2013 網際網路程式設計全國大賽
國中組初賽

- 題目：本次比賽共 六 題（含本封面共 15 頁）。
- 題目輸入：全部題目的輸入都來自**標準輸入**。
- 題目輸出：全部題目的輸出皆輸出到螢幕 (**標準輸出**)。
  輸出和裁判的答案必須完全一致，英文大小寫不同或多餘空白換行字元皆視為錯誤答案。
- 時間限制：每一題的執行時間限制如下表所示。
  其間執行的電腦上不會有別的動作、也不會使用鍵盤或滑鼠。
- 比賽中上傳之程式碼請依照以下規則命名：

  1. 若使用 C 做為比賽語言則命名為 `pa.c`, `pb.c`, 以此類推。
  2. 若使用 C++ 做為比賽語言則命名為 `pa.cpp`, `pb.cpp`，以此類推。

  未按照此規則命名之程式碼將可能因此得到 Compilation Error。
- `long long` 型別的整數使用方式請參考下一頁。
- `cin` 輸入經測試發現速度遠慢於 `scanf` 輸入，答題者若使用需自行承擔因輸入速度過慢導致 Time Limit Exceeded 的風險。

表 1: 題目資訊

	題目名稱	執行時間限制
題目 **A**	挑食的大胃王	1 秒
題目 **B**	餅乾，餅乾，更多的餅乾	1 秒
題目 **C**	混色模式	10 秒
題目 **D**	跑者的修煉	1 秒
題目 **E**	數位文獻修復	5 秒
題目 **F**	時光機實驗	1 秒

2013 網際網路程式設計全國大賽
解題程式輸入輸出範例

C 程式範例：

```c
#include <stdio.h>
int main(void)
{
    int cases, i;
    long long a, b;
    scanf("%d", &cases);
    for(i = 0;i < cases;i++)
    {
        scanf("%I64d %I64d", &a, &b);
        printf("%I64d\n", a + b);
    }
    return 0;
}
```

C++ 程式範例：

```cpp
#include <iostream>
int main(void)
{
    int cases;
    std::cin >> cases;
    for(int i = 0; i < cases; ++i)
    {
        long long a, b;
        std::cin >> a >> b;
        std::cout << a + b << std::endl;
    }
    return 0;
}
```

題目 A
挑食的大胃王

執行時間限制: 1 秒

松板老師為陵櫻國小三年四班的老師，陵櫻國小原本是由中央廚房供應營養午餐給小朋友們食用，但是很不幸的，最近大地震把中央廚房震壞了，因此在廚房修復之前，松板老師必須每天訂便當讓小朋友有午餐可以吃。

訂便當並不是一件簡單的工作，因為小朋友們都非常的挑食，對於不吃的食物，小朋友會將其連同底下的部分白飯一起挖掉，如果白飯吃的不夠多，小朋友會吃不飽以至於無法專心上課，學習效果不佳，因此訂便當的首要目標就是找到可以讓大家都吃得飽的便當！

松板老師決定以班上最挑食的同時也是食量最大的大胃王阿力當作買便當的標準，如果最挑食的阿力都可以吃飽了，那麼全班也都可以吃飽。老師記錄了阿力所有不吃的食物，當她選購便當的時候，會先看看便當的表面有哪些菜，試算出阿力會吃的白飯量。

為了方便起見，老師的計算方法是先將便當切成以 1 cm × 1 cm 為單位的小格子，每個格子不是阿力會吃就是阿力不會吃。對於阿力不吃的菜，老師假設阿力會挖掉的區域為不吃的食物所存在的格子與其上下左右相連的格子，對於所有會被挖掉的區域，阿力會挖掉 3 **cm** 厚的白飯。示意圖如下，下圖是一個便當的表面，0 代表是阿力會吃的食物，1 代表是阿力不吃會挖掉的食物，灰色區域就是阿力所有會挖掉的區域，總共是 18 cm^2 ，因此阿力總共會挖掉 $18 \times 3 = 54$ cm^3 的白飯。

0	0	0	0	0
0	1	1	0	0
0	0	0	1	1
0	0	0	0	1
0	0	1	0	0

阿力可以食用的白飯量即為白飯原本的體積減掉被挖掉的白飯體積。

因為市面上的便當琳瑯滿目，因此松板老師想請你幫忙寫個程式，協助計算各種便當小朋友可食用的白飯量是多少。

■ 輸入說明

輸入的第一行有一個正整數 $T(T \leq 100)$，代表測試資料的組數。

每一組測試資料的第一行有三個正整數 M, N, K $(3 \leq M, N, K \leq 50)$ 分別以空白隔開。M 代表便當的長度，N 代表便當的寬度，K 代表便當的高度。

接下來會有 M 行，每行有 N 個數字，分別以一個空白隔開，表示便當表面每一格是否是阿力會吃的食物。數字由 0 與 1 組成，0 代表是阿力會吃的食物，1 代表是阿力不吃會挖掉的食物。

■ 輸出說明

對於每一筆測試資料請輸出一列，表示阿力可以食用的白飯量。

■ 範例輸入

```
2
9 10 5
0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 1
5 5 3
0 0 0 0 0
0 1 1 0 0
0 0 0 1 1
0 0 0 0 1
0 0 1 0 0
```

■ 範例輸出

```
336
21
```

題目 B
餅乾，餅乾，更多的餅乾

執行時間限制: 1 秒

庫奇從小就是一位非常愛吃餅乾的人，他的夢想就是可以生活在一個充滿餅乾的地方。為了讓夢想得以實現，庫奇很努力地學習、研究各種可以實現夢想的方法，最後他決定開設餅乾工廠讓自己一邊可以身處餅乾世界，一邊藉此維生！

在庫奇賣力地打工賺取足夠的資金後，他終於買下一塊土地並建造了工廠的外殼！但是光有外殼是不夠的，庫奇還需要有設備來生產餅乾才行。

庫奇詢問了許多販賣設備的商人們，希望可以用划算的價格買到所需的設備；然而擁有設備的設備商人們也希望可以賺取最多的金錢，不肯以划算的價格出售。幸好在庫奇與設備商人談判協商後，他們達到了共識！

他們訂出了以下的購買設備規則：

1. 購買第一台設備的價錢是一塊餅乾，第二台設備的價錢是兩塊餅乾，第三台設備的價錢是三塊餅乾 …… 以此類推，第 n 台設備需要用 n 塊餅乾購買。
2. 購買一台新設備時，新設備需花一秒的時間裝設，裝設期間，所有舊有設備皆無法使用，以免發現危險。
3. 所有設備在運作期間都是一秒鐘生產一塊餅乾。
4. 庫奇的工廠在一開始時會有一台免費贈送的設備，若要添購新設備則依規則 1.

(很顯然的，當庫奇擁有越多台的設備，庫奇一秒鐘可以獲得的餅乾數就會愈多)

現在庫奇要開始營運工廠了！庫奇想要知道從現在只有一台設備、隨時可以購買新設備的情況下，N 秒鐘最多可以擁有多少的餅乾？(購用新設備時支付給商人的餅乾是屬於商人的，非庫奇的)

由於庫奇不擅長於計算，無法自行算出答案，請問你幫幫庫奇嗎？

■ 輸入說明

輸入的第一行有一個正整數 $T(T \leq 100)$，代表測試資料的組數。

每一組測試資料有一個正整數 $N(N < 10000)$ 代表經過的時間，單位為秒。

■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示 N 秒後庫奇最多可以擁有的餅乾數量。

■ 範例輸入

```
2
4
10
```

■ 範例輸出

```
4
18
```

■ 範例說明

1. 第一筆測試資料，4 秒時有兩種情況可以獲得最多的餅乾：
   (a) 使用一開始的機器生產 4 秒鐘後產生 4 塊餅乾
   (b) 一開始的機器在第 1 秒時生產 1 塊餅乾，第 2 秒時添購 1 台新設備，2 台設備在第 3, 4 秒分別生產 2 塊餅乾，最後庫奇獲得 4 塊餅乾
2. 第二筆測試資料，10 秒時有一種情況可以獲得最多的餅乾：第 1 秒時生產 1 塊餅乾，第二秒時添購 1 台新設備，第 3 秒時生產 2 塊餅乾，第 4 秒時添購 1 台新設備，利用此 3 台機器，於剩下的 6 秒生產 18 塊餅乾。

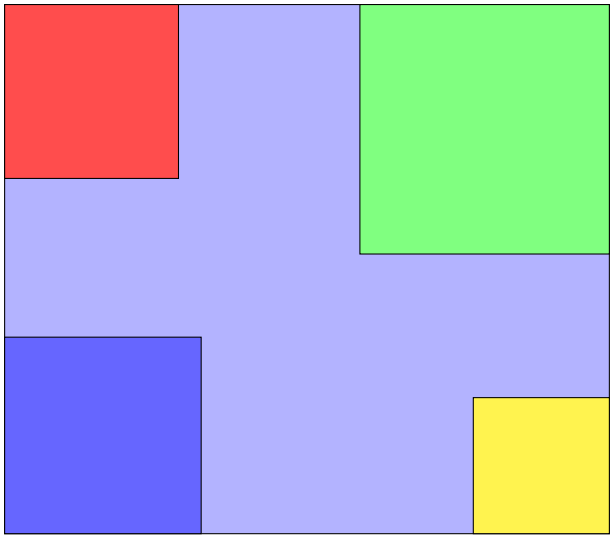
題目 C
混色模式

執行時間限制: 10 秒

在專業的製圖軟體當中，當兩張圖重疊在一起的時候，總會有許多模式可供選擇，以製造各式絢麗色彩的合成效果。

大可魚是一個對於美學極為講究的人，他常常為了要怎麼將幾張圖完美地合成出最好看的圖而大傷腦筋。就以最簡單的 Logo 來說吧，要如何合成出最好看的 Logo 簡直是耗時費神的工作。

現在大可魚打算做出一個史上最強 Logo：它是一個長 N 公分、寬 M 公分的長方形空間，四個角落分別畫上邊長為 a, b, c, d 公分的正方形，其中 a, b, c, d 都是**正整數**。



「現在這個配色好像 $\cdots$」大可魚的好朋友老蚯如是說。

總之中可魚很喜歡這個配色。為了避免矩形的顏色重疊時，將導致 XOR 混色模式開啟——畫面會變得亂七八糟——原本的配色也會跟著跑掉。新的 Logo 四個角落的任兩個正方形**內部**都不能重疊 (正方形的邊或角碰到的話沒有關係)。

請你幫忙算算，在長寬固定的情形下，大可魚究竟有幾種選擇呢？

■ 輸入說明

輸入的第一行有一個正整數 $T(T \leq 1000)$，代表測試資料的組數。

每一組測試資料只佔一行，包含兩個正整數：N 以及 M $(2 \leq N, M \leq 80000)$。

■ 輸出說明

對於每一筆測試資料請輸出一行，表示總共有幾種方法。請注意這個方法數可能會超過 2^{31}，但是不會超過 2^{63}。

■ 範例輸入

```
4
2 2
3 3
7 10
80000 80000
```

■ 範例輸出

```
1
5
421
51198720016000000000
```

題目 D
跑者的修練

執行時間限制: 1 秒

在一次偶然的機會下，直子看到了電視轉播的馬拉松比賽，電視上的馬拉松選手們個個專注在比賽上，盡全力地向前奔跑，展現無比的強度與速度。在電視機前的直子深深地被選手們的力與美所感動，下定決心，在將來的某一天也要成為一個很棒的馬拉松選手，帶給他人一樣的感動。

直子知道成就並不是隨便可以達成的，在成就來臨的那一天前，她必須慢慢地、一點一滴琢磨自己，努力練習！由於還是學生，也沒有什麼時間可以到其他地方練習，所以直子選擇以學校的操場當作練習場地。

直子的體育老師為直子訂定了跑步計畫，目標是每天都要慢跑操場，第一天時會跑一圈，之後以每天增加一圈的方式來練習。除了單純的跑步之外，在跑步時觀察自己的身體是很重要的，因此教練希望當直子要跑比 M 圈還多的距離的話，每跑 M 圈必須快走一圈用以調整呼吸與評估自己的身體狀態再繼續跑，讓自己可以維持最好的狀態！

直子十分好奇在開始執行跑步計畫 N 天之後，總共會繞幾圈操場呢？所謂的繞操場包含了跑步與快走都算，也就是說，假設今天直子是第七天且教練規定跑超過五圈需要先快走一圈，那麼她在跑完五圈之後，需要先快走一圈，再跑剩下的兩圈，才能達成教練的規定。因此總共是繞了八圈操場！

請問你可以告訴直子從第一天到第 N 天總共繞了幾圈操場嗎？

■ 輸入說明

輸入的第一行有一個正整數 T ($T \leq 100$)，代表測試資料的組數。

每一組測試資料有兩個正整數 M, N 以空白隔開，M 代表跑多少圈後需要快走，N 代表跑了幾天。($M, N \leq 10^9$)

■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示直子總共繞了幾圈操場。

注意: 答案可能大於 **int** 的數值範圍。

■ 範例輸入

```
3
5 5
5 13
10 50
```

■ 範例輸出

```
15
102
1375
```

■ 範例說明

1. 第一筆測試資料，直子繞操場的圈數如下

第幾天	1	2	3	4	5
圈數	1	2	3	4	5

第五天時，因為沒有要跑超過五圈，所以不需要快走一圈，將第一天至第五天加起來為 15。

2. 第二筆測試資料，直子繞操場的圈數如下

第幾天	1	2	3	4	5	6	7	8	9	10	11	12	13
圈數	1	2	3	4	5	7	8	9	10	11	13	14	15

將第一天至第十三天加起來為 102。

題目 E
數位文獻修復

執行時間限制: 5 秒

對於較有歷史、古老的典籍，要妥善地保存是相當困難的。尤其是古代書寫的紙張，不僅本身可能會破損、風化，書寫於其上的文字也可能再經過數千年的時間後淡化。在這種情形之下，其實是很難供眾人查閱內容的，畢竟光是要保存都需花費很大的功夫了。如果任意翻閱，可能會破壞到典籍本身。

故現代掀起一陣數位文獻典藏的潮流，對於一些以前的文物、文獻，我們可以用將其數位化的方式保存。對於一些古人的手抄文本等書籍，如果我們有辦法將內容輸入至電腦、資料庫中保存的話，就不用擔心有朝一日會再也讀不出內文了。

然而，有些典籍文本經過歲月的摧殘，上頭有些字跡早已模糊不清，或者是有所破損。對於這類的文字，有許多是我們很難做修補的。

有一天，曉涵在圖書館查閱相關資料時發現，有些內容有所缺漏，而數位文本竟然有超過一個來源。重要的是，從不同來源數位文本，有所殘缺的地方還不一定相同！聰明的曉涵於是便想到，「如果兩份理應相同的文本能夠互相填補缺失，那不是太棒了?」於是，曉涵找出了所有有兩種以上來源的相同文本，並嘗試想要修補它們。

然而，有的文本不管在哪個來源的電子化內容中，有些位置總是都缺失了，那我們就沒有辦法將其內容經由交互比對來完全復原。

為了簡化題目，我們將每份文獻資料當作一個字串，並且假設給定的文獻資料都只有兩個不同的來源。請你寫一個程式幫幫曉涵，判斷每一組給定的文獻有沒有辦法藉由相互比較完全復原。

■ 輸入檔說明

輸入的第一行有一個正整數 $T(T \leq 50)$，代表測試資料的組數。

每一筆測試資料有兩行，分別對應到兩個來源的文獻內容。每一篇文獻都是以一個長度不超過 1024 的字串表示。其中，對於破損的字元，在輸入中將以問號 “?”(不含雙引號) 字元表示。

所有輸入的文獻內容保證除了表示破損的 “?’’ 之外，只會出現大寫英文字母及小寫英文字母。並且，輸入保證兩個給定的字串皆至少有一個 “?’’ 字元。兩個字串除了 “?’’ 以外的內容將完全相同。

■ 輸出檔說明

對於每一筆測試資料，請輸出一行。

如果曉涵有辦法成功完全修復整份文獻資料的話，請輸出 “Yes:”(不含雙引號)，之後緊接著是修復完全的文獻內容；反之，沒有辦法的話，請輸出一行 “No”。

(注意：請不要在任何地方自行加上空白或其他字元。)

■ 範例輸入

```
5
De?rMyFrien?
DearM??riend
????????????sAndAlgorithms
DataStructuresAnd?????????s
?ffect
?ffect
ant?diluvian
ant?diluv?an
NPS?
N?SC
```

■ 範例輸出

```
Yes:DearMyFriend
Yes:DataStructuresAndAlgorithms
No
No
Yes:NPSC
```

題目 F
時光機實驗

執行時間限制: 1 秒

幾千萬年前，天上掉下了一顆隕石，在地球表面上引發了浩劫。

你是個科學家，想要研究這個傳說中的隕石坑。隕石坑周圍的紋路十分詭異，如同蚯蚓身上的環節，故又稱蚯蚓坑。

你發現當初這個隕石所挾帶的巨大的重力在撞擊時扭曲了周圍的時空。至今在隕石坑附近仍會不時產生時空狹縫，如果你穿過他，就可以穿越時空。由於蚯蚓是一種蟲，所以這種時空狹縫又叫蟲洞。

可惜蟲洞非常寫小且只有特製的儀器才能平安穿越，否則一般的物體都會被奇怪的力場徹底破壞，消失於虛空之中。

於是很久之前你曾不時對著蟲洞投入了不少感應器，這天你想要整理感應器的資料。可是每個感應器當初只有記錄穿越蟲洞前的時間和穿越後的時間，你想要計算每個感應器被丟入蟲洞後的時間差了多少。

■ 輸入說明

輸入的第一行有一個正整數 $T(T \leq 50000)$，代表測試資料的組數。

每一組測試資料的只有一行。該行會有兩組時間，分別表示穿越蟲洞前的時間和穿越後的時間，中間以一個空白和逗號隔開。

時間內所有數值都是可能有 0 開頭的兩位整數。時間格式：“m-d H:M:S” (不含雙引號)

- m: 月，$01 \leq m \leq 12$。
- d: 日，
  - 如果 $m = 1, 3, 5, 7, 8, 10, 12$，$01 \leq d \leq 31$。
  - 如果 $m = 4, 6, 9, 11$，$01 \leq d \leq 30$。
  - 如果 $m = 2$，$01 \leq d \leq 28$。
- H: 時，$00 \leq H \leq 23$。
- M: 分，$00 \leq M \leq 59$。
- S: 秒，$00 \leq S \leq 59$。

■ 輸出說明

對於每一筆測試資料請輸出一行，表示感應器進入蟲洞前後的時間差，且需把時間差換算成可讀的格式。

時間差格式：“d H:M:S” (不含雙引號)

- d: 天數。**如果是回到過去 (時間差是負數)，則 d 需要加上負號**。
- H: 時，$00 \leq H \leq 23$。
- M: 分，$00 \leq M \leq 59$。
- S: 秒，$00 \leq S \leq 59$。

H, M, S 強制**使用 2 位數**，如果只有 1 位，請在開頭補上 “0” (不含雙引號)。

■ 範例輸入

```
3
05-14 05:14:00, 05-14 05:14:00
05-14 05:14:00, 05-15 15:31:36
05-15 15:31:36, 05-14 05:14:00
```

■ 範例輸出

```
0 00:00:00
1 10:17:36
-1 10:17:36
```