖然 Rujia Liu 通常給比賽設計較困難的題目（例如 2006 年西安地區、2007 年北京和 2009 年的武漢，或是 UVa OJ 的比賽中的'Rujia Liu 的禮物 1 和 2'），他偶爾也會設計簡單的題目（例如 UVa OJ 裡的'the Coco-Cola Store'），以鼓勵更多的人來解決他的問題 :D

給定一個陣列，你必需找出某一特定的整數 v 在此中重複出現第 k 次時的序號（指在此陣列中的序號，以 1 開始），為了讓題目更困難（和更有趣！），你需要以查詢方式回答答案 m 。

Input

There are several test cases. The first line of each test case contains two integers n, m ($1 \leq n; m \leq 100,000$), the number of elements in the array, and the number of queries. The next line contains n positive integers not larger than 1,000,000. Each of the following m lines contains two integer k and v ($1 \leq k \leq n, 1 \leq v \leq 1000000$). The input is terminated by end-of-file (EOF).

輸入會有許多組測試資料，每組資料的第一列有兩個整數 n, m ($1 \leq n, m \leq 100,000$)， n 表示有多少元素在陣列中，再接下行就有 n 個小於 1,000,000 的正整數。接著有 m 行皆有一組 k 和 v 值 ($1 \leq k \leq n, 1 \leq v \leq 1,000,000$)，請依題目回答，輸入以 EOF 為結束。

Output

For each query, print the 1-based location of the occurrence. If there is no such element, output '0' instead.

對於每筆問題，請以 1 為第一個位置輸出要求的序號，如果不存在請輸出 0。

Sample Input

```
8 4
1 3 2 2 4 3 2 1
1 3
2 4
3 2
4 2
```

Sample Output

```
2
0
7
0
```

Analysis

題目的規模是 10^6 如果直接用暴力的話那麼 TLE 是肯定的，所以這裡應該考慮用 C++ 裡的 `vector`(STL) 來做 Map，這樣才能做到二維暴力以找解。

11988 Broken Keyboard (a.k.a. Beiju Text)

分級：1

分類：模擬

You're typing a long text with a broken keyboard. Well it's not so badly broken. The only problem with the keyboard is that sometimes the "home" key or the "end" key gets automatically pressed (internally).

You're not aware of this issue, since you're focusing on the text and did not even turn on the monitor! After you finished typing, you can see a text on the screen (if you turn on the monitor).

In Chinese, we can call it Beiju. Your task is to find the Beiju text.

你用了一個壞掉的鍵盤來打一長串的字，好吧它是沒有那麼的嚴重破損，該鍵盤唯一的問題是會自動按下” Home” 鍵與”End”鍵。

直到打完整個內容以前，你專注於打字，都沒有發現到這個問題，甚至忘記打開螢幕！當然如果你有開是看得到文字的。

在中國，我們可以稱它為 Beiju(杯具)。你的任務就是找出這 Beiju text。

Input

There are several test cases. Each test case is a single line containing at least one and at most 100,000 letters, underscores and two special characters `[` and `]`. `[` means the "Home" key is pressed internally, and `]` means the "End" key is pressed internally. The input is terminated by end-of-file (EOF).

輸入有多組測試資料，每組一行，其長度介於 1 ~ 100,000 之間，包含小寫的字母及兩個符號 '[' 與 ']'， '[' 表示被自動按下的 Home 鍵； ']' 表示 End 鍵。輸入資料以 EOF 表示結束。

Output

For each case, print the Beiju text on the screen.

對於每筆測資，請你輸出該字串在螢幕上顯示的內容。

Sample Input

This_is_a_[Beiju]_text

[[[]]]Happy_Birthday_to_Tsinghua_University

Sample Output

BeijuThis_is_a__text

Happy_Birthday_to_Tsinghua_University

Analysis

按照題意：若輸入 '['，就跑回起點；若輸入 ']'，就跑到終點。所以可用鏈表模擬插入的過程，最後再沿鏈表將各個字元輸出即可。

11965 Extra Spaces

分級：1

分類：字元與字串

In programming multiple whitespaces are used to only to make code more readable, so mostly all programming languages totally ignore multiple spaces in code (except for some esoteric ones). In general there are different types of whitespace characters: space itself, tabs, newline symbol, various control characters, etc. Tabs and spaces bring one or the biggest holywar to a programmers world as there is no common rule what to use for code indentation – tab or space characters.

In this holywar you stand for tab side and your project code convention requires to use only them for code indentation. However you have recently spotted that someone is using space characters instead of it. Four spaces and tab character look the same in our text editor, so you have decided to write a parser that will change all consequent space characters to one. After that you would be able to determine amount of corrupted code.

寫程式時，連續的空白字元習慣被使用來增加程式碼的可讀性，且大多數的程式語言都會忽略程式碼中的連續多個空白字元，而一般來說，空白字元也有不同種的表示方式，例如空白字元本身、Tab、換行字元，及其他控制字元等等。其中 Tab 與空白鍵的使用一直以來都存有爭議。

你是 Tab 派的，你的程式碼之首行縮排一定會用 Tab，但是你最近發現有人用上了一堆空白鍵來取代之。而在程式編輯器上，四個空白鍵看起來與一個 Tab 鍵一樣，所以你決定寫一個程式把所有連續的空白鍵全都換成一個空白鍵，這樣才能容易看出誤用的地方在哪裡。

Input

The number of tests T ($T \leq 100$) is given on the first line. At first line of each test there is integer N ($N \leq 50$). Next N lines with text that must be processed for extra spaces follow. Maximal line length is equal to 500 characters.

第一列有一個整數 T ($T \leq 100$) 表示測試資料的組數。每組測資的第一行有一個整數 N ($N \leq 50$) 表示接下來有 N 列原始碼要處理連續的空白，每列最多不超過 500 個字元。

Output

For each test case output a single line `Case T :`. Where T is the test case number (starting from 1). Next N lines must be output with input text having no consequent spaces. By the way, always leave a blank line between tests. Please refer to the sample output for clarity.

對於每組測試資料，請先輸出單行的"Case T:"（T 表示測試資料的編號），再去除資料內連續的空白後輸出。每組測試資料間請以一個空白行隔開，請參考範例資料。

Sample Input

2

3

Sample test one:

there was 2 spaces and
here are also 2 spaces

2

Sample test two:

there was 4 spaces

Sample Output

Case 1:

Sample test one:

there was 2 spaces and
here are also 2 spaces

Case 2:

Sample test two:

there was 4 spaces

Analysis

按照題意：若輸入 '['，就跑回起點；若輸入 ']'，就跑到終點。所以可用鏈表模擬插入的過程，最後再沿鏈表將各個字元輸出即可。

11953 Battleships

分級：2

分類：模擬

Battleships game is a pen and paper game that was invented by Clifford Von Wickler in the early 1900s. In this game each player uses two $N \times N$ grids. One to arrange his ships and record the shots of the opponent. On the other grid the player records his own shots. Ships in battleship game can vary in size from 1×1 to $1 \times N/2$ and can be placed both vertically and horizontally. When all of the ship's cells have been hit, the ship is considered sunk, otherwise it is still "alive". Beside this, there can be more than one ship of each size, however none of two ships can overlap or touch.

In this problem you will be given the placement of ships on the player's grid. You will have to calculate the number of ships that the player still owns.

「戰艦遊戲」為 Clifford Von Wickler 於二十世紀初發明的紙上遊戲，遊戲中每位玩家分別有兩張 $N \times N$ 的方格，來安排我方戰艦的位置與紀錄被敵方炮擊的位置。在戰艦遊戲中，船的大小為 1×1 到 $1 \times N/2$ ，且可垂直或水平放置。當戰艦的每一位置都被炮擊則該戰艦就沈沒了，反之為"存活"。除了這點，我方可能會有多艘大小不一的戰艦，但不會有任兩艘戰艦重疊或相互接觸。

在本題中，你會得到安置戰艦位置的方格資訊，請你計算還有幾艘戰艦存活在戰場上。

Input

There is a number of tests T ($T \leq 100$) on the first line. Each test case contains a positive number N ($N \leq 100$) — grid size. Next N lines contain N characters each, describing the playing grid. Character `.` stands for an empty cell, `x` for a cell containing a ship or its part and `@` for already hit part of a ship.

輸入資料的第一列有一個整數 T ($T \leq 100$) 表示測試資料的筆數，每組資料有一個正整數 N ($N \leq 100$) 表示方格的大小，接下來有 N 列每列有 N 個字元，描述玩家的方格資訊，字元`.`表示空格，`x`表示戰艦的一部分，`@`表示戰艦被炮擊的部位。

Output

For each test case output a single line 'Case T : N'. Where T is the test case number (starting from 1) and N is the number of still "alive" ships.

每組資料請輸出格式"Case T: N"於單行。T 表示測試資料編號(由 1 開始)，N 表示還存活在戰場上的戰艦數目。

Sample Input

```
2
4
X...
..X.
@.@.
....
2
..
X.
```

Sample Output

```
Case 1: 2
Case 2: 1
```

Analysis

基本模擬題。依照題意利用 DFS 下去搜索船隻數目即可。

11946 Code Number

分級：2

分類：字元與字串

Adrian and Maria are relatives that live in different towns. As they inhabit a rural area, it is very difficult for them to keep in touch. One way they found to overcome their communication problem was to send a line through their parents that used to visit each other. The point is that Adrian and Maria did not want that their parents read their messages, and they decided to create a secret code for the messages. The code is not very sophisticated, but you should keep in mind Adrian and Maria are just children.

In general, the meaning of a message is based on changing some letters by numbers. Each message is composed by several lines using uppercase letters of the English alphabet, space and punctuation symbols: dot and comma. The letters that are changed by numbers can be seen in the following example; this change is the same for all messages between Adrian and Maria.

Message in "Code Number ":

H3LL0 MY L0V3, 1 M H499Y **83C4U53 500N 1 WILL 83 70 YOUR 51D3.**
7H15 71M3 W17H0U7 YOU H45 **833N 373RN4L. 1 INV173 YOU 70 7H3 200**
0N3 70 533 7H3 238R42 4ND 60R1L42.

DecodeMessage:

HELLO MY LOVE, I M HAPPY BECAUSE SOON I WILL BE TO YOUR SIDE.
THIS TIME WITHOUT YOU HAS BEEN ETERNAL. I INVITE YOU TO THE ZOO
ONE TO SEE THE ZEBRAS AND GORILAS.

Hint: In this example, the relation Number-Letter between the message in "Code Number" and the "Decode Message" this indicated by the numbers and letters in bold.

Adrian 與 Maria 是住在不同鎮上的親戚，他們住在農村溝通不便，為了解決溝通不便的問題，他們會請常互相拜訪彼此的父母代為傳遞訊息。兩人不希望父母讀懂他們的訊息，所以他們決定將訊息作編碼，你要理解他們年紀還小，所以編碼的方式非常簡單。

在一般情況下，每則密文是基於數字來改變的一些字母。而他們的訊息會有許多列，且皆由大寫英文字母與空白、標點符號所組成，他們把 10 個字母分別用數

字代替，取代的十個字母可由下列資料觀察得出。

加密資料：

H3LL0 MY L0V3, 1 M H499Y 83C4U53 500N 1 WILL 83 70 YOUR 51D3.
7H15 71M3 W17H0U7 YOU H45 833N 373RN4L. 1 INV173 YOU 70 7H3 200
0N3 70 533 7H3 238R42 4ND 60R1L42.

解密資料：

HELLO MY LOVE, I M HAPPY BECAUSE SOON I WILL BE TO YOUR SIDE.
THIS TIME WITHOUT YOU HAS BEEN ETERNAL. I INVITE YOU TO THE ZOO
ONE TO SEE THE ZEBRAS AND GORILAS.

在這個例子中，Code Number 數字信與 Decode Message 不同點，已用粗體來表示。

Input

The first line of input gives the number of test cases, T ($1 \leq T \leq 30$). Each test case represents a message written in Code Number, which is composed by $1 \leq N \leq 100$ lines and each line cannot exceed 80 characters of length. Two words in the same line are separated by one white space. One blank line separated each test case.

輸入的第一列有一個表示測試資料組數的整數 T ($1 \leq T \leq 30$)。每組測試資料是加密後的文字，最少 1 列最多 100 列來組成，每列不超過 80 個字元。以一個空白字元隔開同一列的兩個字，並以一列空行隔開兩組測試資料。

Output

For each test case you must print the decoded message. You must print a blank line between each test case.

請輸出解密後的訊息，每組測試資料間請輸出一列空白。

Sample Input

```
1
H3LL0 MY L0V3, 1 M H499Y 83C4U53 500N 1 WILL 83 70 YOUR 51D3. 7H15
71M3 W17H0U7 YOU H45 833N 373RN4L. 1 INV173 YOU 70 7H3 200 0N3 70
533 7H3 238R45 4ND 60R1L45.
```

Sample Output

HELLO MY LOVE, I M HAPPY BECAUSE SOON I WILL BE TO YOUR SIDE.
THIS TIME WITHOUT YOU HAS BEEN ETERNAL. I INVITE YOU TO THE
ZOO ONE TO SEE THE ZEBRAS AND GORILAS.

Analysis

根據題意只有 10 個字母會被加密，所以直接觀察來轉化即可。

11942 Lumberjack Sequencing

分級：1

分類：排列

Another tale of lumberjacks? Let see ...

The lumberjacks are rude, bearded workers, while foremen tend to be bossy and simpleminded. The foremen like to harass the lumberjacks by making them line up in groups of ten, ordered by the length of their beards. The lumberjacks, being of different physical heights, vary their arrangements to confuse the foremen. Therefore, the foremen must actually measure the beards in centimeters to see if everyone is lined up in order.

Your task is to write a program to assist the foremen in determining whether or not the lumberjacks are lined up properly, either from shortest to longest beard or from longest to shortest.

另一個伐木工人的故事？讓我們來看看...

伐木工是粗魯、留著邋邋鬍的一群工人，其中工頭往往是專橫又頭腦簡單的。有一個工頭很喜歡找伐木工們的麻煩，下令要伐木工們以十個一組按照他們的鬍子長短依序排成一列，而伐木工們常會改變他們位置以混淆工頭，因此，工頭必須實際測量鬍鬚的長度，來看看大家是否排列整齊。

你的任務是寫一個程式，判斷伐木工是否有以由長到短，或是由短到長的順序排成一列。

Input

The input starts with line containing a single integer N , $0 < N < 20$, which is the number of groups to process. Following this are N lines, each containing ten distinct positive integers less than 100.

輸入第一列有一個整數 $N(0 < N < 20)$ 表示測試資料的組數，接下來有 N 列，每列有 10 個相異的正整數(< 100)。

Output

There is a title line, then one line per set of beard lengths. See the sample output for capitalization and punctuation.

有一標題列，後面接著一行表示鬍子長度有無照順序排列。詳細格式（大小寫、標點符號）請見範例輸出。

Sample Input

```
3
13 25 39 40 55 62 68 77 88 95
88 62 77 20 40 10 99 56 45 36
91 78 61 59 54 49 43 33 26 18
```

Sample Output

```
Lumberjacks:
Ordered
Unordered
Ordered
```

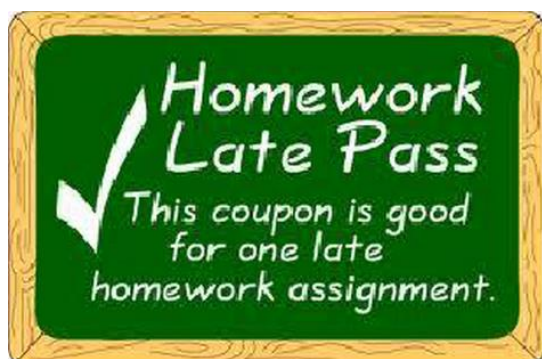
Analysis

設立 `flag` 來判斷是否為順序排列即可。

11917 Do Your Own Homework!

分級：1

分類：查詢



These days Soha is so busy that he doesn't have time to do his own homework. But this is not a big problem since he has got many friends who are willing to help. One of his friend's name is Sparrow. Whenever Soha is assigned any homework, he turns to Sparrow for her help.

Sparrow has given a list of subjects that she is comfortable with along with the number of days it will take her to complete an assignment for each subject. Soha has got only D days to complete his next assignment. However, the professor of this subject is a little flexible and allows late submissions up to 5 days. That means he will not accept any submission that is after $D + 5$ days from now. Will Sparrow be able to do it for Soha this time?

最近 Soha 實在是太忙，忙到沒時間寫作業，不過這不是個大問題，還好他有很多朋友來幫他，Sparrow 就是其中一位肯幫忙他寫。

Sparrow 會列出她幫得上忙的科目，及每個科目作業預計完成的天數。Soha 只有 D 日可以完成他的作業，但還好他的教授允許遲交 5 天，也就是說，作業最晚可以在 $D+5$ 天內交出去即可。不過這時間 Sparrow 都可以幫忙到嗎？

Input

First line of input is a positive integer T ($T \leq 100$) that determines the number of test cases. Each case starts with a line containing an integer N that represents the number of subjects Sparrow is comfortable with. Each of the next N lines contain the name of a subject followed by the number of days it will take Sparrow to complete an assignment of that subject. All these subject names will be distinct. The next line contains an integer D . The meaning of D is described above. The following line contains the name of the subject whose homework is due. All the subjects' names consist of lowercase letters and the length of each is at least 1 and at most 20. All the integer inputs are positive in the range $[1,100]$.

輸入的第一列有一個正整數 T 表示測試資料的組數 ($T < 100$)，每組測試資料的第

一列為整數 N ，表示接下來有 N 個 Sparrow 願意幫忙的科目，每筆科目資料包含科目名稱與完成所需的天數，所有科目不會出現相同的名稱。接下來會有一個整數 D 表示 Soha 只有 D 日可以完成他的作業，再接下來的一列表示他需要繳交的作業科目名稱。所有科目名稱皆為小寫字串且長度為 1~20 個字元，所有整數皆介於 [1,100] 的範圍。

Output

For each case, first output the case number first starting from 1. If Sparrow doesn't take more than D days to completely the assignment, output `Yesss`; if she takes more than D days but not more than $D + 5$, output `Late`; if she takes more than $D + 5$ days or if she isn't comfortable with the subject, output `Do your own homework!`. Quotes are for clarify only and don't need to be part of the output. Look at the samples for more details. Be careful about the spelling.

對於每筆資料，第一行請輸出其編號（從 1 開始）。接下來如果 Sparrow 完成時間不超過 D 日，輸出 "Yesss"；時間超過 D 日但不超過 $D+5$ 日，輸出 "Late"，但若完成時間超過 $D+5$ 日或她不想幫忙做此作業，請輸出 "Do your own homework!"，如下列測試資料所示。輸出資料須明確不用有其他部分，仔細參考範例，請更小心地來輸出。

Sample Input

```
3
3
compiler 4
cplusplus 1
java 8
5
compiler
2
algorithm 3
math 9
4
math
2
java 8
ai 3
6
calculus
```

Sample Output

```
Case 1: Yesss
```

Case 2: Late

Case 3: Do your own homework!

Analysis

簡單題。以字串相同的比較天數即可。

11898 Killer Problem

分級：2

分類：排序

You are given an array of N integers and Q queries. Each query is a closed interval $[L, R]$. You should find the minimum absolute difference between all pairs in that interval.

給你一陣列含 N 個整數和 Q 個查詢，每個查詢皆為一封閉區間 $[L, R]$ 。你須在該區間的每個值中，找出一最小絕對差值。

Input

First line contains an integer T ($T \leq 10$). T sets follow. Each set begins with an integer N ($N \leq 200000$). In the next line there are N integers A_i ($1 \leq A_i \leq 10^4$), the number in the i -th cell of the array. Next line will contain Q ($Q \leq 10^4$). Q lines follow, each containing two integers L_i, R_i ($1 \leq L_i, R_i \leq N, L_i < R_i$) describing the beginning and ending of i -th range. Total number of queries will be less than 15000.

輸入第一行包含一整數 T ($T \leq 10$) 指測試資料的組數，測資開始設有一整數 N ($N \leq 200000$) 表示下一行中，有 N 個整數在陣列 A_i ($1 \leq A_i \leq 10^4$) 裡。再來有一整數 Q ($Q \leq 10^4$) 表示接下來的 Q 行各有兩整數 L_i 和 R_i ($1 \leq L_i, R_i \leq N, L_i < R_i$)，說明 i 值的範圍。全部查詢的值不超過 15000。

Output

For the i -th query of each test output the minimum $|A_j - A_k|$ for $L_i \leq j, k \leq R_i$ ($j \neq k$) a single Line.

對於每個範圍的查詢，須輸出 $|A_j - A_k|$ 的最小值 ($L_i \leq j, k \leq R_i; j \neq k$) 於單行。

Sample Input

```
1
10
1 2 4 7 11 10 8 5 1 10000
4
1 10
1 2
3 5
8 10
```

Sample Output

0
1
3
4

Analysis

數字的最大範圍為 10^4 ，所以直接以暴力法，將區間所有數字拿出來排序找差值即可。

11879 Multiple of 17

分級：2

分類：大數運算

Theorem: If you drop the last digit d of an integer n ($n \geq 10$), subtract $5d$ from the remaining integer, then the difference is a multiple of 17 if and only if n is a multiple of 17.

For example, 34 is a multiple of 17, because $3-20=-17$ is a multiple of 17; 201 is not a multiple of 17, because $20-5=15$ is not a multiple of 17.

Given a positive integer n , your task is to determine whether it is a multiple of 17.

定理：若你移除一個整數 n ($n \geq 10$) 的最後一個位數 d ，其值再減去 $5d$ 之後，又為 17 的倍數，則若且為若 n 亦為 17 的倍數。

例如：34 為 17 的倍數，因為 $3-20 = -17$ 為 17 的倍數；201 非 17 的倍數，因為 $20-5=15$ 非 17 的倍數。

給定一正整數 n ，請你判斷 n 是否為 17 的倍數。

Input

There will be at most 10 test cases, each containing a single line with an integer n ($1 \leq n \leq 10^{100}$). The input terminates with $n = 0$, which should not be processed.

輸入最多有十組測試資料，每組皆有一列含一整數 n ($1 \leq n \leq 10^{100}$)，當 $n = 0$ 表示資料結束。

Output

For each case, print 1 if the corresponding integer is a multiple of 17, print 0 otherwise.

對於每筆測資，若為 17 的倍數請輸出 1，否則請輸出 0。

Sample Input

```
34
201
2098765413
```


11850 Alaska

分級：2

分類：排序

The Alaska Highway runs 1422 miles from Dawson Creek, British Columbia to Delta Junction, Alaska. Brenda would like to be the first person to drive her new electric car the length of the highway. Her car can travel up to 200 miles once charged at a special charging station. There is a charging station in Dawson Creek, where she begins her journey, and also several charging stations along the way. Can Brenda drive her car from Dawson City to Delta Junction and back?

阿拉斯加高速公路由 Dawson Creek 到 Delta Junction 全長共 1422 英哩，Brenda 想成為首位開電動車行駛整條阿拉斯加高速公路的人，她的電動車再專門的充電站充一次電可以跑 200 英哩，而充電站除了在她的起點 Dawson Creek 有之外，沿著公路能充電的地方只有幾家。究竟 Brenda 是否能開著她的電動車成功往返阿拉斯加高速公路呢？

Input

The input contains several scenarios. Each scenario begins with a line containing n , a positive number indicating the number of charging stations. n lines follow, each giving the location of a filling station on the highway, including the one in Dawson City. The location is an integer between 0 and 1422, inclusive, indicating the distance in miles from Dawson Creek. No two filling stations are at the same location. A line containing '0' follows the last scenario.

輸入有多組測試資料，每組資料的第一列有一個正整數 n ，表示充電站的總數，接下來的 n 列表示充電站所在公路的位置，且一定包含起點 Dawson Creek 的充電站，數值表示距離起點的距離，大小介於 0~1422，不會有兩個充電站的位置一樣。當 $n = 0$ 表示測試資料結束。

Output

For each scenario, output a line containing 'POSSIBLE' if Brenda can make the trip. Otherwise, output a line containing the word 'IMPOSSIBLE'.

每筆測試資料輸出一列。假如 Brenda 能成功完成她的旅程請輸出"POSSIBLE"，否則請輸出"IMPOSSIBLE"。

Sample Input

2
0
900
8
1400
1200
1000
800
600
400
200
0
0

Sample Output

IMPOSSIBLE

POSSIBLE

Analysis

排序。直接排序然後計算差值即可。

11849 CD

分級：2

分類：排序

Jack and Jill have decided to sell some of their Compact Discs, while they still have some value. They have decided to sell one of each of the CD titles that they both own. How many CDs can Jack and Jill sell?

Neither Jack nor Jill owns more than one copy of each CD.

傑克與吉兒兩人想要賣出他們的 CD 光碟，他們決定賣出他們兩人皆擁有的 CD 光碟其中的一張，本題要請問他們兩人共想賣多少光碟？

傑克與吉兒擁有不只一份的相同 CD。

Input

The input consists of a sequence of test cases. The first line of each test case contains two non-negative integers N and M , each at most one million, specifying the number of CDs owned by Jack and by Jill, respectively. This line is followed by N lines listing the catalog numbers of the CDs owned by Jack in increasing order, and M more lines listing the catalog numbers of the CDs owned by Jill in increasing order. Each catalog number is a positive integer no greater than one billion. The input is terminated by a line containing two zeros. This last line is not a test case and should not be processed.

輸入有多組測試資料，每組的第一列有兩個非負整數 N 與 M ，其最大值為 1,000,000，各表示他們兩個人擁有的 CD 光碟總數，接下來會有 N 列，每列為一個整數，這 N 個整數為一組遞增序列，表示傑克所擁有的 CD 光碟名稱(編號)。再接下來會有 M 列，每列為一個整數，這 M 個整數亦為遞增序列，表示吉兒擁有的 CD 光碟。表示編號的整數值不會大於 1,000,000,000。當 N, M 皆為零表示測試資料結束。

Output

For each test case, output a line containing one integer, the number of CDs that Jack and Jill both own.

每組測試資料請輸出一列，表示兩人想賣 CD 光碟的總數量。

Sample Input

3 3

1

2

3

1

2

4

0 0

Sample Output

2

Analysis

簡單題。利用合併排序合併所有編號。再設置一個指標，如果這一項與下一項值相同就加一即可求答。

11847 Cut the Silver Bar

分級：2

分類：數學計算

A creditor wants a daily payment during n days from a poor miner in debt. Since the miner can't pay his daily obligation, he has negotiated with the creditor an alternative way, convenient for both parties, to pay his debt: the miner will give an equivalent of a 1μ ($= 0.001$ mm) long piece of a silver bar as a guarantee towards the debt. The silver bar owned by the poor miner is initially $n\mu$ units long.

By the end of n days the miner would not have any more silver to give and the creditor would have received an amount of silver equivalent to that of the silver bar initially owned by the miner. By then, the miner expected to have enough money to pay the debt at the next day so that he would have back all his silver.

With this negotiation in mind, the miner has realized that it was not necessary to cut exactly 1μ silver piece from the bar every day. For instance, at the third day he could give the creditor a 3μ silver piece, taking back the equivalent of a 2μ silver piece which the creditor should already have.

Since cutting the bar was rather laborious and time consuming, the miner wanted to minimize the number of cuts he needed to perform on his silver bar in order to make the daily silver deposits during the n days. Could you help him?

一位債主想要一個貧窮的礦工在 n 天內以一天為單位償還一些債務，但是窮礦工沒有能力負擔這個每日的債務，因此礦工與債主協商出一個對雙方都方便的作法：礦工每天會付出 1 單位(0.001mm)長的銀條作為償還債務的抵押。礦工一開始會擁有 n 個單位長的銀條。

到了第 n 天時礦工將會付出他所有的銀條給債主，這時礦工預期他已經有足夠的錢可以把他的所有銀條給贖回來。

根據這個協商條件，礦工發現他不需要將他的銀條每天都切出 1 個單位來。舉例來說，到了第三天他可以交給債主一個 3 單位的銀條，然後拿回之前付給債主的 2 單位銀條。

由於切割這些銀條既費時又費力，因此礦工希望能將切割的次數最小化且能滿足上述的條件，你能幫助他嗎？

Input

Input consists of several cases, each one defined by a line containing a positive integer number n (representing the length in micros of the silver bar and the number of days of the amortization period). You may assume that $0 < n < 20000$.

The end of the input is recognized by a line with 0.

輸入包含多組測試資料，每組測試資料包含一個正整數 n ，代表礦工一開始的銀條長度以及要攤銷債務的天數。你可以假設 $0 < n < 20000$ 。

輸入為 0 時代表輸入結束。

Output

For each given case, output one line with a single number: the minimum number of cuts in which to cut a silver bar of length n to guarantee the debt during n days.

對每一組測試資料輸出一個數值，代表在 n 天內礦工最少要對長度 n 單位的銀條切割多少次。

Sample Input

```
1
5
3
0
```

Sample Output

```
0
2
1
```

Analysis

分析題意，了解題目要求的就是：最少需要用幾個 bit 來表示小於 n 的所有正整數。也就是，對 n 取以 2 為底的對數，就是銀條需要切割的次數了。

11830 Contract Revision

分級：1

分類：字元與字串

For years, all contracts of the Association of Contracts for Modernization (ACM) were typed using an old typewriter machine.

Recently Mr. Miranda, one of the accountants of the ACM, realized that the machine had a failure in one, and only one, numerical digit. More specifically, the awed digit, when typed, is not printed on the sheet, as if the corresponding key was not pressed. He realized that this could have changed the numerical representation of contract values. Worried about accounting, Mr. Miranda wants to know, from the original values agreed for the contracts (which he kept in handwritten notes) which values are actually represented in the contracts. For example, if the failed digit in the machine is 5, an agreed value of 1500 would be represented in the corresponding contract as 100, because the digit 5 would not be printed. Note that Mr. Miranda wants to know the numeric value represented in the contract, ie, in the same machine, the number 5000 corresponds to the numeric value 0, not 000 (as it actually appears in the contract).

好幾年以來，所有 ACM 的合約都是由老式打字機打出來的。

最近，有一位 ACM 的會計師發現打字機有一個按鍵，而且是數值按鍵打不出字，有打就好像沒打一樣，他知道這樣會造成合約的數值表示出現問題。基於會計上的需要，他想要知道原始合約的數值資料(他有手寫的原稿)經由該台打字機所打出來的值是多少。例如如果壞掉的數值按鍵為 5，而原始合約內容寫的是 1500，則最後會打出 100 的數值，因為 5 印不出來。會計師想知道的是印錯的數值"大小"是多少，而非印出什麼值，例如 5000，壞的打字機印出來的雖然是 000，但是你要告訴會計師的數值是 0，而非 000。

Input

The input consists of several test cases, each in one line. Each line contains two integers D and N ($1 \leq D \leq 9$ and $1 \leq N < 10^{100}$), representing, respectively, the digit that has failed in the machine and the number that was originally agreed for the contract (which can be very large because of hyperinflation).

The last test case is followed by a line which contains only two zeros separated by white space.

輸入會有許多測試資料，每一列有兩個整數 D 與 $N(1 \leq D \leq 9, 1 \leq N \leq 10^{100})$ ，分別表示壞掉的數值按鍵與合約上的原始數值(由於高通膨的關係，該數值可能會是天文數字)。

以空白字元隔開的兩個分別為 0 的整數，表示測試資料結束。

Output

For each test case in the input your program must print one line containing a single integer, the numeric value represented in the contract.

對於每筆測資，請輸出一行表示壞掉的打字機所印出來的數值大小是多少？

Sample Input

```
5 5000000
3 123456
9 23454324543423
9 999999999991999999
7 777
0 0
```

Sample Output

```
0
12456
23454324543423
1
0
```

Analysis

簡單題。將與壞掉數值一樣的數字利用取代方式去除即可。

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
```

Analysis

簡單題。利用輾轉相除法求 GCD，再來找出最大值即可。

11824 A Minimum Land Price

分級：1

分類：排列

Manager of ACM-ICPC Thailand Contest Council is planning to buy lands in Phuket to build the office building for national programming skill camp and programming contest that will be held on Phuket regularly in the future. The land price in Phuket is becoming more expensive in every year. The price increases in the exponential growth curves by a factor of year. If the land i whose initial cost is L_i bought in t years from now, its price will be $2 \times (L_i)^t$. All land prices are different. ACM-ICPC can buy only one land per year. You have to help the manager to buy the lands at lowest price within the budget of 5,000,000 million baht.

For example, if we want to buy 3 lands with costs 7, 2 and 10 in 3 consecutive years, the total price will be calculated as follow:

$$(2 \times 7) + (2 \times 2^2) + (2 \times 10^3) = 2022 \text{ million baht}$$

ACM-ICPC 主辦單位想在泰國普吉島買地蓋大樓，不過普吉島的地價連年不斷攀升，每年以指數遞增。假如第 i 塊地一開始的地價為 L_i ，則 t 年後會漲到 $2 \times (L_i)^t$ ，每塊地的地價不會都一樣。ACM-ICPC 每年只能買一塊地，你必須幫助主辦單位以最便宜的價格買到所有土地，預算上限為 5,000,000 百萬泰銖。

例如，我們想買 3 塊地，一開始的價格分別是 7, 2, 10 百萬泰銖，則買價可能會是：

$$(2 \times 7) + (2 \times 2^2) + (2 \times 10^3) = 2022 \text{ 百萬泰銖}$$

Input

First line of the input contains an integer T ($1 \leq T \leq 10$), the number of test cases. Each test case contains integer L_i which is the cost of land in million baht. There are less than 40 lands in each test case. The line contains '0' (zero) indicates the end of each test case.

輸入的第一列有一個表示測試資料組數的整數 T ($1 \leq T \leq 10$)，接下來會有多個整數 L_i 表示土地的價格(單位:百萬泰銖)，每組測試資料最多有 40 筆土地價格，並以 0 表示每組測試資料的結束。

Output

For each test case, print out the minimum price for purchasing all lands. If the total

price exceeds the budget (5,000,000 millions baht), print out `Too expensive`.

對於每組測資，輸出購買全部土地的最低價格，若總價超過預算 5,000,000 百萬泰銖，請輸出"Too expensive"。

Sample Input

```
3
7
2
10
0
20
29
31
0
42
41
40
37
20
0
```

Sample Output

```
134
17744
Too expensive
```

Analysis

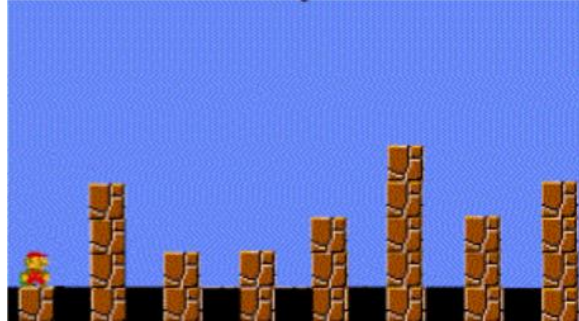
簡單題。先排序再直接按照題意代入公式即可。

11764 Jumping Mario

分級：1

分類：排列

Mario is in the final castle. He now needs to jump over few walls and then enter the Koopa's Chamber where he has to defeat the monster in order to save the princess. For this problem, we are only concerned with the "jumping over the wall" part. You will be given the heights of N walls from left to right. Mario is currently standing on the first wall. He has to jump to the adjacent walls one after another until he reaches the last one. That means, he will make $(N - 1)$ jumps. A high jump is one where Mario has to jump to a taller wall, and similarly, a low jump is one where Mario has to jump to a shorter wall. Can you find out the total number of high jumps and low jumps Mario has to make?



瑪莉歐(Mario)在最後的城堡。他現在需要跳過一些牆壁，然後進入庫巴(Koopa)的房間，因為他要打敗怪物，以拯救公主。而對於這個問題，我們先關注“翻過牆”的一部分。給予你 N 個牆壁(由左至右)的高度。瑪莉歐目前站在第一個牆壁。他必須跳到相鄰的牆壁直到最後一個。這意味著，他將跳躍 $N - 1$ 次。”a high jump”代表瑪莉歐跳到一個較高的牆。相反，”a low jump”代表瑪莉歐跳到一個較矮的牆。你能找出 a high jump 和 a low jump 的總數嗎？

Input

The first line of input is an integer T ($T < 30$) that indicates the number of test cases. Each case starts with an integer N ($0 < N < 50$) that determines the number of walls. The next line gives the height of the N walls from left to right. Each height is a positive integer not exceeding 10.

第一行輸入的是一個整數 T ($T < 30$)，表示接下來有 T 筆測資。每筆測資開始於一個正整數 N ($N < 50$)，表示牆壁的數目。下一行依序為 N 個牆壁的高度(由左至右)。每一個高度是不超過 10 的正整數。

Output

For each case, output the case number followed by 2 integers, total high jumps and total low jumps, respectively. Look at the sample for exact format.

對每筆測資，先輸出這是第幾筆測資，再接著兩整數代表 `high jumps` 和 `low jumps` 的總數。格式請參考範例測資。

Sample Input

```
3
8
1 4 2 2 3 5 3 4
1
9
5
1 2 3 4 5
```

Sample Output

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

Analysis

按照題意，紀錄上一次的高度以及這一次的高度，再比較下一步是往上或是往下跳，將次數累加輸出即可。

11742 Social Constraints

分級：2

分類：排列

Socializing can be a very complicated thing among teenagers. For example, finding a good seating arrangement in a movie theater can be a difficult task. Here is a list of constraints that could potentially apply to two individuals A and B in this situation:

- if A and B are dating, then they must sit beside each other
- if A and B are fighting, then they cannot sit beside each other
- if A and B have just broke up, then they must sit at opposite ends of the row

Teenage politics is a complicated thing meaning the constraints can get even more complicated than those listed above. However, we restrict this problem to a particular form of constraint that simply specifies a lower or upper bound on the number of seats separating two specific individuals.

The group arrives after everyone else watching the show has been seated. By some stroke of luck, there are exactly as many open seats as there are teenagers and all of these seats appear consecutively in the front row. How many possible seating arrangements satisfy the constraints?

青少年間的人際關係非常複雜，例如看電影買票的座位安排是可以成為困難的任務，這裡將可能會發生在 A 與 B 間的情況列成一張清單：

- 如果 A 與 B 在交往，那一定越靠近越好。
- 如果 A 與 B 正在吵架，那互相坐越遠越好。
- 如果 A 與 B 剛分手，那一定是分開坐兩端的座位。

青少年的世界一定比上面所列的更加複雜，所以我們簡化這個問題，只考慮誰跟誰一定要距離幾個座位，或是一定要少於幾個座位。

節目開始其他群眾紛紛入座，連續與前排位置可能會被占走，而若運氣好些有剩下位置，那麼請問在這一排座位中，讓每個人都有座位可以做，符合此條件的情況會有幾種？

Input

Each test case begins with two integers n and m with $0 < n \leq 8$ and $0 \leq m \leq 20$ where n is the size of the group. For simplicity, assume the teenagers are numbered from 0 to

$n - 1$. Then of m lines follow, each describing a constraint, where a line consists of three integers a, b, c satisfying $0 \leq a < b < n$ and $0 < |c| < n$. If c is positive then teenagers a and b must sit at most c seats apart. If c is negative, then a and b must sit at least $-c$ seats apart. The end of input is signaled by a line consisting of $n = m = 0$.

每組測資開始有 N ($0 < N \leq 8$) 和 M ($0 \leq M \leq 20$) 兩整數，其中 N 代表有 N 個青少年，而為了簡單起見，編號給定分別是 $0 \sim n-1$ 。接下來有 M 行，每一行中有 a, b, c ($0 \leq a < b < n$ and $0 < |c| < n$)。如果 C 是正整數， A 跟 B 必須坐相離不可超過 C ；若如果 C 是負數，則 A 跟 B 必須相離 $-C$ 以上。輸入測資以 $n = m = 0$ 為結束。

Output

The output for each test case is a single line containing the number of possible seating arrangements for the group that satisfy all of the social constraints.

對於每個測資輸出一行，表示他們符合所有條件的有幾種可能性。

Sample Input

```
3 1
0 1 -2
3 0
0 0
```

Sample Output

```
2
6
```

Analysis

本題使用暴力法，按照題意， N 個人共有 $N!$ 的排列方式，又每種方式最多會檢查 M 次，依此來算出符合的組合數量。

11730 Number Transformation

分級：2

分類：排列

You are given an integer number S . You can transform any integer number A to another integer number B by adding x to A . This x is an integer number which is a prime factor of A (Please note that 1 and A are not being considered as a factor of A). Now, your task is to find the minimum number of transformations required to transform S to another integer number T .

你可以將整數 A 加上一個數 x 後轉換成 B 。而 x 是 A 的質因數（請注意，1 和 A 不被算在 A 的因數中）。現在給你兩整數 S 和 T ，你的任務是算出最少需要轉換幾次才能把 S 轉換成 T 。

Input

For each test case, there will be a line with two integers, S ($1 \leq S \leq 100$) & T ($1 \leq T \leq 1000$), as described above. The last test case will be followed by a line with two 0's denoting end of input. This case should not be processed.

每組測試資料為一列，如上面所說，會給你 2 個整數 S ($1 \leq S \leq 100$) 和 T ($1 \leq T \leq 1000$)。而當 $S=0$ $T=0$ 時代表輸入結束，且不必執行。

Output

For every test case except the last one, print a line of the form `Case X : Y' where X is the serial number of output (starting from 1). Y is the minimum number of transformations required to transform S to T . If it is not possible to make T from S with the given rules, Y shall be -1 .

Explanation of case 1:

You can make 12 from 6 in 2 steps in this way: $6 \rightarrow 9 \rightarrow 12$.

對每組測試資料（不包含最後一組）輸出於一列，規定格式為`Case X : Y'。X 表示是第幾組的測資（從 1 開始），Y 代表著 S 轉換到 T 的最小次數。若是無法轉換成功，請輸出 -1

事件一說明：

若 $S = 6$, $T = 12$ ，則由 S 轉成 T 的最少步驟為： $6 \rightarrow 9 \rightarrow 12$ ，共 2 次轉換。

Sample Input

6 12

6 13

0 0

Sample Output

Case 1: 2

Case 2: -1

Analysis

每做一次轉換都要重算一次 A 的質因數，所以為了節省時間先建質數表是必須的。再利用 BFS 的方式，從 S 展開搜索。小心可能 TLE。

11727 Cost Cutting

分級：1

分類：排列

Company XYZ have been badly hit by recession and is taking a lot of cost cutting measures. Some of these measures include giving up office space, going open source, reducing incentives, cutting on luxuries and issuing pink slips.

They have got three (3) employees working in the accounts department and are going to lay-off two (2) of them. After a series of meetings, they have decided to dislodge the person who gets the most salary and the one who gets the least. This is usually the general trend during crisis like this.

You will be given the salaries of these 3 employees working in the accounts department. You have to find out the salary of the person who survives.

XYZ 公司由於面臨經濟不景氣正規劃降低營運成本，其中措施包括減少辦公空間，並開源節流，所以他們決定要裁員！

他們決定裁掉會計部三位員工中的其中兩位，並在一系列的會議中，決定把最高薪與最低薪的那兩位裁掉，而這真的是大勢所驅之。

給定會計部三位員工的薪資，你必須找出誰是唯一留下來的人。

Input

The first line of input is an integer T ($T < 20$) that indicates the number of test cases. Each case consists of a line with 3 distinct positive integers. These 3 integers represent the salaries of the three employees. All these integers will be in the range $[1000, 10000]$.

第一列有一個整數 T ($T < 20$) 表示測試資料的組數。每組資料有三個不同的正整數，分別表示三位員工的薪資，所有資都在 $[1000, 10000]$ 的範圍。

Output

For each case, output the case number followed by the salary of the person who survives.

請輸出每組資料該位留下來的員工的薪資。

Sample Input

3

1000 2000 3000

3000 2500 1500

1500 1200 1800

Sample Output

Case 1: 2000

Case 2: 2500

Case 3: 1500

Analysis

排序找中間值即是答案。

11716 Digital Fortress

分級：1

分類：字元與字串

In the last IIUPC there was a problem called Da Vinci Code prepared on the story of the bestselling book of Dan Brown, The Da Vinci Code. Here is another problem based on his another techno-thriller novel Digital Fortress. In this problem, you will be given a cipher text. Your task is to decipher the text using the decrypting technique described below. Let's take an example. A cipher text is given as follows:

WECGEWHYAAIORTNU

The output will be:

WEAREWATCHINGYOU

For this problem, there are 16 characters in the given cipher text "WECGEWHYAAIORTNU" which is square of 4. These letters have to be arranged in $n \times n$ (in this example 4×4) grid and each letter from the given input will be placed in a grid in row major order (1st row, 2nd row, 3rd row, ...). When the given cipher text is placed in the grid it looks like as follow:

```
WECG
EWHY
AAIO
RTNU
```

From the above grid if we take the letters in column major order (1st column, 2nd column, 3rd column, ...) then we get the following decrypted text:

WEAREWATCHINGYOU

上一屆的 IIUPC 有一題「達文西密碼」是以丹布朗的暢銷書「達文西密碼」為故事背景。這一題則是以他另一部科技驚悚小說「數位堡壘」為背景。題目給你一個密文，你的任務便是以下述的解碼方式來解碼。比如說，如果密文如下：

WECGEWHYAAIORTNU

輸出則是：

WEAREWATCHINGYOU

上例中所給的密文 "WECGEWHYAAIORTNU" 有 16 個字元，也就是 4 的平方，而這些字元將會以「列優先」(第一列放滿後再放第二列，再第三列……) 的方式重新置入一個 $n \times n$ (本例為 4×4) 的格子。本例密文放置完成後的格子如下：

WECG

EWHY

AAIO

RTNU

再來，當我們以「行優先」(取完第一行再取第二行，然後第三行)的方式將上面格子的字元取出便可以得到明文：

WEAREWATCHINGYOU

Input

Input starts with a line consisting of a single number T . T test cases follow. Each test case consists of one line. This line contains the cipher text. The cipher text contains either UPPERCASE letters or blank spaces. Total number of character in the text will not be more 10,000.

輸入的第一行有一個數字 T ，表示接著有 T 筆測資。每筆測資一行，內容便是要處理的密文。密文僅包含大寫字母及空白，而其總字元數不會超過 10,000。

Output

For each test case, the output contains a single line containing the decrypted text. If the number of characters in the input text is not square of any number, then give the output 'INVALID'.

對於每筆測資，請將明文輸出於一行。若輸入的字元數不是完全平方數，請輸出「INVALID」。

Sample Input

```
3
WECGEWHYAAIORTNU
DAVINCICODE
DTFRIAEOGLRSITS
```

Sample Output

```
WEAREWATCHINGYOU
INVALID
DIGITALFORTRESS
```

Analysis

每組測資須先判斷它的長度是否合乎 $n \times n$ 的陣列，只須以長度開根號後的長度為間隔，依序印出即可。

11689 Soda Surpler

分級：1

分類：數學計算

Tim is an absolutely obsessive soda drinker, he simply cannot get enough. Most annoyingly though, he almost never has any money, so his only obvious legal way to obtain more soda is to take the money he gets when he recycles empty soda bottles to buy new ones. In addition to the empty bottles resulting from his own consumption he sometimes find empty bottles in the street. One day he was extra thirsty, so he actually drank sodas until he couldn't afford a new one.

Tim 是一個忠實的蘇打迷，也無法有喝膩的一天。可是最煩人的事是他沒有任何錢，所以他買汽水的唯一方式就是收集空瓶子來回收，除了自己喝的，他也會去街上找找有沒有空瓶子。而在某一天他格外的渴，以至於想喝完所有的蘇打，直到剩下瓶子無法換新的一瓶。

Input

The first line of the input file contains an integer N ($N < 15$) which denotes the total number of test cases. The description of each test case is given below:

Three non-negative integers e, f, c , where $e < 1000$ equals the number of empty soda bottles in Tim's possession at the start of the day, $f < 1000$ the number of empty soda bottles found during the day, and $1 < c < 2000$ the number of empty bottles required to buy a new soda.

輸入的第 1 列有一個整數 N ($N < 15$)，代表以下有多少組測試資料。詳細敘述如下：

每組測試資料含有 3 個正整數 e, f, c 。其中 e ($e < 1000$) 代表 Tim 一開始擁有的空瓶子數目； f ($f < 1000$) 代表 Tim 在這一天他在街上收集到的空瓶子數目， c ($1 < c < 2000$) 代表多少個空瓶子可以換一瓶新的汽水。

Output

For each test case print how many sodas did Tim drink on his extra thirsty day? Look at the sample output for details.

對每一組測試資料，輸出一列代表在這天 Tim 可以喝掉多少瓶汽水。細節請參考 Sample Output。

Sample Input

2
903
552

Sample Output

4
9

Analysis

每組測資先將 e 、 c 值相加得知 Tim 可以獲得多少空瓶。再利用迴圈來處理加總即可。

11677 Alarm Clock

分級：1

分類：數學計算

Daniela is a nurse in a large hospital, which causes her working shifts to constantly change. To make it worse, she has deep sleep, and a difficult time to wake up using alarm clocks.

Recently she got a digital clock as a gift, with several different options of alarm sounds, and she has hope that it might help solve her problem. But, lately, she's been very tired and wanted to enjoy every single moment of rest. So she carries her new clock to every place she goes, and whenever she has some spare time, she tries to sleep, setting her alarm clock to the time when she needs to wake up. But, with so much anxiety to sleep, she ends up with some difficulty to fall asleep and enjoy some rest.

A problem that has been tormenting her is to know how many minutes of sleep she would have if she fell asleep immediately and woken up when the alarm clock ringed. But she is not very good with numbers, and asked you for help to write a program that, given the current time and the alarm time, find out the number of minutes she could sleep.

Daniela 在一家大醫院當護士，導致工作時間常變來變去。更糟的是她睡得很沉，鬧鐘很難叫得醒她。

最近她收到了一個有多種鬧鈴聲的數位時鐘，希望它可以解決她的問題。由於近來較為疲累，她希望善用休息時間。她隨身帶著這個鬧鐘，只要有休息時間，她就設好該醒來的時間並試著入睡。不過當她越焦急地想睡著，她越是睡不著。

有個問題一直苦惱著她，就是想知道如果她可以立刻睡著，在鬧鐘響以前她可以有幾分鐘的睡眠。但是她的算術不好，所以請你寫一個程式，根據現在的時間及鬧鈴的時間算出她可以睡幾分鐘。

Input

The input contains several test cases. Each test case is described in one line, containing four integers $H1$, $M1$, $H2$ and $M2$, with $H1 : M1$ representing the current hour and minute, and $H2 : M2$ representing the time (hour and minute) when the alarm clock is set to ring ($0 \leq H1 \leq 23$, $0 \leq M1 \leq 59$, $0 \leq H2 \leq 23$, $0 \leq M2 \leq 59$).

The end of the input is indicated by a line containing only four zeros, separated by blank spaces.

輸入含有多筆測資，每筆測資一行，含有四個整數 H1、M1、H2 及 M2，H1:M1 代表現在的時與分，H2:M2 則代表鬧鈴所設的時間 (時與分)， $(0 \leq H1 \leq 23, 0 \leq M1 \leq 59, 0 \leq H2 \leq 23, 0 \leq M2 \leq 59)$ 。

最後一行含有四個以空格分開的 0，代表輸入的結束。

Output

For each test case, your program must print one line, containing a single integer, indicating the number of minutes Daniela has to sleep.

對於每筆測資，你的程式要印出一整數表示 Daniela 可以睡的分鐘數，須單獨輸出於一行。

Sample Input

```
1 5 3 5
23 59 0 34
21 33 21 10
0 0 0 0
```

Sample Output

```
120
35
1417
```

Analysis

按照題意撰寫程式，只須小心隔天的時間即可。

11636 Hello World!

分級：1

分類：數學計算

When you first made the computer to print the sentence "Hello World!", you felt so happy, not knowing how complex and interesting the world of programming and algorithm will turn out to be. Then you did not know anything about loops, so to print 7 lines of "Hello World!", you just had to copy and paste some lines. If you were intelligent enough, you could make a code that prints "Hello World!" 7 times, using just 3 paste commands. Note that we are not interested about the number of copy commands required. A simple program that prints "Hello World!" is shown in Figure 1. By copying the single print statement and pasting it we get a program that prints two "Hello World!" lines. Then copying these two print statements and pasting them, we get a program that prints four "Hello World!" lines. Then copying three of these four statements and pasting them we can get a program that prints seven "Hello World!" lines (Figure 4). So three pastes commands are needed in total and Of course you are not allowed to delete any line after pasting. Given the number of "Hello World!" lines you need to print, you will have to find out the minimum number of pastes required to make that program from the origin program shown in Figure 1.

<pre>#include<stdio.h> int main(void) { printf("Hello World!\n"); }</pre>	<pre>#include<stdio.h> int main(void) { printf("Hello World!\n"); printf("Hello World!\n"); }</pre>	<pre>#include<stdio.h> int main(void) { printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); }</pre>	<pre>#include<stdio.h> int main(void) { printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); printf("Hello World!\n"); }</pre>
Figure 1	Figure 2	Figure3	Figure 4

當你剛學程式印出「Hello World!」時，你是很高興的，因為你還不曉得以後還會碰到多難多複雜的程式和演算法。假設你還不知道迴圈，如果要印出 7 行的「Hello World!」，你只能複製、然後貼上。但你夠聰明的話，你就會知道輸出 7 行的「Hello World!」只要複製貼上 3 次就夠了。如圖 1 所示，當我們複製 1 行「Hello World!」然後貼上你可以得到 2 行「Hello World!」，當你複製這 2 行然後再次貼上可以得到 4 行，最後再複製 3 行貼上就可以得到上述輸出 7 行的答案（圖 4）。所以複製貼上 3 次就是我們想要的結果。給你一個數字表示須要有多少行的「Hello World!」，請你找出最少要對原始的「Hello World!」複製貼上幾次才能達到需求的數量。

Input

The input file can contain up to 2000 lines of inputs. Each line contains an integer N ($0 < N < 10001$) that denotes the number of "Hello World!" lines are required to be printed. Input is terminated by a line containing a negative integer.

每組測資為一行，最多可能有 2000 筆，每行包括一個正整數 N ($0 < N < 10001$) 代表要印出的「Hello World!」數量。當 N 為負數時代表輸入結束。

Output

For each line of input except the last one, produce one line of output of the form 'Case X : Y' where X is the serial of output and Y denotes the minimum number of paste commands required to make a program that prints N lines of "Hello World!".

對於每行測資請輸出一行，格式為「Case X: Y」，其中 X 代表第幾筆測資，Y 代表「Hello World!」最少要複製貼上的次數。

Sample Input

```
2
10
-1
```

Sample Output

```
Case 1: 1
Case 2: 4
```

Analysis

簡單題，找出 $\lg N$ 的上界數即可。

11608 No Problem

分級：1

分類：數學計算

Programming contests are being arranged so frequently these days. While this might be a good news for the contestants, the scenario is completely opposite for the problemsetters. So far, the problemsetters somehow managed to produce some sorts of a set & say "No problem!". But it is doubtful how long will it be possible if the trend of arranging contests in a short notice continues.

You are given the number of problems created in every month of a year and number of problems required in each month. If N problems are required in a month & there are not enough problems at that time, all contests of that month is canceled. Write a program to determine if there are enough problems for the contests. Please keep in mind that, if a problem is created in month X , it can only be used in month $X + 1$ & the later months.

最近程式競賽非常頻繁。這對參賽者來說是件好事，但對出題者來說卻是件壞事。目前出題者尚能維持一個題庫並說：「沒有問題！」，不過讓人懷疑繼續這樣下去不知還能維持多久。

給你一年中每個月所出的題目數量及各月所需要的題目數量。如果某個月需要 N 個題目，而當時的題庫數量不足，那麼該月的所有比賽均取消。請寫個程式來判斷是否有足夠的題目來辦比賽。記住，如果某個題目是在 X 月出的，該題目必須在 $X+1$ 月或其後的月份才能使用。

Input

The first line of every test case has an integer S ($0 \leq S \leq 100$). Number of problems that is ready at the beginning of the year. The 2-nd line has 12 space separated integers, denoting the number of problems created in each of the 12 months of that year. The months are in the same order as they appear in a year. The 3-rd line has another 12 space separated integers, the number of problems required to use in contests in those 12 months (With the same order as above). These integers will be between 0 & 20 (inclusive). The end of input will be denoted by a negative integer.

每筆測資的第一行有一個整數 S ($0 \leq S \leq 100$)，表示年初已準備的庫存題目數量。第二行有 12 個以空白隔開的整數，依序表示一到十二月每個月所出的題目數量。第三行也有 12 個以空白隔開的整數，依序表示每個月比賽所需要的題目數量。

這些整數會介於 0 到 20 之間 (含)。負數代表輸入的結束。

Output

For each test case, print a line of the form, `Case X :`, where X is the case number. Then print 12 lines. If there are enough problems to meet the requirements in month i ($1 \leq i \leq 12$), print `No problem! :D` in the i -th line, otherwise print `No problem. :(

每筆測資先輸出格式為 "Case X:"於第一行,其中 X 代表測資編號。接著印出 12 行,如果 i 月($1 \leq i \leq 12$) 有足夠的題目,則在第 i 行印出 "No problem! :D" (沒有問題), 否則印出 "No problem. :(" (沒有題目)。

Sample Input

```
5
3 0 3 5 8 2 1 0 3 5 6 9
0 0 10 2 6 4 1 0 1 1 2 2
-1
```

Sample Output

```
Case 1:
No problem! :D
No problem! :D
No problem. :(
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
```

Analysis

簡單題, 設一個變數存總題數, 在每年去做加減比較陣列即可。要小心題目是在 X 月出的, 須在 X 月之後才能加入總數。

11597 Spanning Subtree

分級：1

分類：圖論

Let K_n denote the complete undirected graph with n vertices where n is an even number. In other words, K_n is a graph with n vertices where every two vertices are connected. Your task is to find the maximum number of spanning trees of K_n that can be formed in such a way that no two of these spanning trees have a common edge.

假設 K_n 為 n 個節點的完全無向圖，換句話說， K_n 中兩個點皆連通。這些擴張樹之間皆沒有共同的邊，你的任務就是去找出 K_n 最大數量的所有擴張樹。

Input

Each test case will have an even integer n ($2 \leq n \leq 400$), the number of vertices. The last test case will be followed by a single '0' denoting end of input.

每組測試資料會給定一個偶數 n ($2 \leq n \leq 400$) 表示節點數，當 $n = 0$ 表示測試資料結束。

Output

For each test case, print a line in the format, 'Case X : Y', where X is the case number & Y is the maximum possible number of spanning trees.

每組測資，請以 "Case X: Y" 的格式輸出於一行，X 代表測試資料編號，而 Y 表示最大可能的不共邊擴張樹數量。

Sample Input

```
4
0
```

Sample Output

```
Case 1: 2
```

Analysis

簡單題，設一個變數存總題數，在每年去做加減比較陣列即可。要小心題目是在 X 月出的，須在 X 月之後才能加入總數。

11586 Train Tracks

分級：1

分類：排列

Andy loves his set of wooden trains and rail- road tracks. Each day, Daddy has to build a new track for him. The tracks he likes best form a simple loop with no branches or dead ends, so he can run his trains around and around for hours until it is time for the big crash that destroys the whole construction.

So here is the question: Given a set of track pieces, can you form a simple loop with them, while using up all the pieces?

Each piece of track is described by the connectors at both ends. A standard piece has one "male" and one "female" connector. But there are also track pieces with two male or two female connectors, as shown picture below.

To fit together, each male connector must be connected to a female connector. Unlike real wooden tracks, our pieces are assumed to be flexible, so their length or shape is not an issue here. However, you may not connect the two ends of the same piece together.



Some wooden pieces

Andy 很喜歡木製的火車與軌道，而每一天他爸爸都會組合一個新的軌道給他。Andy 最喜歡這個軌道沒有死路的設計，也就是說這個軌道是環狀，可以讓車子永無止境的跑下去，直到毀壞。

所以這裡有一個問題，給你一些軌道片段，你是否可以用全部給你的軌道片段拼成一個環形軌道。

每一個標準片段的軌道有兩端，為"male"跟 "female"，但是也有一些的兩端都是"male"或 "female"，如下圖所示。

每兩塊連結方式一定要一端是 male 另一端是 female，不能是兩個相同 male 或是 female 連接。這不是真實的木製軌道，所以每一個軌道片段不考慮其長度或形狀。

Input

Input begins with the number of test cases. Each following line contains one test case. Each test case consists of a list of between 1 and 50 (inclusive) train track pieces. A piece is described by two code letters: ' M ' for male or ' F ' for female connector. Pieces are separated by space characters.

輸入第一行說明有幾組測試資料，接下來每一行都代表一組的測試資料，包含 1~50(含)個軌道片段，一片的兩端以 M 跟 F 來表示，M = male, F = female，每個軌道以空白隔開。

Output

For each test case, output a line containing either ' LOOP ' or ' NOLOOP ' to indicate whether or not all the pieces can be joined into a single loop.

對於每個測試資料，印出一行說明是否可以拼成一個環，若可以請輸出 "LOOP" 不行則輸出 "NO LOOP"。

Sample Input

```
4
MFMF
FMFFMFMM
MMFF
MFMFMFMFFF
```

Sample Output

```
LOOP
LOOP
LOOP
NOLOOP
```

Analysis

按照題意，軌道可以是旋轉的，所以計算 M、F 值數量是否相等來判斷即可。要記住當 M、F 均為 1，是無法成為環狀的。

11579 Triangle Trouble

分級：2

分類：幾何

There is trouble at the triangle factory. The triangle assembler has gone down, so all that has been produced for the day is a bunch of triangle sides. To make the best of this situation, it has been decided to create the triangle with the largest possible area from the available sides, and sell it as a limited edition triangle.

You have been hired to write a program that will determine the area of the limited edition triangle.

今天三角形工廠發生了一個很大的麻煩。三角形生產器速度下降，導致現在只產生出一些三角形的邊來，為了對這個情形有最佳解決方案，決定利用這些邊來創造出最大的三角形，並出售它作為限量的三角形。

因此雇用你來寫一個程式來幫助他們尋找這三角形。

Input

Input begins with the number of test cases on its own line. Each test case begins with a positive integer N ($3 \leq N \leq 10,000$), followed by N positive real numbers s_i representing the lengths of the available triangle sides ($0 < s_i \leq 100,000$). A single test case may be spread out over several consecutive lines of the input.

輸入的第一行有一數字表示測資筆數，而每筆測資開始有一正整數 N ($3 \leq N \leq 10,000$) 代表接下來有 N 個實數 s_i 代表可以使用的三角形長度 ($0 < s_i \leq 100,000$)，一組測試資料可能被分成好幾行。

Output

For each test case, output a line containing the largest possible area of a triangle built using three of the given sides (as a real number rounded to 2 decimal places). If it is not possible to construct any triangles then output "0.00" (quotes for clarity).

每筆測資輸出一行，印出給予的邊長所能為出的最大三角形面積(四捨五入到小數點第二位)。如果找不到請印出"0.00"

Sample Input

2

4 3.0 4.0 5.0 100.0

3 1.0 2.0 4.0

Sample Output

6.00

0.00

Analysis

先將資料由大排到小，以連續三個一組，看看使否能夠成三角形，再利用海龍公式求面積

11577 Letter Frequency

分級：1

分類：字元與字串

In this problem we are interested in the frequency of letters in a given line of text. Specifically, we want to know the most frequently occurring letter(s) in the text, ignoring case (to be clear, "letters" refers precisely to the 26 letters of the alphabet).

在這次問題中，我們對一行文字中字母出現頻率很有興趣，且特別想知道頻率最高的字母（"letters"指的是字母表中的 26 個字）是什麼。請忽略一些非字母的文字。

Input

Input begins with the number of test cases on its own line. Each test case consists of a single line of text. The line may contain non-letter characters, but is guaranteed to contain at least one letter and less than 200 characters in total.

輸入開始有一數字代表有幾個測試資料。每一測試資料為一行，而這一行可能有空白，不過至少會有一個字母。一行字母全部加起來不超過 200 個。

Output

For each test case, output a line containing the most frequently occurring letter(s) from the text in lowercase (if there are ties, output all such letters in alphabetical order).

對每個測試資料，輸出頻率最高的小寫字母。（如果超過兩個一樣，請照字母表順序輸出）

Sample Input

1

Computers account for only 5% of the country's commercial electricity consumption.

Sample Output

co

Analysis

按照題意統計即可。大小寫須一起算，且空白字元不用計算。

11538 Chess Queen

分級：2

分類：數學計算

You probably know how the game of chess is played and how chess queen operates. Two chess queens are in attacking position when they are on same row, column or diagonal of a chess board. Suppose two such chess queens (one black and the other white) are placed on (2×2) chess board. They can be in attacking positions in 12 ways, these are shown in the picture below:

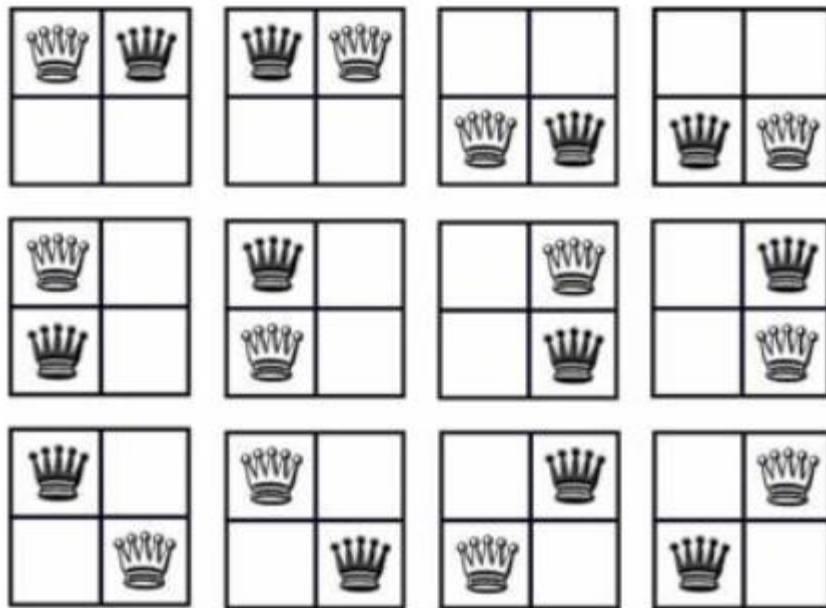


Figure: in a (2×2) chessboard 2 queens can be in attacking position in 12 ways

Given an $(N \times M)$ board you will have to decide in how many ways 2 queens can be in attacking position in that.

西洋棋棋盤上有一個白色的皇后與一個黑色的皇后，彼此可以互相攻擊的位置是在同一列，或同一行，或是在對角線的位置上。假設這兩個皇后被放置在 2×2 的棋盤上，則共有 12 種可以互相攻擊的位置，如圖：

本題給定棋盤的大小 $(N \times M)$ ，請你計算兩個皇后可以彼此相互攻擊的位置總數。

Input

Input file can contain up to 5000 lines of inputs. Each line contains two non-negative integers which denote the value of M and N ($0 < M, N \leq 10^6$) respectively.

Input is terminated by a line containing two zeroes. These two zeroes need not be processed.

輸入資料最多 5000 列，每列給定兩個非負整數 $N, M(0 < N, M \leq 10^6)$ 。

當 $N = M = 0$ 時表示測試資料結束。

Output

For each line of input produce one line of output. This line contains an integer which denotes in how many ways two queens can be in attacking position in an $(M \times N)$ board, where the values of M and N came from the input. All output values will fit in 64-bit signed integer.

請每組資料輸出一列，表示 $N \times M$ 棋盤上所有可能攻擊位置的總數，輸出值大小可用 64 位元有號整數表示。

Sample Input

```
2 2
100 223
2300 1000
0 0
```

Sample Output

```
12
10907100
11514134000
```

Analysis

因為皇后的攻擊範圍為分隔號、橫線和斜線，所以枚舉每條上兩個皇后放的位置，共有三種情況需推導：

1. 兩個皇后排一行 $a * C(b, 2) * 2!$
2. 兩個皇后排一列 $b * C(a, 2) * 2!$
3. 兩個皇后排一對角線 $2 * [(b - a + 1) * C(a, 2) + 2 * C(a, 3)]$

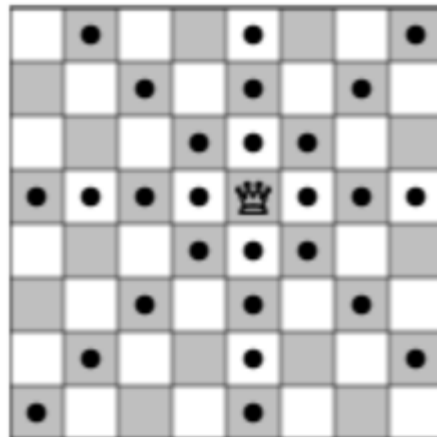
將上面三式加總即為解。

11494 Queen

分級：1

分類：模擬

The game of Chess has several pieces with curious movements. One of them is the Queen, which can move any number of squares in any direction: in the same line, in the same column or in any of the diagonals, as illustrated by the figure below (black dots represent positions the queen may reach in one move):



The great Chess Master Kary Gasparov invented a new type of chess problem: given the position of a queen in an empty standard chess board (that is, an 8×8 board) how many moves are needed so that she reaches another given square in the board?

Kary found the solution for some of those problems, but is having a difficult time to solve some others, and therefore he has asked that you write a program to solve this type of problem.

西洋棋中有幾個子的走法滿特別的，其中一個就是皇后。她可以循垂直、水平、或對角線的方向隨她走幾格，如圖（黑點表示皇后可以一步走到的格子）：

西洋棋大師 Kary Gasparov 提出了一新的問題：在標準的西洋棋空棋盤（ 8×8 棋盤）上擺一個皇后，它要走幾步才能走到某個特定的格子？

Kary 已經找到了某些情況的解，但有些情況他卻解不出來。因此他請你幫他寫一個程式來解這個問題。

Input

The input contains several test cases. The only line of each test case contains four integers $X1$, $Y1$, $X2$ and $Y2$ ($1 \leq X1, Y1, X2, Y2 \leq 8$). The queen starts in the square

with coordinates $(X1, Y1)$, and must finish at the square with coordinates $(X2, Y2)$. In the chessboard, columns are numbered from 1 to 8, from left to right; lines are also numbered from 1 to 8, from top to bottom. The coordinates of a square in line X and column Y are (X, Y) .

The end of input is indicated by a line containing four zeros, separated by spaces.

輸入檔包含了好幾筆測試資料。每筆測試資料只有一行，其中含有整數 $X1, Y1, X2$ 及 $Y2$ ($1 \leq X1, Y1, X2, Y2 \leq 8$)。皇后從座標 $(X1, Y1)$ 的格子開始，必須在座標 $(X2, Y2)$ 的格子結束。在棋盤中的行由左至右編號為 1 到 8，列則由上至下編號為 1 到 8。位於第 X 列第 Y 行的格子其座標為 (X, Y) 。

輸入的結尾以四個由空白隔開的 0 來表示。

Output

For each test case in the input your program must print a single line, containing an integer, indicating the smallest number of moves needed for the queen to reach the new position.

對於每筆測試你的程式要印出一行，該行含有一個整數，表示皇后要走到新的位置至少需要幾步。

Sample Input

```
4 4 6 2
3 5 3 5
5 5 4 3
0 0 0 0
```

Sample Output

```
1
0
2
```

Analysis

皇后要到任何一格的步數只在 0~2 之間：

- 0 步：從自己這格走到自己這格。
- 1 步：在自己米字範圍內的格子。
- 2 步：非 0 步、1 步走到的格子。

11479 Is this the easiest problem?

分級：1

分類：模擬

A triangle is a geometric shape with three positive sides. However, any given three sides won't necessarily form a triangle. The three sides must form a closed region. Triangles are categorized depending on the values of the sides of a valid triangle. In this problem you are required to determine the type of a triangle.

三角形是一種有三個正邊的幾何形狀。但是並非任意三個邊都可以形成一個三角形，這三個邊必需構成一個封閉區域。三角形可以其邊長來分類，本題便是要你判定三角形的種類。

Input

The first line of input will contain a positive integer $T < 20$, where T denotes the number of test cases. Each of the next T lines will contain three 32 bit signed integer.

輸入的第一行有一個正整數 $T < 20$ ，表示有 T 筆測試資料。接下來的 T 行每行有三個 32 位元的有號整數。

Output

For each case of input there will be one line of output. It will be formatted as:

Case x : triangle type.

Where x denotes the case number being processed and triangle type is the type of the triangle. triangle type will be one of the following, depending on the values of the three sides:

- Invalid - The three sides can't form a triangle
- Equilateral - All three sides of valid triangle are equal
- Isosceles - Exactly two of the sides of a valid triangle are equal.
- Scalene - No pair of sides are equal in a valid triangle.

對於每筆測資請輸出一行，其格式為：Case x : triangle type。

其中 x 表示測資編號，triangle type 為三角形的種類。依三邊長的值，triangle type 是以下的一種：

- Invalid - 這三邊無法形成一個三角形

- Equilateral - 三邊均等長
- Isosceles - 恰有兩邊等長
- Scalene - 三邊均不等長

Sample Input

4

1 2 5

1 1 1

4 4 2

3 4 5

Sample Output

Case 1: Invalid

Case 2: Equilateral

Case 3: Isosceles

Case 4: Scalene

Analysis

依照題目判斷即可。

11461 Square Numbers

分級：1

分類：數學計算

A square number is an integer number whose square root is also an integer. For example 1, 4, 81 are some square numbers. Given two numbers a and b you will have to find out how many square numbers are there between a and b (inclusive).

完全平方數就是一個平方根為整數的整數，例如 1, 4, 81 就是完全平方數。給你兩個整數 a 和 b，請你找出 a 與 b 之間 (含) 有幾個完全平方數。

Input

The input file contains at most 201 lines of inputs. Each line contains two integers a and b ($0 < a \leq b \leq 100000$). Input is terminated by a line containing two zeroes. This line should not be processed.

輸入最多有 201 行的測資，每筆包含兩個整數 a 和 b ($0 < a \leq b \leq 100000$)。若 $a = b = 0$ 則測資結束，且此行不須執行。

Output

For each line of input produce one line of output. This line contains an integer which denotes how many square numbers are there between a and b (inclusive).

每筆輸入皆產生一行輸出。此行含一整數表示 a 與 b 之間(含)會有幾個完全平方數。

Sample Input

```
1 4
1 10
0 0
```

Sample Output

```
2
3
```

Analysis

依照題目判斷即可。

11455 Behold My Quadrangle

分級：1

分類：幾何

Any square is a rectangle, any rectangle is a quadrangle, and any quadrangle is composed of four sides. But not all rectangles are squares, not all quadrangles are rectangles, and not all sets of four sides are quadrangles.

We have the length of four sides. You have to determine if they can form a square. If not, determine if they can form a rectangle. If not, determine if they can form a quadrangle.

任一正方形都是矩形，任一矩形都是四邊形，且任一四邊形都有四個邊。但是並非所有的矩形都是正方形，並非所有的四邊形都是矩形，並非任四個邊都可以構成四邊形。

目前我們有了四個邊的長度，你必須判斷它們是否可以構成一個正方形。不行的話，判斷它們是否可以構成一個矩形。不行的話，判斷它們是否可以構成一個四邊形。

Input

The first line of the input contains an integer indicating the number of test cases.

For each test case, there is a line with four positive integer numbers, between 0 and 2^{30} .

輸入的第一行包含一個整數表示有幾筆測試資料。

每筆測試資料一行，包含四個介於 0 和 2^{30} 的正整數。

Output

For each test case, the output should consist of a line with the text "square", "rectangle", "quadrangle" or "banana", if the sides of the corresponding case can form a square, a rectangle, a quadrangle or none, respectively.

就每筆測試資料請輸出於一列，如果該筆料可以構成正方形、矩形、四邊形、或都不行，請分別輸出「square」、「rectangle」、「quadrangle」或「banana」。

Sample Input

4

10 8 7 6

9 1 9 1

29 29 29 29

5 12 30 7

Sample Output

quadrangle

rectangle

square

banana

Analysis

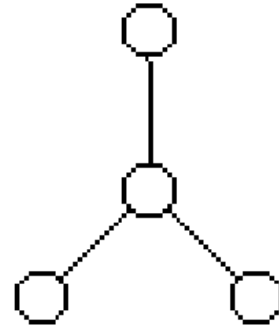
依照題目判斷即可。

11396 Claw Decomposition

分級：2

分類：圖論

A claw is defined as a pointed curved nail on the end of each toe in birds, some reptiles, and some mammals. However, if you are a graph theory enthusiast, you may understand the following special class of graph as shown in the following figure by the word claw.



If you are more concerned about graph theory terminology, you may want to define claw as $K_{1,3}$.

Let's leave the definition for the moment and come to the problem. You are given a simple undirected graph in which every vertex has degree 3. You are to figure out whether the graph can be decomposed into claws or not.

Just for the sake of clarity, a decomposition of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

“爪”被定義為鳥類、一些爬行動物與一些哺乳動物腳末端彎曲的尖鉤。但若你是一個圖論愛好者，你可以了解這特殊爪圖。

如果你更在意它圖論中的術語，你可以定義爪為 $K_{1,3}$ 。

現在我們捨棄這定義，並提出個問題：在一個簡單的無向量圖中，每個點度為 3，你要判斷是否能將圖分解成一個個的爪。

為了清楚起見，無向量圖將分解成多個的爪，且每條邊只能出現在一個爪型中。

Input

There will be several cases in the input file. Each case starts with the number of vertices in the graph, V ($4 \leq V \leq 300$). This is followed by a list of edges. Every line in the list has two integers, a and b , the endpoints of an edge ($1 \leq a, b \leq V$). The edge list ends with a line with a pair of '0'. The end of input is denoted by a case with $V = 0$. This case should not be processed.

輸入有多筆測試資料，每筆測資開始有一數字 $V(4 \leq V \leq 300)$ 表示無向圖中有多

少個點。接下來為邊的資料，每列有 2 個整數 a 和 b ($1 \leq a, b \leq V$) 指一條邊的兩端點，當 $a = b = 0$ 表示邊資料的結束。若 $V = 0$ 則輸入結束，且不須處理。

Output

For every case in the input, print `YES' if the graph can be decomposed into claws and `NO' otherwise.

對於每筆測資，如果將圖分解成一個個的爪便印出`YES'，反之則印`NO'。

Sample Input

```
4
1 2
1 3
1 4
2 3
2 4
3 4
0 0
6
1 2
1 3
1 6
2 3
2 5
3 4
4 5
4 6
5 6
0 0
0
```

Sample Output

```
NO
NO
```

Analysis

將爪的中心點視為關鍵點，對於每個關鍵點隻屬於唯一的爪中，對於關鍵點周圍的 3 個點均不可能成為其它爪的關鍵點（否則與題目不符），所以原圖可以分解

為關鍵點與非關鍵點，關鍵點與關鍵點之間沒有邊相連，同理非關鍵點之間也沒有邊，那麼此題就可以轉化為二分圖的判定問題，使用二染色的模版即可。

11388 GCD LCM

分級：2

分類：數學計算

The GCD of two positive integers is the largest integer that divides both the integers without any remainder. The LCM of two positive integers is the smallest positive integer that is divisible by both the integers. A positive integer can be the GCD of many pairs of numbers. Similarly, it can be the LCM of many pairs of numbers. In this problem, you will be given two positive integers. You have to output a pair of numbers whose GCD is the first number and LCM is the second number.

最大公因數(GCD)是兩個正整數的最大相同除數；最小公倍數(LCM)是可以除以兩個正整數的最小整數。一個正整數可以有很多對的 GCD 與 LCM。在這題中會給你 a, b 的最大公因數(GCD)與最小公倍數(LCM)，請你求出 a, b 為何。

Input

The first line of input will consist of a positive integer T ($T \leq 100$). T denotes the number of cases. Each of the next T lines will contain two positive integer, G and L .

輸入第一列包含一正整數 T ($T \leq 100$)，代表有多少的測資。每筆測資為一行皆有 2 個整數 G 和 L ($G, L \leq 2^{31}$)。

Output

For each case of input, there will be one line of output. It will contain two positive integers a and b , $a \leq b$, which has a GCD of G and LCM of L . In case there is more than one pair satisfying the condition, output the pair for which a is minimized. In case there is no such pair, output `-1`.

每組測試資料輸出一列，請印出兩個正整數 a 與 b ($a \leq b$)，且 a, b 的最大公因數 G ，最小公倍數 L 。由於可能會有許多可能的解，所以請輸出 a 為最小的那組，如果無解則輸出 `-1`。

Sample Input

```
2
1 2
3 4
```

Sample Output

1 2

-1

Analysis

題目要求 a 為最小時，所以可知 $a = G$ ，因此只要 $L \% G == 0$ ，就表示存在。

11342 Three-square

分級：1

分類：數學計算

Lagrange's four-square theorem states that every positive integer can be expressed as the sum of four squares of integers. For example:

$$3 = 1^2 + 1^2 + 1^2 + 0^2$$

$$31 = 5^2 + 2^2 + 1^2 + 1^2$$

However some positive integers can be expressed even as the sum of three squares of non-negative integers. For example:

$$3 = 1^2 + 1^2 + 1^2$$

$$17 = 0^2 + 1^2 + 4^2$$

In this problem you have to find expression of given integer K as the sum of three squares, or state that it is impossible.

根據 Lagrange 的四平方和定理，每個正整數可以用四個完全平方數的和來表示。例如：

$$3 = 1^2 + 1^2 + 1^2 + 0^2$$

$$31 = 5^2 + 2^2 + 1^2 + 1^2$$

可是有些正整數可以只用三個非負的完全平方和來表示。例如：

$$3 = 1^2 + 1^2 + 1^2$$

$$17 = 0^2 + 1^2 + 4^2$$

此題中會給你一個整數 K ，請你以三個平方數的和來表示，或說明這不可能成立的。

Input

The first line contains integer N ($0 < N \leq 10000$), it is number of tests. Each of the next N lines contains a positive integers K ($0 < K \leq 50000$).

第一行包含一整數 $N(0 < N \leq 10000)$ 代表測資筆數。接下來的 N 列每列皆有一正

整數 K ($0 < K \leq 50000$)。

Output

For each test case print a line formatted like this: `a b c`. Where $a \leq b \leq c$ and $K = a^2 + b^2 + c^2$. If there is more than one possible answer, print the one that comes first lexicographically. If expression in three squares of non-negative integers do not exist print `-1` (see examples).

對於每一筆測資，請以"a b c"的格式印出一行。其中 $a \leq b \leq c$ 且 $K = a^2 + b^2 + c^2$ 。如果答案不只一個，就印出字典順序最小的。而若無法以三個非負平方數來表示則印"-1"（請參考範例）。

Sample Input

3
13
15
17

Sample Output

0 2 3
-1
0 1 4

Analysis

題目要求 a 為最小時，所以可知 $a = G$ ，因此只要 $L \% G == 0$ ，就表示存在。

11321 Sort! Sort!! and Sort!!!

分級：1

分類：排序

Hmm! Here you are asked to do a simple sorting. You will be given N numbers and a positive integer M . You will have to sort the N numbers in ascending order of their modulo M value. If there is a tie between an odd number and an even number (that is their modulo M value is the same) then the odd number will precede the even number. If there is a tie between two odd numbers (that is their modulo M value is the same) then the larger odd number will precede the smaller odd number and if there is a tie between two even numbers (that is their modulo M value is the same) then the smaller even number will precede the larger even number.

For remainder value of negative numbers follow the rule of C programming language: A negative number can never have modulus greater than zero. E.g. $-100 \text{ MOD } 3 = -1$, $-100 \text{ MOD } 4 = 0$, etc.

嗨！來做簡單的排序吧！給你兩個整數 N 與 M ，請你將 N 個整數依照除以 M 後的餘數由小到大排序。若排序中比較的兩數取模數後其值相同，在此情況下分三種情況進行討論：

- 若兩個元素值分別為一個奇數與一個偶數，則將奇數排在偶數前面。
- 若兩個元素皆為奇數，則較大的奇數將排在較小的奇數前面。
- 若兩個元素皆為偶數，則較小的偶數將排在較大的偶數前面。

至於負數的餘數計算以 C 語言取餘數的方式為準，即負數的餘數絕對不會大於零。例如 $-100 \text{ MOD } 3 = -1$, $-100 \text{ MOD } 4 = 0$ 依此類推。

Input

The input file contains 20 sets of inputs. Each set starts with two integers N ($0 < N \leq 10000$) and M ($0 < M \leq 10000$) which denotes how many numbers are within this set. Each of the next N lines contains one number each. These numbers should all fit in 32-bit signed integer. Input is terminated by a line containing two zeroes.

輸入有 20 組測試資料，每組資料一開始有兩個整數 N ($0 < N \leq 10000$) 與 M ($0 < M \leq 10000$)，表示接下來會有 N 列皆包含一整數，每個整數皆能以 32 位元的有號整數來表示。當 $N = M = 0$ 時表示測試資料結束。

Output

For each set of input produce $N + 1$ lines of outputs. The first line of each set contains the value of N and M . The next N lines contain N numbers, sorted according to the rules mentioned above. Print the last two zeroes of the input file in the output file also.

對於每一組輸入請輸出 $N+1$ 行整數。第一行為兩個整數 N, M 。接下來的 N 列請分別輸出排序後的陣列元素。對於輸入測資尾端的兩個 $0 0$ 請也輸出以空白分隔的 $0 0$ 。

Sample Input

```
15 3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0 0
```

Sample Output

```
15 3
15
9
3
6
12
13
7
1
4
```

10
11
5
2
8
14
00

Analysis

按照題目要求排序即可。

11222 Only I did it!

分級：1

分類：數學計算

Once upon a time, in the land of Ceeplenty, lived 3 friends that liked to solve problems. They used internet engines to look for problems and they often tried to solve the problems that none of the other 2 had solved. They once met you and you managed to convince them that you were better at problem solving. So they asked you to write a program that tells which of the 3 friends solved more problems that none of the other 2 have solved.

很久很久以前，在 Ceeplenty 住著三個很愛解題的朋友。他們習慣用搜尋引擎來找題目，並往往試著去解其他兩人未曾解過的題目。有一次他們遇到了你，你成功說服他們你比他們還會解題。因此他們請你寫一個程式來判定這三個朋友中那一個人解了比較多其他兩人未曾解過的題目。

Input

The first line of input gives the number of cases, T ($1 \leq T \leq 20$). T test cases follow. Each test case is composed of three lines corresponding to the problems solved by the first, second and third friend, respectively. Each of these lines has an integer S ($0 \leq S \leq 1000$) followed by the list of S solved problems. A solved problem is identified uniquely by a positive integer smaller or equal than 10000.

輸入的第一行有一整數 T ($1 \leq T \leq 20$) 表示測試的筆數。接下來每筆測試有三行分別對應到第一、第二、及第三個朋友所解出的題目，而每一行的開頭皆有一個整數 S ($0 \leq S \leq 1000$) 然後會列出 S 個已解決的題目編號清單。題目編號以小於等於 10000 的正整數來識別。

Output

The output is comprised a line identifying the test case with the string Case #C: (where C is the number of the current test case). Then print another line with the number of the friend (1, 2 or 3) asked in the description followed by the number of problems that he solved but none of the other 2 did, followed by the sorted list of these problems in one line. When there is a tie, print one such line for each tied friend, sorted by the number of the friend.

每筆測試輸出一行字串「Case #C:」(其中 C 為目前這筆測試的編號)。接下來那行先印出題目所要求的那個朋友的編號(1, 2 or 3)再印出他解了幾題其他兩人未

解出的題目，並依序列出這些題目。如果平手，每個平手的朋友都依編號順序印出一行。

Sample Input

```
4
3 1 2 3
4 4 5 6 7
5 8 9 10 11 12
2 1 5
2 2 3
3 2 3 1
6 400 401 402 403 404 405
2 101 100
7 400 401 402 403 404 405 406
1 1
1 2
1 3
```

Sample Output

```
Case #1:
3 5 8 9 10 11 12
Case #2:
1 1 5
Case #3:
2 2 100 101
Case #4:
1 1 1
2 1 2
3 1 3
```

Analysis

使用陣列紀錄總題數與每個人所解題的題目，再找出解別人沒解的題目最多者，按照題目要求輸出即可。

11188 Ternary

分級：1

分類：數學計算

You will be given a decimal number. You will have to convert it to its ternary (Base 3) equivalent.

你將會得到一個十進制的數字，請你將它轉換成它三進制的數值。

Input

The input file contains at most 100 lines of inputs. Each line contains a non-negative decimal integer N ($N < 1000000001$). Input is terminated by a line containing a negative value. This line should not be processed.

輸入含有最多 100 列的測資。每列包含一個非負十進制整數 N ($N < 1000000001$)。輸入以一含負值的列為終止，且不會被處理。

Output

For each line of input produce one line of output. This line contains the ternary equivalent of decimal value N .

對於每筆測資皆輸出於一列，來顯示 N 值所對應的三進制數值。

Sample Input

```
10
100
1000
-1
```

Sample Output

```
101
10201
1101001
```

Analysis

將 N 以迴圈來除以 3 取餘數並紀錄即可印出。

11094 Continents

分級：2

分類：圖論

Mijid the Great is the king of Dodars territory. He likes to travel between the cities in his territory and actually, you can never see him in the same city as where he was the day before. Therefore, he captured all territories of his continent! In spite of this fact, he has seen all cities of his territory so far and wants to capture another continent in order to have some choices to travel into new cities. Now, having the world map, he needs your help to find the biggest continent except the one in which he resides.

Maps are given as $M \times N$ tables, filled with at most two different letters denoting land and water regions. A continent is a set of connected land regions which is completely surrounded by water regions or the end of map. Two regions are assumed to be connected if they have an edge in common. The coordinates of top left region is $(0, 0)$ and bottom right region $(M - 1, N - 1)$. Region with coordinates $(x, N - 1)$ should be assumed to have a common edge with region $(x, 0)$ for every x between 0 and $M - 1$ (inclusive).

Mijid 是 Dodars 領土的國王，他喜歡在自己領土上的城市之間旅行。實際上，你永遠無法看到他在同一個城市待上兩天。因此，他佔領了目前大陸上所有的領土！儘管是這樣，他也已經到過了這大陸上的所有城市，所以他希望奪取另一塊大陸，以便有新的城市可選擇來旅行。現在，他得到了一份世界地圖，並需要你的幫助來找到除了在他所居住之地以外的另一個最大洲。

地圖將分為 $M \times N$ 的格子，充滿了兩個不同的字母表示土地和水的地區。而大陸定義為一塊會完全被水或地圖邊緣所包圍的相連土地，而若兩個區域在共同的邊緣上，即視為它們是連接的。假定左上區域的坐標是 $(0, 0)$ ；右下區域坐標 $(M - 1, N - 1)$ ，且右邊的邊界坐標 $(x, N - 1)$ 的區域與左邊的邊界 $(x, 0)$ 的區域為互通的。每個 x 介於區間 0 和 $M - 1$ (含) 中。

Input

There will be several test cases. Each test case contains two integers $M \leq 20$ and $N \leq 20$ in the first line denoting the number of rows and columns in the map respectively. Next, there will be M lines of exactly N characters representing the map. Finally in the last line there would be two integers $0 \leq X < M$ and $0 \leq Y < N$, the coordinates of the region in which Mijid the Great currently stays. There will one blank line after each test case.

輸入有多筆的測資，每筆測資第一列包含兩個整數 M ($M \leq 20$) 和 N ($N \leq 20$)，意旨接下來有 M 列皆包含 N 個字元來代表地圖的長寬。而最後一列包含兩整數 X ($0 \leq X < M$) 和 Y ($0 \leq Y < N$) 代表 Mijid 原先的位置。每筆測資以一空白列為間隔來輸入。

Output

For each test case, output a line containing an integer that is the number of regions in the biggest continent that Mijid the Great can capture.

對於每筆測資，輸出一列包含一整數指 Mijid 可以奪取的最大陸地。

Sample Input

```
5 5
wwwww
wllw
wwwww
wllww
wwwww
1 3
```

Sample Output

```
2
```

Analysis

把國王目前位置用 BFS 搜索其他陸地，找到再用 BFS 找完所有相鄰土地的個數，最後輸出最大的土地即可。

11054 Wine trading in Gergovia

分級：2

分類：模擬

As you may know from the comic "Asterix and the Chieftain's Shield", Gergovia consists of one street, and every inhabitant of the city is a wine salesman. You wonder how this economy works? Simple enough: everyone buys wine from other inhabitants of the city. Every day each inhabitant decides how much wine he wants to buy or sell. Interestingly, demand and supply is always the same, so that each inhabitant gets what he wants.

There is one problem, however: Transporting wine from one house to another results in work. Since all wines are equally good, the inhabitants of Gergovia don't care which persons they are doing trade with, they are only interested in selling or buying a specific amount of wine. They are clever enough to figure out a way of trading so that the overall amount of work needed for transports is minimized.

In this problem you are asked to reconstruct the trading during one day in Gergovia. For simplicity we will assume that the houses are built along a straight line with equal distance between adjacent houses. Transporting one bottle of wine from one house to an adjacent house results in one unit of work.

正如你可能從漫畫"Asterix and the Chieftain's Shield"知道，Gergovia 城市的某條街道中，每一位居民都是酒的推銷員。而你想知道其經濟如何運作？這很簡單，每個人都會從城市中的其他居民購買葡萄酒，所以每一天居民們會決定他有多少酒想要購買或出售。有趣的是，需求和供給始終是相同的，而每個居民都會得到所想要的了。

而其中存在著一個問題：將葡萄酒從一間房子送到另外一間是很費力的，因為所有的酒品質都很好，Gergovia 的居民並不在意跟誰交易，他們只關注買了或賣了多少酒，因此他們會用聰明的腦袋來找出花費最少力氣來運送這些葡萄酒的方法。

所以在這問題中，你被請求去重建 Gergovia 一天的貿易，為了簡化問題，我們假設街上相鄰兩棟房子之間的距離是一樣的，從一棟房子運送一瓶葡萄酒到相鄰的房子需花費一單位的力氣。

Input

The input consists of several test cases. Each test case starts with the number of inhabitants n ($2 \leq n \leq 100000$). The following line contains n integers A_i ($-1000 \leq A_i$

≤ 1000). If $A_i \geq 0$, it means that the inhabitant living in the i -th house wants to buy A_i bottles of wine, otherwise if $A_i < 0$, he wants to sell $-A_i$ bottles of wine. You may assume that the numbers A_i sum up to 0.

The last test case is followed by a line containing `0`.

輸入包含數組測試資料。每組測試資料一開始為 n ($2 \leq n \leq 100000$)，接下來的一行有 n 個整數 A_i ($-1000 \leq A_i \leq 1000$)，如果 $A_i \geq 0$ ，代表住在第 i 棟房子的居民想要買 A_i 瓶紅酒；如果 $A_i < 0$ ，則想賣 $-A_i$ 瓶。你可以假設所有 A_i 的總和為 0。

最後一組測試資料為只有包含 0 的一行。

Output

For each test case print the minimum amount of work units needed so that every inhabitant has his demand fulfilled. You may assume that this number fits into a signed 64-bit integer (in C/C++ you can use the data type "long long", in JAVA the data type "long").

對於每組測資，印出滿足每位居民需求所花費的最小力氣。你可以假設這個數字可以用 64 位元有號整數表示（在 C/C++ 中，你可以用 long long 型態；在 JAVA 則用 long 型態）。

Sample Input

```
5
5 -4 1 -3 1
6
-1000 -1000 -1000 1000 1000 1000
0
```

Sample Output

```
9
9000
```

Analysis

這題看上去麻煩，實際上只要想清楚就很簡單。因為對酒的需求量和勞動力是一樣的，所以關鍵是以等價轉換的思維方式，讓鄰居間進行交易，即讓後一個來滿足前一個。而如果前一個是買，後一個也是買，也讓後一個來賣給前一個人。

11044 Searching for Nessy

分級：1

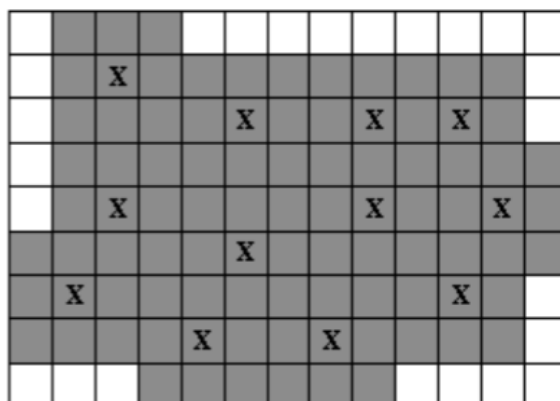
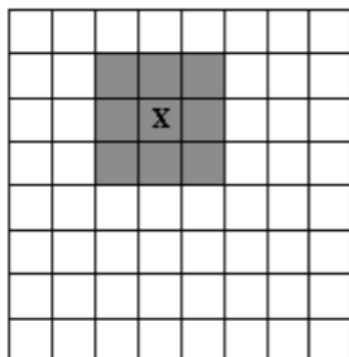
分類：模擬

In July 2003, the BBC reported an extensive investigation of Loch Ness by a BBC team, using 600 separate sonar beams, found no trace of any "sea monster" (i.e., any large animal, known or unknown) in the loch. The BBC team concluded that Nessie does not exist. Now we want to repeat the experiment.

Given a grid of n rows and m columns representing the loch, $6 \leq n, m \leq 10000$, find the minimum number s of sonar beams you must put in the square such that we can control every position in the grid, with the following conditions:

- one sonar occupies one position in the grid; the sonar beam controls its own cell and the contiguous cells;
- the border cells do not need to be controlled, because Nessy cannot hide there (she is too big).

For example,



where X represents a sonar, and the shaded cells are controlled by their sonar beams; the last figure gives us a solution.

2003 年 7 月，BBC (英國國家廣播電視公司) 曾報導了一項他們對尼斯湖所作的大規模研究，他們用 600 支聲納也沒有辦法在湖中找到任何「水怪」(也就是任何已知或未知的大型動物) 的踪跡。他們推論尼斯怪並不存在。現在我們要重覆這項實驗。

給你一個 n 列 m 行的格子代表尼斯湖， $6 \leq n, m \leq 10000$ ，找出最少要放幾個聲納才能控制所有的方格，條件如下：

- 一個聲納佔一格；它的監控範圍為所在的那一格及緊鄰的格子；
- 邊緣的格子不需要監控，因為尼斯怪太大了，無法藏在那兒。

如圖示，

其中 **X** 代表聲納，灰色區域則是它所監控的範圍。最後一個圖則是一組可接受的解答。

Input

The first line of the input contains an integer, t , indicating the number of test cases. For each test case, there is a line with two numbers separated by blanks, $6 \leq n, m \leq 10000$, that is, the size of the grid (n rows and m columns).

輸入的第一行有一個整數， t ，代表測試筆數。每筆測資一行，含有兩個由空白分開的數字， $6 \leq n, m \leq 10000$ ，也就是格子的大小 (n 列 m 行)。

Output

For each test case, the output should consist of one line showing the minimum number of sonars that verifies the conditions above.

每筆測資輸出一行，顯示符合上述條件的最小數字。

Sample Input

```
3
6 6
7 7
9 13
```

Sample Output

```
4
4
12
```

Analysis

邊界可不計，即是指說除不盡則完全捨棄，故只要 $(n/3)*(m/3)$ 即是答案。

11039 Building designing

分級：2

分類：排序

An architect wants to design a very high building. The building will consist of some floors, and each floor has a certain size. The size of a floor must be greater than the size of the floor immediately above it. In addition, the designer (who is a fan of a famous Spanish football team) wants to paint the building in blue and red, each floor a color, and in such a way that the colors of two consecutive floors are different.

To design the building the architect has n available floors, with their associated sizes and colors. All the available floors are of different sizes. The architect wants to design the highest possible building with these restrictions, using the available floors.

有個建築師要設計一棟很高的大樓。這大樓會有許多樓層，每個樓層的面積必須大於它上一層的面積。再者，設計師（他是某個西班牙足球隊的球迷）要把每層樓漆成藍色或紅色，相鄰的兩個樓層顏色必須不同。

建築師現有 n 個特定顏色與面積的樓層可供建構大樓，每個樓層的面積均不同。建築師希望在上述的條件下以現有可用的樓層建構出最高的大樓。

Input

The input file consists of a first line with the number p of cases to solve. The first line of each case contains the number of available floors. Then, the size and color of each floor appear in one line. Each floor is represented with an integer between -999999 and 999999 . There is no floor with size 0 . Negative numbers represent red floors and positive numbers blue floors. The size of the floor is the absolute value of the number. There are not two floors with the same size. The maximum number of floors for a problem is 500000 .

輸入的第一行有測資的筆數 p 。每筆測資的第一行是可用的樓層數。接下來每一行代表一個樓層的顏色與面積。每個樓層以一個介於 -999999 與 999999 間的整數表示。沒有面積為 0 的樓層。負數為紅色樓層；正數則為藍色樓層。絕對值則是面積。沒有任何兩個樓層的面積相同。最大的樓層數為 500000 。

Output

For each case the output will consist of a line with the number of floors of the highest building with the mentioned conditions.

每筆測資輸出依上述條件所得的最大樓層數於一行。

Sample Input

```
2
5
7
-2
6
9
-3
8
11
-9
2
5
18
17
-15
4
```

Sample Output

```
2
5
```

Analysis

讀入時先將正數與負數分別儲存在兩個陣列，負數直接儲存絕對值。然後將兩個陣列按照大小排序，然後在兩個陣列中交替選出儘量小的數即可。

11000 Bee

分級：1

分類：模擬

In Africa there is a very special species of bee. Every year, the female bees of such species give birth to one male bee, while the male bees give birth to one male bee and one female bee, and then they die!

Now scientists have accidentally found one "magical female bee" of such special species to the effect that she is immortal, but still able to give birth once a year as all the other female bees. The scientists would like to know how many bees there will be after N years. Please write a program that helps them find the number of male bees and the total number of all bees after N years.

在非洲有一種非常特別的蜜蜂。每一年母蜂會生一隻公蜂，而公蜂會生一隻公蜂和一隻母蜂，然後死去。

現在，科學家在偶然中發現了一隻這種品種的母蜂，而且這是一隻「神奇」的母蜂，因為她永遠都不會死，而且每年都可以像其他正常的母蜂一樣生一隻公蜂。科學家想要知道，在 N 年後會有多少隻蜜蜂。請寫一個程式幫他們算出在 N 年後公蜂的數目以及所有蜜蜂的數目。

Input

Each line of input contains an integer N (≥ 0). Input ends with a case where $N = -1$ (This case should NOT be processed.)

每組測試資料一列，有 1 個正整數 N ($N \geq 0$)。當 $N = -1$ 時代表輸入結束（且不須執行）。

Output

Each line of output should have two numbers, the first one being the number of male bees after N years, and the second one being the total number of bees after N years. (The two numbers will not exceed 2^{32} .)

每組測試資料輸出兩整數於一列，第一個數字為 N 年後公蜂的數目，第二個數字為 N 年後所有蜜蜂的數目。（這 2 個數都不會超過 2^{32} 。）

Sample Input

1
3
-1

Sample Output

1 2
4 7

Analysis

單純的加減邏輯問題。注意只有一開始那隻母蜂不會死，其餘蜜蜂生完就死了。

10931 Parity

分級：1

分類：數學計算

We define the parity of an integer n as the sum of the bits in binary representation computed modulo two. As an example, the number $21 = 10101_2$ has three 1s in its binary representation so it has parity $3(\text{mod}2)$, or 1. In this problem you have to calculate the parity of an integer $1 \leq I \leq 2147483647$.

我們定義整數 n 的「同位元」為：其二進制中每位元的和再除以 2 的餘數。例如： $21 = 10101_2$ ，二進制中有三個 1，因此它的同位元為 $3(\text{mod}2)$ 或是 1。在本題中，你要計算一個整數 $1 \leq I \leq 2147483647$ 的同位元。

Input

Each line of the input has an integer I and the end of the input is indicated by a line where $I = 0$ that should not be processed.

輸入的每一行有一個整數 I 。當 $I = 0$ 表示輸入結束，且該行無需處理。

Output

For each integer I in the input you should print a line `The parity of B is $P(\text{mod}2)$.' , where B is the binary representation of I .

對於每筆的整數 I ，你要印一行 `The parity of B is $P(\text{mod}2)$.'，其中 B 是 I 的二進位表示法， P 是其中 1 的個數。

Sample Input

```
1
2
10
21
0
```

Sample Output

```
The parity of 1 is 1 (mod 2).
The parity of 10 is 1 (mod 2).
The parity of 1010 is 2 (mod 2).
The parity of 10101 is 3 (mod 2).
```

Analysis

用短除法把數字轉成二進位，並紀錄 1 的個數來輸出即可。

10908 Largest Square

分級：1

分類：模擬

Given a rectangular grid of characters you have to find out the length of a side of the largest square such that all the characters of the square are same and the center [intersecting point of the two diagonals] of the square is at location (r, c) . The height and width of the grid is M and N respectively. Upper left corner and lower right corner of the grid will be denoted by $(0, 0)$ and $(M - 1, N - 1)$ respectively. Consider the grid of characters given below. Given the location $(1, 2)$ the length of a side of the largest square is 3.

```
abbbabaaaaa
abbbabaaaaa
abbbabaaaaa
aaaaaaaaaaa
aaaaaaaaaaa
aaccaaaaaaa
aaccaaaaaaa
```

給你一個由字元構成的矩形以及座標 (r,c) ，你必須找出由相同字元所構成的最大正方形。 (r, c) 代表此最大正方形的中心（兩條對角線的交點）。矩形的長寬分別以 M 與 N 代表，左上角座標為 $(0, 0)$ ，右下角為 $(M - 1, N - 1)$ 。如圖所示，給你座標 $(1,2)$ ，則此最大正方形的邊長為 3。

Input

The input starts with a line containing a single integer T (< 21). This is followed by T test cases. The first line of each of them will contain three integers M , N and Q (< 21) separated by a space where M , N denotes the dimension of the grid. Next follows M lines each containing N characters. Finally, there will be Q lines each containing two integers r and c . The value of M and N will be at most 100.

輸入的第一列為一整數 T ($T < 21$)，代表測資的筆數。每組輸入的第一行包括以空白分隔的三個整數 M 、 N 和 Q ，其中 M 、 N 指矩形的大小，指接下來的 M 行包含 N 個字元。最後有 Q 行包含兩個整數 (r, c) 。 M 、 N 的值不超過 100。

Output

For each test case in the input produce $Q + 1$ lines of output. In the first line print the

value of M , N and Q in that order separated by single space. In the next Q lines, output the length of a side of the largest square in the corresponding grid for each (r, c) pair in the input.

對於每組測資輸出 $Q+1$ 行。第一行請印出以空白分隔 M 、 N 和 Q 。而接下來的 Q 行，請輸出以 (r, c) 為中心做出的最大正方形的邊長。

Sample Input

```
1
7 10 4
abbbaaaaa
abbbaaaaa
abbbaaaaa
aaaaaaaaa
aaaaaaaaa
aaccaaaaa
aaccaaaaa
1 2
2 4
4 6
5 2
```

Sample Output

```
7 10 4
3
1
5
1
```

Analysis

從點 (r, c) 往外擴張正方形，直到碰到不同的點無法繼續擴張就是答案。

10905 Children's Game

分級：2

分類：排序

There are lots of number games for children. These games are pretty easy to play but not so easy to make. We will discuss about an interesting game here. Each player will be given N positive integer. (S)He can make a big integer by appending those integers after one another. Such as if there are 4 integers as 123, 124, 56, 90 then the following integers can be made — 1231245690, 1241235690, 5612312490, 9012312456, 9056124123, etc. In fact 24 such integers can be made. But one thing is sure that 9056124123 is the largest possible integer which can be made.

You may think that it's very easy to find out the answer but will it be easy for a child who has just got the idea of number?

現在，有許多給小孩子玩的數字遊戲，這些遊戲玩起來簡單，但要創造一個就不是那麼容易的了。在這，我們將介紹一種有趣的遊戲。每位玩家將會得到 N 個正整數，你可以將一個整數接在另一個整數之後以製造一個更大的整數。例如，這有 4 個數字 123, 124, 56, 90，他們可以製造下列整數 — 1231245690, 1241235690, 5612312490, 9012312456, 9056124123....等，總共可以組合出 $24(4!)$ 種數字。但是，9056124123 是最大的那一個。

你可能會想這是個簡單的事情，但對剛有數字概念小孩來說，這會是個簡單的任務嗎？

Input

Each input starts with a positive integer N ($N \leq 50$). In next lines there are N positive integers. Input is terminated by $N = 0$, which should not be processed.

輸入第一列有一個正整數 N ($N \leq 50$)，而下一列將有 N 個正整數。當 $N = 0$ 代表輸入結束，且不須處理。

Output

For each input set, you have to print the largest possible integer which can be made by appending all the N integers.

每一筆測試資料輸出一列，你須印出利用這 N 個整數可結合成的最大整數。

Sample Input

```
4
123 124 56 90
5
123 124 56 90 9
5
9 9 9 9 9
0
```

Sample Output

```
9056124123
99056124123
99999
```

Analysis

對字串進行排序時候加點變化：對於 A、B 兩個字串，若長度相等，按照字典序排序；若長度不相等，就只有兩種組合情況，要麼 AB，要麼 BA，比較兩者的大小即可。

10814 Simplifying Fractions

分級：2

分類：大數運算

You are to write a program that reduces a fraction into its lowest terms.

請寫一個程式把一個分數化簡為最簡分數。

Input

The first line of the input file gives the number of test cases N (≤ 20). Each of the following N lines contains a fraction in the form of p / q ($1 \leq p, q \leq 10^{30}$).

輸入的第一行為測試資料的筆數 N (≤ 20)。接下來的 N 行每行有一個格式為 p / q ($1 \leq p, q \leq 10^{30}$) 的分數。

Output

For each test case, output the fraction after simplification.

相對於每筆測資，輸出化簡後的分數。

Sample Input

```
4
1 / 2
2 / 4
3 / 3
4 / 2
```

Sample Output

```
1 / 2
1 / 2
1 / 1
2 / 1
```

Analysis

利用到大整數乘法、除法和比較。除法使用試商法，即從最高位開始從 9-0 一次試乘，每位找到乘以除數後第一個小於被除數的數值即為商。

10394 Twin Primes

分級：2

分類：質數、因數與倍數

Twin primes are pairs of primes of the form $(p, p + 2)$. The term "twin prime" was coined by Paul Stckel (1892-1919). The first few twin primes are (3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43). In this problem you are asked to find out the S -th twin prime pair where S is an integer that will be given in the input.

孿生質數 (Twin Prime) 指的是，若 p 為質數，且 $p+2$ 也是質數，則我們說 $(p,p+2)$ 是一對 twin prime。這個定理是 Paul Stckel(1892-1919) 杜撰的，一開始的 twin primes 是 (3,5), (5,7), (11,13), (17,19), (29,31), (41,43)。這個問題會給定一 S 值，請你找出第 S 對 twin prime。

Input

The input will contain less than 10001 lines of input. Each line contains an integers S ($1 \leq S \leq 100000$), which is the serial number of a twin prime pair. Input file is terminated by end of file.

輸入最多含有 10001 列的測資，每列有 1 個整數 S ($1 \leq S \leq 100000$) 代表要找出的第 S 對 twin prime。輸入以 EOF 結束。

Output

For each line of input you will have to produce one line of output which contains the S -th twin prime pair. The pair is printed in the form $(p_1, \text{<space>} p_2)$. Here <space> means the space character (ASCII 32). You can safely assume that the primes in the 100000-th twin prime pair are less than 20000000.

對於每組測試資料，須輸出第 S 對 twin prime 於單行，格式為 $(p_1, \text{<space>} p_2)$ ，其中 <space> 代表空白字元 (ASCII 32)。你可以放心的假設在第 100000 對 twin prime 中的質數比 20000000 小。

Sample Input

```
1
2
3
4
```

Sample Output

(3, 5)

(5, 7)

(11, 13)

(17, 19)

Analysis

此題建立質數表查詢輸出即可。

10192 Vacation

分級：2

分類：動態規劃

You are planning to take some rest and to go out on vacation, but you really don't know which cities you should visit. So, you ask your parents for help. Your mother says "My son, you MUST visit Paris, Madrid, Lisboa and London. But it's only fun in this order." Then your father says: "Son, if you're planning to travel, go first to Paris, then to Lisboa, then to London and then, at last, go to Madrid. I know what I'm talking about."

Now you're a bit confused, as you didn't expected this situation. You're afraid that you'll hurt your mother if you follow your father's suggestion. But you're also afraid to hurt your father if you follow you mother's suggestion. But it can get worse, because you can hurt both of them if you simply ignore their suggestions!

Thus, you decide that you'll try to follow their suggestions in the better way that you can. So, you realize that the "Paris-Lisboa-London" order is the one which better satisfies both your mother and your father. Afterwards you can say that you could not visit Madrid, even though you would've liked it very much.

If your father have suggested the "London-Paris-Lisboa-Madrid" order, then you would have two orders, "Paris-Lisboa" and "Paris-Madrid", that would better satisfy both of your parent's suggestions. In this case, you could only visit 2 cities.

You want to avoid problems like this one in the future. And what if their travel suggestions were bigger? Probably you would not find the better way very easy. So, you decided to write a program to help you in this task. You'll represent each city by one character, using uppercase letters, lowercase letters, digits and the space. Thus, you can have at most 63 different cities to visit. But it's possible that you'll visit some city more than once.

If you represent Paris with "a", Madrid with "b", Lisboa with "c" and London with "d", then your mother's suggestion would be "abcd" and you father's suggestion would be "acdb" (or "dacb", in the second example).

The program will read two travel sequences and it must answer how many cities you can travel to such that you'll satisfy both of your parents and it's maximum.

你正計畫去歐洲旅行，但是你不知道該到哪些城市，所以你就向父母尋求建議。你的母親說："兒子呀！你應該去巴黎、馬德里、里斯本和倫敦，而且要按照這樣的順序去玩。"你的父親卻說："兒子如果你要去旅遊，你應該先去巴黎，然後里斯本，然後倫敦，最後才去馬德里。相信我。"

現在你陷入一個選擇的困擾中了，如果你聽從父親的建議，那會傷了母親的心。如果你聽從了母親的建議，那會傷了父親的心。但是如果你不管他們的建議，更可能傷了他們 2 個人的心。

所以你決定盡可能的聽從他們 2 人的建議。所以你決定了：巴黎，里斯本，倫敦這樣順序的旅程，以滿足你的父母親。雖然這個決定可能使你無法去馬德里。

如果你的父親建議："倫敦—巴黎—里斯本—馬德里"這樣的旅程，那麼你將有 2 組組合："巴黎—里斯本"及"巴黎—馬德里"來盡可能滿足你的父母。在這個情況下，你就只能去 2 個城市玩了。

你將來想要避免面臨上面的難題，且如果他們又建議更多的城市導致你可能難以找到更好的辦法，所以你需要寫一個程式來幫助你完成任務。你將每個城市以字元（包括大小寫英文字母，數字，以及空白字元）來表示。因此，你可以到 63 個城市可以去玩。但是請注意：你可能會到一個城市不止一次。

假如你以 a 代表巴黎，b 代表馬德里，c 代表里斯本，d 代表倫敦，那你母親建議的旅程順序就是：abcd，而你父親建議的則是：acdb（上面的第一個例子）

程式必須輸入父母所建議的 2 個旅程，然後回答在盡可能滿足你父母的情況下，你最多可以去多少個城市旅行。

Input

The input will consist on an arbitrary number of city sequence pairs. The end of input occurs when the first sequence starts with an "#"character (without the quotes). Your program should not process this case. Each travel sequence will be on a line alone and will be formed by legal characters (as defined above). All travel sequences will appear in a single line and will have at most 100 cities.

輸入含有多組測試資料，當遇到一列內容為單一"#"（不包含引號）時，代表輸入結束，也不須再處理。每一組測試資料 2 列，分別代表你父母所建議的 2 個旅程，所有的行程最多 100 個字元。

Output

For each sequence pair, you must print the following message in a line alone:

Case #d: you can visit at most K cities.

Where d stands for the test case number (starting from 1) and K is the maximum number of cities you can visit such that you'll satisfy both your father's suggestion and your mother's suggestion.

每組測試資料須先輸出一行：

Case #d: you can visit at most K cities.

其中 d 代表測資編號（從 1 開始）；K 表示在滿足你父母的情況下，你最多可以去多少個城市旅行。

Sample Input

```
abcd
acdb
abcd
dacb
#
```

Sample Output

```
Case #1: you can visit at most 3 cities.
Case #2: you can visit at most 2 cities.
```

Analysis

此題就是求最長公共子序列。利用 LCS 運算即可。

10066 The Twin Towers

分級：2

分類：動態規劃

Once upon a time, in an ancient Empire, there were two towers of dissimilar shapes in two different cities. The towers were built by putting circular tiles one upon another. Each of the tiles was of the same height and had integral radius. It is no wonder that though the two towers were of dissimilar shape, they had many tiles in common.

However, more than thousand years after they were built, the Emperor ordered his architects to remove some of the tiles from the two towers so that they have exactly the same shape and size, and at the same time remain as high as possible. The order of the tiles in the new towers must remain the same as they were in the original towers. The Emperor thought that, in this way the two towers might be able to stand as the symbol of harmony and equality between the two cities. He decided to name them the Twin Towers.

Now, about two thousand years later, you are challenged with an even simpler problem: given the descriptions of two dissimilar towers you are asked only to find out the number of tiles in the highest twin towers that can be built from them.

很久以前在一個古老師王國有兩座形狀不同的塔聳立在兩個不同的城市裡。塔是以圓形磁磚疊起來的。每塊磁磚的高度相同且半徑均為整數。因此，儘管兩座塔形狀不同，卻包含了許多相同的磁磚。

然而在建塔的一千多年後，國王命令建築師移除某些磁磚好使兩座塔的形狀大小都相同，並且要維持可能的最大高度。磁磚的順序須與原始建築相同。國王認為這樣可以象徵兩個城市的和諧與平等。他名之為「雙子星塔」。

現在，大約兩千年後，你被賦予一個更簡單的任務：給你兩座塔的描述，你只要找出可能的最大磁磚數。

Input

The input file consists of several data blocks. Each data block describes a pair of towers. The first line of a data block contains two integers N_1 and N_2 ($1 \leq N_1, N_2 \leq 100$) indicating the number of tiles respectively in the two towers. The next line contains N_1 positive integers giving the radii of the tiles (from top to bottom) in the first tower. Then follows another line containing N_2 integers giving the radii of the tiles (from top

to bottom) in the second tower.

The input file terminates with two zeros for N_1 and N_2 .

輸入有若干筆測資。每筆測資代表一對雙子星塔。每筆測資的第一行有兩個整數 N_1 及 N_2 ($1 \leq N_1, N_2 \leq 100$) 代表兩座塔的磁磚數。下一行含有 N_1 個正整數，代表第一座塔由上而下磁磚的半徑。下一行的 N_2 個整數則是第二座塔由上而下磁磚的半徑。

N_1 及 N_2 為 0 代表輸入結束。

Output

For each pair of towers in the input first output the twin tower number followed by the number of tiles (in one tower) in the highest possible twin towers that can be built from them. Print a blank line after the output of each data set.

對於每一對雙子星塔，輸出它的編號（從 1 開始）及每座塔所能保留的最大可能磁磚數。測資間請空一行。

Sample Input

```
7 6
20 15 10 15 25 20 15
15 25 10 20 15 20
8 9
10 20 20 10 20 10 20 10
20 10 20 10 10 20 10 10 20
0 0
```

Sample Output

```
Twin Towers #1
Number of Tiles : 4
Twin Towers #2
Number of Tiles : 6
```

Analysis

此題就是求最長公共子序列。利用 LCS 運算填表來做即可。

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
```

Analysis

根據題意計算即可。須小心題目說明。

10023 Square root

分級：2

分類：大數運算

You are to determinate X by given Y , from expression $X = \sqrt{Y}$

給你一個整數 Y，請你求出根號 Y 的值。（ $X = \sqrt{Y}$ ）

Input

The first line is the number of test cases, followed by a blank line. Each test case of the input contains a positive integer Y ($1 \leq Y \leq 10^{1000}$), with no blanks or leading zeroes in it. It is guaranteed, that for given Y, X will be always an integer. Each test case will be separated by a single line.

第 1 列有一個數字代表接下來有幾列的測試資料。每筆測資只有一個正整數 Y ($1 \leq Y \leq 10^{1000}$)，沒有任何其他字元。我們保證 Y 一定是某一個正整數的平方。

Output

For each test case, your program should print X in the same format as Y was given in input. Print a blank line between the outputs for two consecutive test cases.

對於每個測資，你的程式須印出根號後的值。測資間需隔一空白行。

Sample Input

```
1
7206604678144
```

Sample Output

```
2684512
```

Analysis

Y 最大值為 10^{1000} ，須先理解中國直式開方的方法，便可手動開方。

10004 Bicoloring

分級：2

分類：模擬

In 1976 the "Four Color Map Theorem" was proven with the assistance of a computer. This theorem states that every map can be colored using only four colors, in such a way that no region is colored using the same color as a neighbor region.

Here you are asked to solve a simpler similar problem. You have to decide whether a given arbitrary connected graph can be bicolored. That is, if one can assign colors (from a palette of two) to the nodes in such a way that no two adjacent nodes have the same color. To simplify the problem you can assume:

- no node will have an edge to itself.
- the graph is nondirected. That is, if a node a is said to be connected to a node b ,

then you must assume that b is connected to a .

- the graph will be strongly connected. That is, there will be at least one path from any node to any other node.

1976 年，在電腦協助之下證明了 4 色地圖理論（Four Color Map Theorem）。就是僅以 4 種顏色在地圖上不同的區域塗色，使得相鄰的區域顏色均不相同。現在，你要解決一個類似，但比較簡單的問題。給你一個相連的圖，請你在節點上塗色（只有 2 種不同的顏色），並且回答是否可以使得相鄰的節點顏色均不相同。為了使問題簡單一些，你可以假設：

- 沒有節點會有連向自己的邊。
- 邊是沒有方向性的，也就是說如果節點 A 可以連到節點 B ，那麼代表節點 B 也可以連到節點 A 。
- 圖形是強連通的，也就是說任 2 節點之間皆有路徑相連。

Input

The input consists of several test cases. Each test case starts with a line containing the number n ($1 < n < 200$) of different nodes. The second line contains the number of edges m . After this, m lines will follow, each containing two numbers that specify an edge between the two nodes that they represent. A node in the graph will be labeled using a number a ($0 \leq a < n$).

An input with $n = 0$ will mark the end of the input and is not to be processed.

輸入含有多組測試資料。每組測試資料的第一列有一個正整數 n ($1 < n < 200$) 代表節點的數目。第二列有一個正整數 m ，代表邊的數目。接下來的 m 列每列有 2 個整數代表邊所連接的 2 個節點的代號，每個節點的代號將使用一個數字 a ($0 \leq a < n$) 來代表。

當 $n=0$ 指輸入結束，且不需處理。

Output

You have to decide whether the input graph can be bicolored or not, and print it as shown below.

對每一組測試資料輸出是否可以用 2 種顏色塗節點使得相鄰的節點顏色均不相同。如 Sample Output 所示，若可以請輸出：BICOLORABLE.，否則輸出：NOT BICOLORABLE.。

Sample Input

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

Sample Output

NOT BICOLORABLE.

BICOLORABLE.

BICOLORABLE.

Analysis

可從任一點出發進行 DFS 搜尋。先假定出發點的顏色之後，搜尋到的下一個點必須填入與上一個點不同的顏色，同時要確認與現在上色的這個點相鄰的所有點是否都是還未上色、或是正好有不同的顏色。假如上色後搜尋與其相鄰的點有相同的顏色，則必為 NOT BICOLORABLE. 可直接中斷 DFS，假如所有的點都被搜尋結束，則為 BICOLORABLE.。

846 Steps

分級：2

分類：模擬

One steps through integer points of the straight line. The length of a step must be nonnegative and can be by one bigger than, equal to, or by one smaller than the length of the previous step.

What is the minimum number of steps in order to get from x to y ? The length of the first and the last step must be 1.

一個人沿著一數線前進。他每次走的長度（整數）必須是正的，而且比上一步走的長度多 1，一樣，或少 1。

請問這個人若要從 x 走到 y ，最少需要走幾步？須注意第一步及最後一步的長度一定是 1。

Input

Input consists of a line containing n , the number of test cases. For each test case, a line follows with two integers: $0 \leq x \leq y < 2^{31}$.

輸入的第一列有一個正整數代表以下有幾組測試資料。每組測試資料一列，含有 2 個整數 x 與 y ($0 \leq x \leq y < 2^{31}$)。

Output

For each test case, print a line giving the minimum number of steps to get from x to y .

每組測試資料輸出一列，最少需要走幾步才能從 x 走到 y 。

Sample Input

```
2
1000000
1996
5
8
123456
```

Sample Output

1

3626619

5957

5

11

426566

Analysis

本題僅求最少的步數，因此過程如何安排就不重要，只要找出步數與距離的關係即可得解。

725 Division

分級：2

分類：數學計算

Write a program that finds and displays all pairs of 5-digit numbers that between them use the digits 0 through 9 once each, such that the first number divided by the second is equal to an integer N , where $2 \leq N \leq 79$. That is,

$$abcde / fghij = N$$

where each letter represents a different digit. The first digit of one of the numerals is allowed to be zero.

如果把數字 0 到 9 分配成 2 個整數（各五位數），現在請你寫一支程式找出所有的配對使得第一個數可以整除第二個數，而且商為 N （ $2 \leq N \leq 79$ ），也就是：

$$abcde / fghijk = N$$

這裡每個英文字母代表不同的數字，第一個數字可以為 0。

Input

Each line of the input file consists of a valid integer N . An input of zero is to terminate the program.

輸入包含許多筆待測資料，每列代表一筆待測資料，每筆待測資料包含一個正整數 N ， N 為 0 時代表輸入結束。

Output

Your program have to display ALL qualifying pairs of numerals, sorted by increasing numerator (and, of course, denominator).

Your output should be in the following general form:

$$xxxxx / xxxxx = N$$

$$xxxxx / xxxxx = N$$

In case there are no pairs of numerals satisfying the condition, you must write ``There are no solutions for N .". Separate the output for two different values of N by a blank

line.

對每筆待測資料由小到大輸出每一對符合條件的數。你的輸出須符合下面格式：

xxxxx / xxxxx = N

xxxxx / xxxxx = N

如果找不到符合條件的數對，則輸出 `There are no solutions for N.`。每筆測試資料間請空一列。

Sample Input

61

62

0

Sample Output

There are no solutions for 61.

79546 / 01283 = 62

94736 / 01528 = 62

Analysis

此題很簡單，以暴力枚舉來判斷即可。

612 DNA Sorting

分級：2

分類：排序

One measure of "unsortedness" in a sequence is the number of pairs of entries that are out of order with respect to each other. For instance, in the letter sequence "DAABEC", this measure is 5, since D is greater than four letters to its right and E is greater than one letter to its right. This measure is called the number of inversions in the sequence. The sequence "AACEDGG" has only one inversion (E and D)--it is nearly sorted--while the sequence "ZWQM" has 6 inversions (it is as unsorted as can be--exactly the reverse of sorted).

You are responsible for cataloguing a sequence of DNA strings (sequences containing only the four letters A, C, G, and T). However, you want to catalog them, not in alphabetical order, but rather in order of "sortedness", from "most sorted" to "least sorted". All the strings are of the same length.

在一個字串中，「未排序」的程度是以各字元間彼此的大小關係來計算的。例如在字串 DAABEC 中，「未排序」的程度為 5，因為 D 比它右邊的 4 個字元大，E 比它右邊的 1 個字元大。而字串 AACEDGG 「未排序」的程度為 1（幾乎是快排序好的，唯一的未排序發生在 E 和 D 之間），字串 ZWQM 「未排序」的程度為 6（剛好是完全排序的相反）。

現在你的任務是為許多的 DNA 字串來做排序。每個字串中僅含有 A,C,G 和 T 這 4 種字元。但是你不按字母順序排序，而是根據各字串「未排序」的程度，由小到大輸出。在這裡每個字串的長度均相同。

Input

The first line of the input is an integer M, then a blank line followed by M datasets. There is a blank line between datasets.

The first line of each dataset contains two integers: a positive integer n ($0 < n \leq 50$) giving the length of the strings; and a positive integer m ($0 < m \leq 100$) giving the number of strings. These are followed by m lines, each containing a string of length n.

輸入的第一列有一個整數 M 代表以下有幾組測試資料。各組測試資料須隔空白列再輸入。

資料的第一列含有 2 個正整數 n ($0 < n \leq 50$) 和 m ($0 < m \leq 100$)， n 代表字串的長度， m 代表字串的數目。所以接下來的 m 列，每列有一個長度為 n 的字串。

Output

For each dataset, output the list of input strings, arranged from "most sorted" to "least sorted". If two or more strings are equally sorted, list them in the same order they are in the input file.

Print a blank line between consecutive test cases.

對每組測試資料按照「未排序」的程度，由小到大輸出各字串。假如有不只 2 個字串「未排序」的程度相同，則按照它們在輸入中出現的順序輸出。

各組測試資料之間請輸出一空白列。

Sample Input

```
1

10 6
AACATGAAGG
TTTTGGCCAA
TTTGGCCAAA
GATCAGATTT
CCCGGGGGGA
ATCGATGCAT
```

Sample Output

```
CCCGGGGGGA
AACATGAAGG
GATCAGATTT
ATCGATGCAT
TTTTGGCCAA
TTTGGCCAAA
```

Analysis

根據題意找出各列未排序程度作答即可。

572 Oil Deposits

分級：2

分類：動態規劃

The GeoSurvComp geologic survey company is responsible for detecting underground oil deposits. GeoSurvComp works with one large rectangular region of land at a time, and creates a grid that divides the land into numerous square plots. It then analyzes each plot separately, using sensing equipment to determine whether or not the plot contains oil.

A plot containing oil is called a pocket. If two pockets are adjacent, then they are part of the same oil deposit. Oil deposits can be quite large and may contain numerous pockets. Your job is to determine how many different oil deposits are contained in a grid.

有家石油公司 GeoSurvComp 負責探勘地底下的石油含量，長期在某一塊長矩行的土地上工作，並且為了探勘的方便將土地切割為許多小塊，然後使用儀器對每個小塊去探勘。

含有石油的小塊稱為一個 pocket。假如兩個 pocket 相連，則這兩個 pocket 屬於同一個 oil deposit。oil deposit 可以是相當大的土地和很多的 pockets，你的任務就是要找出這塊地包含幾個不同的 oil deposit。

Input

The input file contains one or more grids. Each grid begins with a line containing m and n , the number of rows and columns in the grid, separated by a single space. If $m = 0$ it signals the end of the input; otherwise $1 \leq m \leq 10$ and $1 \leq n \leq 100$. Following this are m lines of n characters each (not counting the end-of-line characters). Each character corresponds to one plot, and is either `*`, representing the absence of oil, or `@`, representing an oil pocket.

輸入包含好幾組資料，每組資料的第一行有 2 個整數 m, n 。 m 代表這塊地的列數， n 代表這塊地的行數。 ($1 \leq m, n \leq 100$)，接下來的 m 行就是這塊地探勘的內容。 '@' 代表此小塊含石油， '*' 代表此小塊不含石油。 $m = n = 0$ 代表輸入結束。

Output

For each grid, output the number of distinct oil deposits. Two different pockets are part of the same oil deposit if they are adjacent horizontally, vertically, or diagonally. An oil deposit will not contain more than 100 pockets.

對每組測試資料輸出 oil deposit 的數目。如果是相鄰的兩個油區,可視為同一塊油田的一部分。而除了上下左右鄰居視為相鄰以外,對角線相差一格也視為相鄰。一塊油田不超過 100 個 pockets。

Sample Input

```
1 1
*

3 5
*@*@*
**@**
*@*@*

1 8
@@****@*

5 5
****@
*@@*@
*@**@
@@@*@
@@**@

0 0
```

Sample Output

```
0
1
2
2
```

Analysis

DFS 用遞迴實作,遇到油田就換成*字號,每一個 DFS 結束後計數即可。

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:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2} \text{ for all } i \geq 2$$

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：

$$F_0 = 0$$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2} \text{ for all } i \geq 2$$

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 ($0 \leq n \leq 5000$)，代表要求的第幾個 Fibonacci 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

Analysis

因是大數加法要利用字串來處理，並將每一項記錄下來即可得解。

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.

Shssssssssh ... I am hunting wabbits. Heh Heh Heh Heh ...

Sample Output

2

7

10

9

Analysis

將輸入以陣列儲存，再來統計並輸出即可。

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,
.
"

Analysis

將輸入以二維陣列儲存，小心輸出細節便可印出。

406 Prime Cuts

分級：1

分類：質數、因數與倍數

A prime number is a counting number (1, 2, 3...) that is evenly divisible only by 1 and itself. In this problem you are to write a program that will cut some number of prime numbers from the list of prime numbers between (and including) 1 and N. Your program will read in a number N; determine the list of prime numbers between 1 and N; and print the $C*2$ prime numbers from the center of the list if there are an even number of prime numbers or $(C*2)-1$ prime numbers from the center of the list if there are an odd number of prime numbers in the list.

一個質數的定義為：除了 1 和它本身之外，沒有別的數可以整除它的。在本題中會給你 N 及 C，請你寫一程式找出 1 到 N 中所有的質數，如果總個數是偶數，請輸出中間那 $C*2$ 個質數。如果總個數是奇數，則輸出中間那 $(C*2)-1$ 個質數。

Input

Each input set will be on a line by itself and will consist of 2 numbers. The first number ($1 \leq N \leq 1000$) is the maximum number in the complete list of prime numbers between 1 and N. The second number ($1 \leq C \leq N$) defines the $C*2$ prime numbers to be printed from the center of the list if the length of the list is even; or the $(C*2)-1$ numbers to be printed from the center of the list if the length of the list is odd.

每組測試資料一列，各含有 2 個整數。第一個數 N ($1 \leq N \leq 1000$) 代表須找出質數的範圍 1~N。第二個數 C ($1 \leq C \leq N$)，代表若總個數為偶數，則有 $2C$ 個質數會被印出；若為奇數，則印出 $2C-1$ 個質數。

Output

For each input set, you should print the number N beginning in column 1 followed by a space, then by the number C, then by a colon (:), and then by the center numbers from the list of prime numbers as defined above. If the size of the center list exceeds the limits of the list of prime numbers between 1 and N, the list of prime numbers between 1 and N (inclusive) should be printed. Each number from the center of the list should be preceded by exactly one blank. Each line of output should be followed by a blank line. Hence, your output should follow the exact format shown in the sample output.

對每組測試資料須先輸出"N C: "，接著輸出題目要求的質數。如果要求量大於總質數量，則全部印出。或者要求質數包含 1 與 N 也須印出。每個質數前方有一空格，每組測試資料後亦空一列。詳細請參考 Sample Output。

Sample Input

21 2

18 2

18 18

100 7

Sample Output

21 2: 5 7 11

18 2: 3 5 7 11

18 18: 1 2 3 5 7 11 13 17

100 7: 13 17 19 23 29 31 37 41 43 47 53 59 61 67

Analysis

建立質數表查詢即可。注意在此題中 1 也為質數，須將 1 也加入。

401 Palindromes

分級：2

分類：模擬

A regular palindrome is a string of numbers or letters that is the same forward as backward. For example, the string "ABCDEDCBA" is a palindrome because it is the same when the string is read from left to right as when the string is read from right to left.

A mirrored string is a string for which when each of the elements of the string is changed to its reverse (if it has a reverse) and the string is read backwards the result is the same as the original string. For example, the string "3AIAE" is a mirrored string because "A" and "I" are their own reverses, and "3" and "E" are each other's reverses.

A mirrored palindrome is a string that meets the criteria of a regular palindrome and the criteria of a mirrored string. The string "ATOYOTA" is a mirrored palindrome because if the string is read backwards, the string is the same as the original and because if each of the characters is replaced by its reverse and the result is read backwards, the result is the same as the original string. Of course, "A", "T", "O", and "Y" are all their own reverses.

A list of all valid characters and their reverses is as follows.

Character	Reverse	Character	Reverse	Character	Reverse
A	A	M	M	Y	Y
B		N		Z	5
C		O	0	1	1
D		P		2	S
E	3	Q		3	E
F		R		4	
G		S	2	5	Z
H	H	T	T	6	
I	I	U	U	7	
J	L	V	V	8	8
K		W	W	9	
L	J	X	X		

Note that 0(zero) and O (the letter) are considered the same character and therefore ONLY the letter " O " is a valid character.

一般回文(a regular palindrome)是一個由數字或字母組成的字串，且正反向會相同的。例如，字串“ABCDEDCBA”是回文，因為它從左至右和從右至左讀出是

相同的。

一個鏡像字串(a mirrored string)指的是其字母都會變為反向(如果它有一個反向),且字串被讀取的結果與原始字串是相同的。例如,字串“3AIAE”是一個鏡像字串,因為“A”和“I”反向還是自己,而“3”和“E”互為反向。

而鏡像回文(mirrored palindrome)是要同時符合回文與鏡像字串的標準。字串“ATOYOTA”就是一個鏡像回文,因為如果其字串以反向讀取,且如果每個字元被替換成鏡像,其結果還是和原始的字串相同。當然,“A”、“T”、“O”和“Y”的反向都是自己。

所有有效字元與其反向如表顯示。

注意:0(數字零)與O(字母O)被視為相同的字元,所以只有字母O才是有效字元。

Input

Input consists of strings (one per line) each of which will consist of one to twenty valid characters. There will be no invalid characters in any of the strings. Your program should read to the end of file.

輸入資料有多列字串,每一列字串長度介於1~20之間,且必定都是有效字元(如上表),絕不會有非有效字元出現。你的程式必須讀到檔案結尾後中止。

Output

For each input string, you should print the string starting in column 1 immediately followed by exactly one of the following strings.

" -- is not a palindrome." : if the string is not a palindrome and is not a mirrored string

" -- is a regular palindrome." : if the string is a palindrome and is not a mirrored string

" -- is a mirrored string." : if the string is not a palindrome and is a mirrored string

" -- is a mirrored palindrome." : if the string is a palindrome and is a mirrored string

Note that the output line is to include the '-'s and spacing exactly as shown in the table above and demonstrated in the Sample Output below.

In addition, after each output line, you must print an empty line.

對每一筆輸入字串，你必須先印出該字串後，再接著印出下列字串中的其中一個。

" -- is not a palindrome."：假如輸入字串既非迴文也非鏡像。

" -- is a regular palindrome."：假如輸入字串是迴文但非鏡像。

" -- is a mirrored string."：假如輸入字串非迴文但為鏡像。

" -- is a mirrored palindrome."：假如輸入字串既為迴文亦為鏡像。

須注意字元 - (dash)與空白字元是必須的，如上說明與下面 Sample Output 所示。

另外，每一列輸出之後你必須另外再印出一列空行。

Sample Input

NOTAPALINDROME

ISAPALINILAPASI

2A3MEAS

ATOYOTA

Sample Output

NOTAPALINDROME -- is not a palindrome.

ISAPALINILAPASI -- is a regular palindrome.

2A3MEAS -- is a mirrored string.

ATOYOTA -- is a mirrored palindrome.

Analysis

照著題意解，將鏡像的部份存成陣列： $Mir[] = \{AAE3HHIIJLLJ\dots\}$ ，再以 $Mir[i]$ 和 $Mir[i+1]$ 是否符合，來比對兩個字元是否為鏡像。

397 Equation Elation

分級：2

分類：模擬

The author of an elementary school algebra text book has approached you to write a program to solve simple algebra equations. The author wants to use a program to avoid human errors in preparing the solutions manual. The text book author will provide a text file of the simple problems for your problem to solve. All of the problems will be in the form of an algebraic equality. The specific syntax of the problems will be an algebraic statement consisting of integer constants and the four basic arithmetic operators, an equal sign, and a variable. For example:

$$12 - 4 * 3 = x$$

For the solutions manual the problem is not just to be solved, but solved one step at a time. For the above input line, the corresponding output would be:

$$12 - 4 * 3 = x$$

$$12 - 12 = x$$

$$0 = x$$

The simple problems your program is to solve are limited to integer values with multiplication, division, addition and subtraction operations. Note that, as in the above example, the computation must follow the standard order of precedence for arithmetic operations. All multiplications and divisions are performed, from left to right, before any additions and subtractions, and then all additions and subtractions are performed from left to right. You may assume that all divisions will result in integer values.

小學數學教科書的作者請你幫忙寫一個程式處理簡單的代數運算，希望利用程式來避免課本的編寫錯誤。他將會提供簡單的問題文本，你需要處理的是代數恆等式，等式的形式為整數的四則運算後接上一個等號與一個變數，如下所示：

$$12 - 4 * 3 = x$$

不只要解決這問題，而是要一個步驟一個步驟地處理等式，其輸出如下：

$$12 - 4 * 3 = x$$

$$12 - 12 = x$$

$$0 = x$$

你的程式處理的問題僅限於整數的四則運算（乘法，除法，加法和減法運算），且需遵守先乘除後加減的法則，由左至右依序處理每一項，你可以假設所有除法結果必為整數。

Input

The input file will consist of several equations to be solved. Each equation will contain from 1 to 20 operations with 2 to 21 integer operands (there will, of course, always be one more operand than operators). Integer operators in the input may have a leading sign (i.e. may be preceded by a unary operator). Spaces in the input line are optional; that is, spaces may be present between operators and operands, or they may not. The variable names will consist of 1 to 8 alphabetic characters.

輸入的每一列代表一個等式，等式包含 1~20 個運算符號，且會有 2~21 個運算元（運算元的個數必然比運算符多一個），運算元前面可能會有正負號，等式中可能包含空白字元，也就是說空白可能會出現在運算符號與運算元間。變數名稱長度介於 1~8 個英文字母之間。

Output

Output for a problem should begin with the problem to be solved, then followed by one line of output after each operation. The spacing between the numbers and operations in the output is not critical. Having the correct answers and all the correct steps in the output is important.

A typical input file will consist of multiple algebraic problems, each on a separate line. The output for each input problem should be separated by a single blank line in the output. The end of the file marks the end of the input.

對每組測試資料先輸出其原來的等式再接著其計算的過程，輸出格式中的空白字元可有可無，正確的運算過程及答案才是最重要的。

對於每個輸入問題，請在每組輸出之間以一系列空行隔開。輸入的結尾代表輸出的結束。

Sample Input

$$3 * 4 + 4 - 1 / 1 = xyzzy$$

$$12 + 2 * 12 / 2 - 1 = y$$

$$2 * -3 + -6 - +4 = r$$

$$2*-3+-6-+4=r$$

Sample Output

$$3 * 4 + 4 - 1 / 1 = xyzzy$$

$$12 + 4 - 1 / 1 = xyzzy$$

$$12 + 4 - 1 = xyzzy$$

$$16 - 1 = xyzzy$$

$$15 = xyzzy$$

$$12 + 2 * 12 / 2 - 1 = y$$

$$12 + 24 / 2 - 1 = y$$

$$12 + 12 - 1 = y$$

$$24 - 1 = y$$

$$23 = y$$

$$2 * -3 + -6 - 4 = r$$

$$-6 + -6 - 4 = r$$

$$-12 - 4 = r$$

$$-16 = r$$

$$2 * -3 + -6 - 4 = r$$

$$-6 + -6 - 4 = r$$

$$-12 - 4 = r$$

$$-16 = r$$

Analysis

宣告兩陣列分別紀錄整數與運算操作。根據題意印出步驟，先找出*與/來計算，再找+與-運算。

382 Perfection

分級：1

分類：數學計算

From the article Number Theory in the 1994 Microsoft Encarta: "If a, b, c are integers such that $a = bc$, a is called a multiple of b or of c , and b or c is called a divisor or factor of a . If c is not ± 1 , b is called a proper divisor of a . Even integers, which include 0, are multiples of 2, for example, -4, 0, 2, 10; an odd integer is an integer that is not even, for example, -5, 1, 3, 9. A perfect number is a positive integer that is equal to the sum of all its positive, proper divisors; for example, 6, which equals $1 + 2 + 3$, and 28, which equals $1 + 2 + 4 + 7 + 14$, are perfect numbers. A positive number that is not perfect is imperfect and is deficient or abundant according to whether the sum of its positive, proper divisors is smaller or larger than the number itself. Thus, 9, with proper divisors 1, 3, is deficient; 12, with proper divisors 1, 2, 3, 4, 6, is abundant."

Given a number, determine if it is perfect, abundant, or deficient.

1994 年微軟(Microsoft)工程師的 Encarta 提出一篇“數論(Number Theory)”文章：

「假設 a, b, c 為三整數，且 $a = bc$ 成立， a 被稱為 b 或 c 的倍數，而 b 或 c 被稱為 a 的除數或因數。假設 c 不為正 1 或負 1，則 b 可被稱為“ a 的恰當因數(proper divisor of a)”。包含零的偶數為 2 的倍數，例如：-4, 0, 2, 10...等，奇數為非偶數的整數，例如：-5, 1, 3, 9...等。一個“完全數(perfect number)”為一正整數，它等於它的所有正的恰當因數(proper divisor)的和，例如 6 即為完全數，它等於 $1 + 2 + 3$ ，而 $28 = 1 + 2 + 4 + 7 + 14$ 亦為完全數。一正整數若非完全數，則稱它為“缺陷數(deficient)”或“充裕數(abundant)”，端看該數的所有正的恰當因數的和是否小於或大於該數本身。因此，9 的恰當因數為 1, 3，故 9 為“缺陷數”，12 為“充裕數”因為它的恰當因數等於 1, 2, 3, 4, 6」

給定一數，請判斷它為完全數、缺陷數或充裕數。

Input

A list of N positive integers (none greater than 60,000), with $1 < N < 100$. A 0 will mark the end of the list.

有 N 個正整數一列(每一個整數的大小不超過 60,000)，且 $1 < N < 100$ 。 $N = 0$ 代表輸入結束。

Output

The first line of output should read **PERFECTION OUTPUT**. The next N lines of output should list for each input integer whether it is perfect, deficient, or abundant, as shown in the example below. Format counts: the echoed integers should be right justified within the first 5 spaces of the output line, followed by two blank spaces, followed by the description of the integer. The final line of output should read **END OF OUTPUT**.

第一列必須輸出 **PERFECTION OUTPUT**，接下來的 N 列用來表示對應的輸入值為完全數、缺陷數或充裕數，如下面例子中的輸出所示。數值的格式為 5 個字元寬且靠右對齊，再接兩個空白字元，並以該數的類型作為該列的結束，最後一列必須為 **END OF OUTPUT**。

Sample Input

```
15 28 6 56 60000 22 496 0
```

Sample Output

```
PERFECTION OUTPUT
  15  DEFICIENT
  28  PERFECT
   6  PERFECT
  56  ABUNDANT
60000 ABUNDANT
  22  DEFICIENT
  496 PERFECT
END OF OUTPUT
```

Analysis

可以利用質因數分解的公式來求出因數總和，在減掉自己本身數值，便可以比較來解答。

374 Big Mod

分級：2

分類：大數運算

Calculate

$$R = B^P \bmod M$$

for large values of B, P, and M using an efficient algorithm. (That's right, this problem has a time dependency — 3.000 seconds!!!)

計算 $R = B^P \bmod M$

對 B、P、M 為相當大的數值，請寫一個有效率的演算法來。（沒錯！此題有規定時間 3.000 seconds！！）

Input

Three integer values (in the order B, P, M) will be read one number per line. B and P are integers in the range 0 to 2147483647 inclusive. M is an integer in the range 1 to 46340 inclusive.

每筆測試資料有 3 行，各有 1 個整數分別代表 B、P、M。其中 $0 \leq B, P \leq 2147483647$ ，且 $1 \leq M \leq 46340$ 。

Output

The result of the computation. A single integer.

輸出計算的結果，每筆測試資料一行。

Sample Input

3

18132

17

17

1765

3

2374859

3029382

36123

Sample Output

13

2

Analysis

這題須先了解餘數運算的分配律規則，也就是 $(B^P) \% M = (B^{(P/2)} \% M) * (B^{(P/2)} \% M)$ ，便可對 $B^P \bmod N$ 來做遞迴運算。

305 Joseph

分級：1

分類：模擬

The Joseph's problem is notoriously known. For those who are not familiar with the original problem: from among n people, numbered $1, 2, \dots, n$, 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 us the message about the incident. For example when $n = 6$ and $m = 5$ then the people will be executed in the order $5, 4, 6, 2, 3$ and 1 will be saved.

Suppose that there are k good guys and k bad guys. In the circle the first k are good guys and the last k bad guys. You have to determine such minimal m that all the bad guys will be executed before the first good guy.

Joseph's problem 是個惡名昭彰的故事：有 n 個俘虜（編號從 $1, 2, 3, \dots, n$ ）圍成一圈，然後每隔 m 人開始點名，點到就要被殺掉。按照此規則，最後會只剩下一個人可以保留性命。Joseph 是個很聰明的人，總是能挑到最後存留的位置，所以他才能披露出這件事來。例如： $n=6, m=5$ 則被殺掉的人的編號依序是 $5, 4, 6, 2, 3$ ，最後只有編號 1 活了下來。

現在假設共有 k 個好人和 k 個壞人，其中排在編號 1 到 k 的是好人，排在編號 $k+1$ 到 $2k$ 的是壞人。你的任務就是要找出一個最小的 m ，使得在所有 k 個壞人被殺之前，沒有一個好人會被殺掉。

Input

The input file consists of separate lines containing k . The last line in the input file contains 0 . You can suppose that $0 < k < 14$.

輸入每行有一個整數 k ($0 < k < 14$)，若 $k = 0$ 代表輸入結束。

Output

The output file will consist of separate lines containing m corresponding to k in the input file.

每筆測資輸出一行。根據輸入的 k 來印出 m 值。

Sample Input

3

4

0

Sample Output

5

30

Analysis

這題基本上有公式解，可以利用迴圈去做，或者也可以用暴力法（使用 `Array`）照著題目做。

294 Divisors

分級：2

分類：數學計算

Mathematicians love all sorts of odd properties of numbers. For instance, they consider 945 to be an interesting number, since it is the first odd number for which the sum of its divisors is larger than the number itself.

To help them search for interesting numbers, you are to write a program that scans a range of numbers and determines the number that has the largest number of divisors in the range. Unfortunately, the size of the numbers, and the size of the range is such that a too simple-minded approach may take too much time to run. So make sure that your algorithm is clever enough to cope with the largest possible range in just a few seconds.

數學家們愛著奇數的各種屬性。例如，他們認為 945 是一個有趣的數字，因為它所有因數總和值大於它質因數總和（因數總和=質因數總和+1）。

而為了幫助他們尋找有趣的數字，請你寫一個程式，使它可以輸入一範圍的數字，且找出在範圍中具有最多因數的數字。但不幸的是，數的大小很大，範圍也不小，所以你的程式必須有效率，否則可能無法在幾秒內跑完。

Input

The first line of input specifies the number N of ranges, and each of the N following lines contains a range, consisting of a lower bound L and an upper bound U , where L and U are included in the range. L and U are chosen such that $1 \leq L \leq U \leq 1000000000$ and $0 \leq U - L \leq 10000$.

輸入的第一列有一個正整數 N ，代表以下有幾組測試資料。每組測試資料一列，含有代表範圍下限的 L 與上限的 U ，且 $1 \leq L \leq U \leq 1000000000$ ， $0 \leq U - L \leq 10000$ 。

Output

For each range, find the number P which has the largest number of divisors (if several numbers tie for first place, select the lowest), and the number of positive divisors D of P (where P is included as a divisor).

Print the text 'Between L and H , P has a maximum of D divisors.', where L , H , P , and

D are the numbers as defined above.

對每一組測試資料，找出在範圍內有最多除數的數 P（如果有不止一個數有最多除數，請找最小的那個），以及它有多少個除數 D。

輸出格式：'Between L and H, P has a maximum of D divisors.'。其中 L、H、P 與 D 依上面敘述來定義。

Sample Input

```
3
1 10
1000 1000
999999900 1000000000
```

Sample Output

```
Between 1 and 10, 6 has a maximum of 4 divisors.
Between 1000 and 1000, 1000 has a maximum of 16 divisors.
Between 999999900 and 1000000000, 999999924 has a maximum of 192 divisors.
```

Analysis

這題的關鍵就在於是否理解題目質因數的定理：

1. 一個數的質因數一定小於等於這個數的開根值
2. 一個數的因數總和值等於其各質因數總和值+1

264 Count on Cantor

分級：1

分類：模擬

One of the famous proofs of modern mathematics is Georg Cantor's demonstration that the set of rational numbers is enumerable. The proof works by using an explicit enumeration of rational numbers as shown in the diagram below.

```
1/1  1/2  1/3  1/4  1/5 ...
2/1  2/2  2/3  2/4
3/1  3/2  3/3
4/1  4/2
5/1
```

In the above diagram, the first term is 1/1, the second term is 1/2, the third term is 2/1, the fourth term is 3/1, the fifth term is 2/2, and so on.

現代數學中有一個有名的證明（由 Georg Cantor 所提出的）：有理數是可數的。他使用一個圖表明確列舉出有理數，如下所示：

```
1/1  1/2  1/3  1/4  1/5 ...
2/1  2/2  2/3  2/4
3/1  3/2  3/3
4/1  4/2
5/1
```

在此中，第一項是 1/1，第 2 項是 1/2，第三項是 2/1，第四項是 3/1，第五項是 2/2，以下依此類推。

Input

You are to write a program that will read a list of numbers in the range from 1 to 10^7 . The input list contains a single number per line and will be terminated by end-of-file.

請你寫一個程式，每筆資料輸入 1 行，含有 1 個正整數 n ($1 \leq n \leq 10^7$)。EOF 表示輸入結束。

Output

Print for each number the corresponding term in Cantor's enumeration as given below.

No blank line should appear after the last number.

對每筆測資，請輸出 Cantor's 列舉中的第 n 項。最後的值輸出後不該有空行出現。

Sample Input

3
14
7

Sample Output

TERM 3 IS 2/1
TERM 14 IS 2/4
TERM 7 IS 1/4

Analysis

找出第幾斜排以及第幾斜排的最末項還有第 n 項的分母與分子這四者的關係，即可得解。

256 Quirksome Squares

分級：1

分類：數學計算

The number 3025 has a remarkable quirk: if you split its decimal representation in two strings of equal length (30 and 25) and square the sum of the numbers so obtained, you obtain the original number:

$$(30+25)^2=3025$$

The problem is to determine all numbers with this property having a given even number of digits.

For example, 4-digit numbers run from 0000 to 9999. Note that leading zeroes should be taken into account. This means that 0001 which is equal to $(00+01)^2$ is a quirksome number of 4 digits. The number of digits may be 2,4,6 or 8. Although maxint is only 32767 and numbers of eight digits are asked for, a well-versed programmer can keep his numbers in the range of the integers. However efficiency should be given a thought.

3025 這個數相當特別。如果你把他從中間切開你會得到 2 個長度相同的數 30 和 25，且此 2 數和的平方恰等於原來的數，我們稱這種數為 quirksome number。

$$(30+25)^2=3025$$

在這個問題中，定義所有的位數值都為偶數。

例如說，4 位數就是從 0000 到 9999（數字前方的 0 也要算在內），也就是說 0001 等於 $(00+01)^2$ ，是一個 4 位數的 quirksome number。由於位數可能是 2, 4, 6, 8，即使最大整數範圍到 32767，而題目可到 8 位元，但是身為一位優秀的程式設計師是可以將給予數字儲存在整數範圍內的。不過效率要好好思考就是了。

Input

The input of your program is a text file containing numbers of digits (taken from 2,4,6,8), each number on a line of its own.

輸入可能包含了好幾列測試資料，每一列只有一個數字 N（N=2,4,6,8），代表要求的是幾位數的 quirksome number。

Output

The output is a text file consisting of lines containing the quirky numbers (ordered according to the input numbers and for each input number in increasing order).

對每一個輸入 N ，你應該要輸出所有 N 位數的 quirky number，每個一行，由小到大排列。（注意：數字前方的 0 不可忽略，例如 2 位數中的 quirky number 01 不可只輸出 1。）

Sample Input

```
2
2
```

Sample Output

```
00
01
81
00
01
81
```

Analysis

此題按照題意來窮舉即可，主要注意輸出格式的規定。

245 Uncompress

分級：2

分類：字元與字串

A simple scheme for creating a compressed version of a text file can be used for files which contain no digit characters. The compression scheme requires making a list of the words in the uncompressed file. When a non-alphabetic character is encountered in the uncompressed file, it is copied directly into the compressed file. When a word is encountered in the uncompressed file, it is copied directly into the compressed file only if this is the first occurrence of the word. In that case, the word is put at the front of the list. If it is not the first occurrence, the word is not copied to the compressed file. Instead, its position in the list is copied into the compressed file and the word is moved to the front of the list. The numbering of list positions begins at 1.

Write a program that takes a compressed file as input and generates a reproduction of the original uncompressed file as output.

對於不含數字的文章有一種簡單的壓縮法，而這方法需用一個串列（list）來記錄曾經出現過的文字（word）。當遇到非英文字母的字元，該字元直接複製到壓縮後的檔案；遇到一個字時，如果這一個字是第一次出現，除了把這個字複製到壓縮後的檔案之外，並把他加到串列的開頭。如果這一個字不是第一次出現，則這個字不會複製到壓縮後的檔案，而是把這個字在串列中的位置複製到壓縮後的檔案，並且在串列中把這個字移到串列的開頭。串列的開頭位置為 1。

現在你的任務是給你一篇用上述方法壓縮後的文章，請你把他還原回來並輸出。

Input

You can assume that no word contains more than 50 characters and that the original uncompressed file contains no digit characters.

For the purposes of this problem, a word is defined to be a maximal sequence of upper- and lower-case letters. Words are case-sensitive - the word abc is not the same as the word Abc. For example,

x-ray contains 2 words : x and ray

Mary's contains 2 words : Mary and s

It's a winner contains 4 words : It and s and a and winner

There is no upper limit on the number of different words in the input file. The end of the input file is signified by the number 0 on a line by itself. The terminating 0 merely indicates the end of the input and should not be part of the output produced by your program.

你可以假設所有的字都不會超過 50 個字元，並且未壓縮的文章不含有數字。

另外在這個問題中，字的定義為連續的英文字母 (A~Z, a~z)，且字有分大小寫，因此 abc 和 Abc 是不同的 2 個字。例如：

x-ray 包含了 2 個字：x 和 ray

Mary's 包含了 2 個字：Mary 和 s

It's a winner 包含了 4 個字：It、s、a 和 winner

輸入只有一組測試資料，資料內文字數沒有上限。輸入最後一列僅含有一個 0，代表輸入結束（此列無輸出）。

Output

See the sample output below for details of the output format.

對於輸入的每一列，輸出解壓縮後的文章。

Sample Input

```
Dear Sally,  
    Please, please do it--1 would 4  
Mary very, 1 much.    And 4 6  
8 everything in 5's power to make  
14 pay off for you.  
    -- Thank 2 18 18--  
0
```

Sample Output

```
Dear Sally,  
    Please, please do it--it would please  
Mary very, very much.    And Mary would  
do everything in Mary's power to make  
it pay off for you.  
    -- Thank you very much--
```

Analysis

這題重點在於須將輸入分成字母、數字與非字母三種 Case，再搭配 linked list 結構來處理。

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
```

Analysis

以 DFS 生成重複排列，排序出現順序是先大寫再小寫。也可先對字串做 sort，再利用 C++ 的 STL 中的 next_permutation() 即可得解。

151 Power Crisis

分級：2

分類：模擬

During the power crisis in New Zealand this winter (caused by a shortage of rain and hence low levels in the hydro dams), a contingency scheme was developed to turn off the power to areas of the country in a systematic, totally fair, manner. The country was divided up into N regions (Auckland was region number 1, and Wellington number 13). A number, m , would be picked 'at random', and the power would first be turned off in region 1 (clearly the fairest starting point) and then in every m 'th region after that, wrapping around to 1 after N , and ignoring regions already turned off. For example, if $N = 17$ and $m = 5$, power would be turned off to the regions in the order: 1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7.

The problem is that it is clearly fairest to turn off Wellington last (after all, that is where the Electricity headquarters are), so for a given N , the 'random' number m needs to be carefully chosen so that region 13 is the last region selected.

Write a program that will read in the number of regions and then determine the smallest number m that will ensure that Wellington (region 13) can function while the rest of the country is blacked out.

能源危機在今年冬天發生於紐西蘭的能源危機事件中(由於缺水導致水壩因低水位無法發電所引起)，便發展出了一套緊急配套計畫來有系統且完全公正地決定國內哪個地區要被斷電。這個國家分成 N 個區域(奧克蘭是 1 號區域，威靈頓是第 13 號區域)。首先他們會「隨機」選擇一個數 m ，接著從第 1 號區域(這是最公平的起始地點了)開始斷電，接著每往後數 m 個區域(忽略掉已經斷電過的區域，超過 N 就從頭繼續數)，被點到的最後一個區域就會成為下一個被斷電的區域。例如說，如果 $N = 17$ 並且 $m = 5$ ，那麼依序被斷電的區域是：1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7.

問題是，最公平的情況下，威靈頓應該要是最後一個被斷電的區域(畢竟那是電力供應廠的所在地)，所以每一個「隨機」選擇的數字 m 應當被小心地選擇好讓區域 13 是最後一個被選上的斷電區域。

編寫一個程式，可以讀入區域的數量並且決定最小的數字 m 使得威靈頓(區域 13)在剩下的區域都黯淡無光以前仍然在運作。

Input

Input will consist of a series of lines, each line containing the number of regions (N) with $13 \leq N < 100$. The file will be terminated by a line consisting of a single 0.

輸入含有多組測試資料。每組測資包含一個數字 N ($13 \leq N < 100$) 於一列。當 $N = 0$ 代表輸入結束。

Output

Output will consist of a series of lines, one for each line of the input. Each line will consist of the number m according to the above scheme.

對每組測試資料輸出一列，須根據題目說明印出數字 m 。

Sample Input

```
17
0
```

Sample Output

```
7
```

Analysis

本題可以利用窮舉法，輸入一個 N ，然後定義一個長度是 $N+1$ 的陣列 x ，讓 m 值從 1 開始迴圈。先把陣列 x 全部值設為 0，然後從 $x[1]$ 開始值設為 1 表示該地區已經斷電，再往後數 m 個不等於 1 的地區，同設值為 1。直到數了 N 個地區之後，說明全部都斷電了，那麼判斷最後一個斷電的地區是不是 13。如果是，則找到了 m ，如果不是的話，就 $m++$ 再重複上面的步驟。

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: $1 \times 20c$, $2 \times 10c$, $10c + 2 \times 5c$, and $4 \times 5c$.

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
------	-----

Analysis

利用兩個整數方式解決浮點數的問題，並注意會有重複的組合，所以先選最小的幣值(5c)加入，再選次小的幣值(10c)，以此類推。先建立好 dp 表，我們不可能用小數當索引值，所以用乘上 100 的數來做索引即為答案。此題亦可使用遞迴解得。

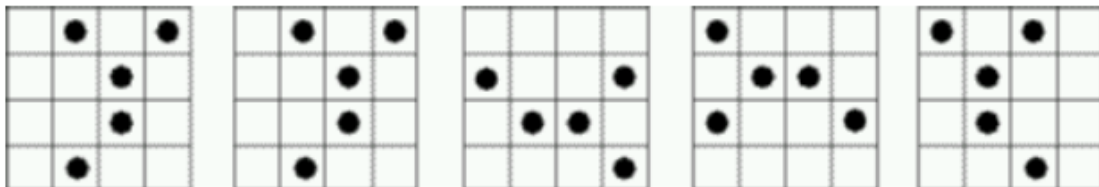
141 The Spot Game

分級：2

分類：模擬

The game of Spot is played on an $N \times N$ board as shown below for $N = 4$. During the game, alternate players may either place a black counter (spot) in an empty square or remove one from the board, thus producing a variety of patterns. If a board pattern (or its rotation by 90 degrees or 180 degrees) is repeated during a game, the player producing that pattern loses and the other player wins. The game terminates in a draw after $2N$ moves if no duplicate pattern is produced before then.

Consider the following patterns:



If the first pattern had been produced earlier, then any of the following three patterns (plus one other not shown) would terminate the game, whereas the last one would not.

放石頭遊戲(The game of Spot)在一塊 $N \times N$ 的板子上進行，如下圖為 $N=4$ 的板子。遊戲的玩法是兩個玩家輪流放一塊石頭在空的格子上，或是可以從板子上拿一塊石頭起來，遊戲的進行中可以發現，板子上 石頭 的佈局會不斷變化，當一玩家排出已重複出現過的佈局時，他就算輸了這一局(一種佈局如果將之旋轉 90 度、180 度、270 度亦視為相同的佈局)。若在 $2N$ 步內未出現過相同的佈局就算和局。

請參考圖片的幾種佈局：

若出現過第一種佈局，則再出現 2、3、4 種佈局即結束比賽(還有另一種能結束比賽的佈局未畫出)，注意，第 5 種佈局並不能算是相同的佈局。

Input

Input will consist of a series of games, each consisting of the size of the board, N ($2 \leq N \leq 50$) followed, on separate lines, by $2N$ moves, whether they are all necessary or not. Each move will consist of the coordinates of a square (integers in the range $1..N$) followed by a blank and a character '+' or '-' indicating the addition or removal of a

spot respectively. You may assume that all moves are legal, that is there will never be an attempt to place a spot on an occupied square, nor to remove a non-existent spot. Input will be terminated by a zero (0).

輸入會有多組測試資料，一開始會給定板子的大小 N ($2 \leq N \leq 50$)，接下來會有 $2N$ 個移步方式，當然也有可能 $2N$ 步還沒走完就有人贏得了比賽。每一列會有一個座標位置，並以 + 或 - 來表示新增或移除一塊石頭。你可以假定所有的步驟都是合法的，也就是說，不會在空格子上拿走一塊石頭，也不會重複放置石頭在同一個位置上。輸入的最後會以 $N=0$ 做結束。

Output

Output will consist of one line for each game indicating which player won and on which move, or that the game ended in a draw.

請輸出哪位玩家贏得了比賽，並在哪一步贏得比賽，若平手則輸出 draw。

Sample Input

```
2
1 1 +
2 2 +
2 2 -
1 2 +
2
1 1 +
2 2 +
1 2 +
2 2 -
0
```

Sample Output

```
Player 2 wins on move 3
Draw
```

Analysis

注意棋盤是 $N \times N$ 的，不是 4×4 的。題目主要判斷每次移動的局面是否出現過，所以只要能夠找到向右旋轉 90 度後的狀態和初始狀態的關係，然後每得到一個狀態就去判斷，如果初始狀態和由它旋轉而來的狀態都不再 set 裡面，那麼就把這些狀態插入 set。

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>.

Analysis

第 n 個數必定是從之前 n-1 個數中的其中一值 x_2 或 x_3 或 x_5 而來的，因此要找第 n 個數(N[n])，就把前 n-1 個數 x_2, x_3, x_5 ，找出大於(N[n-1])的最小值便是解(N[n])。

118 Mutant Flatworld Explorers

分級：1

分類：模擬

Given the dimensions of a rectangular grid and a sequence of robot positions and instructions, you are to write a program that determines for each sequence of robot positions and instructions the final position of the robot.

A robot position consists of a grid coordinate (a pair of integers: x-coordinate followed by y-coordinate) and an orientation (N, S, E, W for north, south, east, and west). A robot instruction is a string of the letters 'L', 'R', and 'F' which represent, respectively, the instructions:

Left: the robot turns left 90 degrees and remains on the current grid point.

Right: the robot turns right 90 degrees and remains on the current grid point.

Forward: the robot moves forward one grid point in the direction of the current orientation and maintains the same orientation.

The direction North corresponds to the direction from grid point (x, y) to grid point $(x, y+1)$.

Since the grid is rectangular and bounded, a robot that moves "off" an edge of the grid is lost forever. However, lost robots leave a robot "scent" that prohibits future robots from dropping off the world at the same grid point. The scent is left at the last grid position the robot occupied before disappearing over the edge. An instruction to move "off" the world from a grid point from which a robot has been previously lost is simply ignored by the current robot.

給你一塊矩形土地的長寬，再依序給定每個機器人的初始位置狀況及一連串的指令集，你必須用你的程式求出每個機器人最後的位置狀況。

一個機器人的位置狀況包括了其坐標（ x 坐標， y 坐標），和它面向的方向（用 N, S, E, W 來分別代表北、南、東、西）。至於一個機器人所收到的指令集，是一個由字母 'L'，'R'，和 'F' 所構成的字串，其分別代表：

左轉（Left）：機器人在原地往左轉 90 度。

右轉（Right）：機器人在原地往右轉 90 度。

前進（Forward）：機器人往其面向的方向向前走一格，且不改變其面向之方向。從坐標 (x, y) 走至 $(x, y+1)$ 的這個方向我們定義為北方。

因為此矩形土地是有邊界的，所以一旦一個機器人走出邊界掉落下去，就相當於永遠消失了。不過這個掉下去的機器人會留下「標記 (scent)」，提醒以後的機器人，避免他們從同一個地方掉下去。掉下去的機器人會把標記，留在他掉落之前所在的最後一個坐標點。所以對於以後的機器人，當他正位在有標記的地方時，這個機器人就會忽略會讓他掉下去的指令。

Input

The first line of input is the upper-right coordinates of the rectangular world, the lower-left coordinates are assumed to be 0, 0.

The remaining input consists of a sequence of robot positions and instructions (two lines per robot). A position consists of two integers specifying the initial coordinates of the robot and an orientation (N, S, E, W), all separated by white space on one line. A robot instruction is a string of the letters 'L', 'R', and 'F' on one line.

Each robot is processed sequentially, i.e., finishes executing the robot instructions before the next robot begins execution.

Input is terminated by end-of-file.

You may assume that all initial robot positions are within the bounds of the specified grid. The maximum value for any coordinate is 50. All instruction strings will be less than 100 characters in length.

輸入裡的第一列有 2 個正整數，代表這個矩形世界右上角頂點的坐標，其中假設這個世界的左下角頂點坐標為 (0 , 0)。

接下來是若干組有關機器人的初始位置狀況和指令集，每個機器人 2 列。第一列為位置狀況，包括了用空白分隔的兩個整數和一個字元 (N , S , E 或 W)，代表機器人初始的位置坐標以及機器人最初所面對的方向。第二列則是指令集，是一個由 'L' , 'R' 和 'F' 所組成的字串。

各機器人是依序動作的，也就是說，直到一個機器人作完他全部的動作，下一個機器人才會開始動作。

輸入以 eof 結束。

可以假設所有機器人的初始位置是在邊界內的。邊界坐標的最大值為 50，而所有的指令集字串，長度將不超過 100 個字元。

Output

For each robot position/instruction in the input, the output should indicate the final grid position and orientation of the robot. If a robot falls off the edge of the grid the word "LOST" should be printed after the position and orientation.

對於每一個機器人，你都必須輸出其最後所在的坐標和面對的方向。如果一個機器人將會掉落出此界外，你必須先輸出他在掉落前，最後的所在位置和面對的方向，再多加一個字： LOST 。

Sample Input

```
5 3
1 1 E
RFRFRFRF
3 2 N
FRRFLLFFRRFLL
0 3 W
LLFFFLFLFL
```

Sample Output

```
1 1 E
3 3 N LOST
2 3 S
```

Analysis

注意題目敘述中的「掉下去的機器人會把標記，留在他掉落之前所在的最後一個坐標點」這句，已經有記號的座標，機器人會忽略指令而不會出界。之後按照題目敘述便可輸出答案。

105 The Skyline Problem

分級：2

分類：模擬

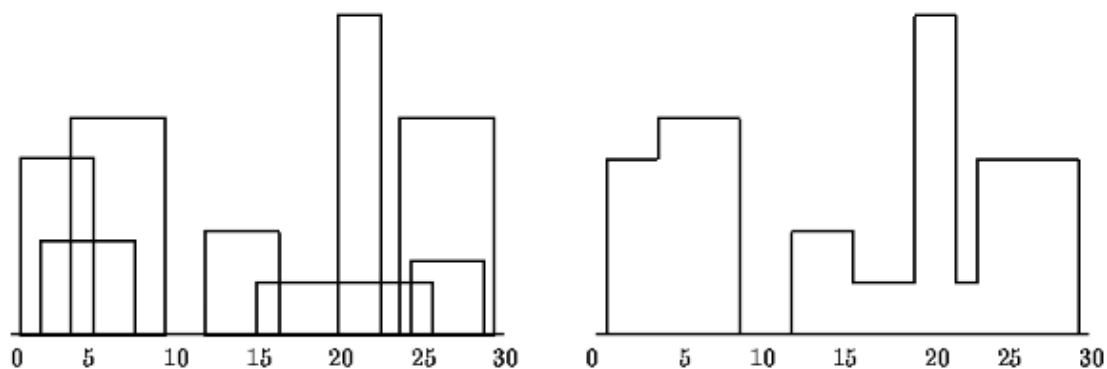
With the advent of high speed graphics workstations, CAD (computer-aided design) and other areas (CAM, VLSI design) have made increasingly effective use of computers. One of the problems with drawing images is the elimination of hidden lines—lines obscured by other parts of a drawing.

You are to design a program to assist an architect in drawing the skyline of a city given the locations of the buildings in the city. To make the problem tractable, all buildings are rectangular in shape and they share a common bottom (the city they are built in is very flat). The city is also viewed as two-dimensional. A building is specified by an ordered triple (L_i, H_i, R_i) where L_i and R_i are left and right coordinates, respectively, of building i and H_i is the height of the building. In the diagram below buildings are shown on the left with triples

$(1, 11, 5), (2, 6, 7), (3, 13, 9), (12, 7, 16), (14, 3, 25), (19, 18, 22), (23, 13, 29), (24, 4, 28)$

the skyline, shown on the right, is represented by the sequence:

$(1, 11, 3, 13, 9, 0, 12, 7, 16, 3, 19, 18, 22, 3, 23, 13, 29, 0)$



隨著高速繪圖電腦工作站的出現，CAD (computer-aided design) 和其他領域 (CAM, VLSI 設計) 都充分使用這些電腦的長處。但其中有一個繪製問題就是消除隱藏線 (繪圖中被其他部分遮擋的線條)。

而在本問題中，你必須幫助建築師，根據他所提供給你都市中建築物的位置，你得幫他找出這些建築物的空中輪廓 (skyline)。為了使問題容易處理一些，所有的建築物都是矩形的，並且都建築在同一個平面上 (城市是建置在非常平坦的地面)。你可以把這城市看成一個二度平面空間。每一棟建築物都以 (L_i, H_i, R_i) 這樣的序列來表示。其中 L_i 和 R_i 分別是該建築物左邊和右邊的位置， H_i 則是建築

物的高度。如左方圖就是(1,11,5), (2,6,7), (3,13,9), (12,7,16), (14,3,25), (19,18,22), (23,13,29), (24,4,28)這八棟建築物的位置圖。而你的任務就是畫出這些建築物所構成的輪廓，並且以(1, 11, 3, 13, 9, 0, 12, 7, 16, 3, 19, 18, 22, 3, 23, 13, 29, 0)這樣的序列來表示如右方圖的輪廓。

Input

The input is a sequence of building triples. All coordinates of buildings are integers less than 10,000 and there will be at least one and at most 5,000 buildings in the input file. Each building triple is on a line by itself in the input file. All integers in a triple are separated by one or more spaces. The triples will be sorted by L_i , the left x-coordinate of the building, so the building with the smallest left x-coordinate is first in the input file.

輸入只有一組測試資料。每列有一棟建築物的資料（以空白分隔的三個整數），其所有坐標是小於 10,000 的整數且至少會有一個。所有建築物不會超過 5000 棟，並且建築物已按照 L_i 排好序。

Output

The output should consist of the vector that describes the skyline as shown in the example above. In the skyline vector $(V_1, V_2, V_3, \dots, V_{n-2}, V_{n-1}, V_n)$, the V_i such that i is an even number represent a horizontal line (height). The V_i such that i is an odd number represent a vertical line (x-coordinate). The skyline vector should represent the "path" taken, for example, by a bug starting at the minimum x-coordinate and traveling horizontally and vertically over all the lines that define the skyline. Thus the last entry in all skyline vectors will be a 0.

輸出為描述建築物輪廓的向量。在輪廓向量 $(V_1, V_2, V_3, \dots, V_{n-1}, V_n)$ 中，在 i 為奇數的情形下， V_i 表示一條垂直線（x 座標），在 i 為偶數的情形下， V_i 表示一條水平線（高度）。天際線輪廓向量就像一隻蟲從最左邊建築物走起，沿著輪廓路徑水平及垂直的行走的路徑，所以輪廓的最後一個數一定為 0。

Sample Input

```
1 11 5
2 6 7
3 13 9
12 7 16
14 3 25
19 18 22
```

23 13 29

24 4 28

Sample Output

1 11 3 13 9 0 12 7 16 3 19 18 22 3 23 13 29 0

Analysis

開一個巨大陣列，儲存各位置的最高點。要注意左邊界及右邊界的狀況，之後每加入一棟房子，就將該房子範圍的高度更新。