2012 網際網路程式設計全國大賽
高中組決賽

- 題目：本次比賽共八題（含本封面共 18 頁）。
- 題目輸入：全部題目的輸入都來自**標準輸入**。
  輸入中可能包含多組輸入，依題目敘述分隔。
- 題目輸出：全部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，英文字母大小寫不同視為答題錯誤。
- 時間限制：每一題的執行時間限制如表 1 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。
- `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**	曉涵的紙牌遊戲	5 秒
題目 **B**	蚯蚓疊積木	5 秒
題目 **C**	蚯蚓的遺跡保衛戰 2	10 秒
題目 **D**	小可魚兒向上游	10 秒
題目 **E**	胖胖天天天胖	10 秒
題目 **F**	史萊姆森林	10 秒
題目 **G**	古代巨塔之謎	5 秒
題目 **H**	好大一座金礦	5 秒

2012 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：

```
#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：

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

題目 A
曉涵的紙牌遊戲

執行時間限制: 5 秒

曉涵和朋友們最近流行玩一個遊戲：首先準備好一疊紙牌，每張紙牌都有一個編號，編號為 1 到 N 之間的正整數，假設現在共有 P 位玩家 (包含曉涵在內)，則每個編號都分別有 P 張紙牌 (故共有 $P \times N$ 張紙牌)。之後，每個人分別寫下一些「禁忌序列 (forbidden sequence)」，每個禁忌序列都是一串遞增的編號，長度和序列內的編號由每個玩家自行決定，但編號不可重複，並且在遊戲開始前都不能給其他人看到。

大家都寫完自己設計的禁忌序列之後，由一個人將這 $P \times N$ 張紙牌拋起、使其散落在地板上，同時，大家都公布出自己設計的禁忌序列，此時遊戲正式開始！遊戲的目標在於，玩家要從散落的紙牌中找出一個比其他玩家都長的遞增序列以獲得勝利，其中不能存在超過一張同樣編號的紙牌，且遞增序列長度必須至少為 2 (至少要選兩張紙牌) ，且其中**不可以包含**任何人 (包括自己) 所設計的禁忌序列。假設有一個人的禁忌序列是「1, 2, 4」，則「1, 2, 4, 5」便是不合法的遞增序列，但「1, 2, 3, 4」則是合法的。當然，像「1, 1, 2, 2」由於有同樣編號的紙牌，所以也是不合法的。

對於這個紙牌遊戲，好奇的曉涵除了想要贏之外，她還想要知道在給定大家的禁忌序列的情況下，究竟有多少種不同的**合法遞增序列**呢？然而慢慢數真的太累了，請你寫一個程式來幫曉涵計算出總共有多少種不同的遞增序列。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 50$)，代表總共有幾筆測試資料。

每筆測試資料第一行有一個整數 N ($2 \leq N \leq 10$)，表示紙牌的最大編號。第二行有一個整數 P ($2 \leq P \leq 10$) 代表共有多少位玩家 (包含曉涵在內)。

之後從第三行開始共有 P 行，每一行包含一位玩家所設計的「禁忌序列」。首先有一個正整數 L_i ($2 \leq L_i \leq N$)，代表該玩家所設計的禁忌序列的長度，之後有 L_i 個正整數，依序代表該禁忌序列中的各個編號。所有人設計的禁忌序列都是遞增的且其中包含的編號都是在 1 到 N 之間。

由於玩家之間遊戲前是看不到其他人設計的禁忌序列的，所以可能會有多位玩家設計出相同的禁忌序列。

■ 輸出檔說明

對於每一組測試資料請輸出一行，該行包含一個正整數代表共有多少種不同的遞增序列。

如果沒有任何遞增序列是合法的話，請輸出一行 "`so sad`" (不含引號)。

■ 範例輸入

```
4
5
3
3 2 3 4
2 2 3
2 2 4
7
2
2 1 2
2 6 7
5
4
5 1 2 3 4 5
5 1 2 3 4 5
5 1 2 3 4 5
5 1 2 3 4 5
3
4
2 1 2
2 2 3
2 1 3
3 1 2 3
```

■ 範例輸出

```
14
64
25
so sad
```

題目 B
蚯蚓疊積木

執行時間限制: 5 秒

還記得老蚯的那些寶物嗎？隨著蚯蚓們挖到的寶物越來越多，蚯蚓的公用倉庫越疊越高，老蚯發現自己竟愛上了疊東西。為了滿足自己疊東西的欲望，老蚯從寶物堆中找出了許多大小不一的積木，不停的疊來疊去。

然而，在疊了七七四十九年後，單純的疊積木再也無法滿足老蚯了。為了讓疊積木變得更有趣，老蚯決定出個考題考考自己：依照某個順序每次拿起一個積木，一一決定是否將這個積木疊到積木塔上，並使得最後積木塔上疊的積木最多。當然，為了要疊出一個穩固的積木塔，任何一個積木都只能疊在一個比他自己大的積木上。

幸運，也不幸的，因為老蚯天生對積木的直覺，這個問題在老蚯嘗試玩了三次以後就變得簡單無聊。所以老蚯決定問問自己一個更有挑戰性的問題：有哪些積木如果被規定了不准疊到積木塔上，會使得能疊到積木塔上的最多積木數量減少？

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 6000$)，表示接下來總共有幾筆測試資料。

每組測試資料的第一行的開頭有一個正整數 N ($1 \leq N \leq 200000$) 代表蚯蚓打算依序拿起 N 個積木，同一行接著有 N 個整數 S_i ($1 \leq S_i \leq N$)，代表那 N 個積木的大小，所有積木的大小都是不同的。我們保證有 99% 的測試資料 $N \leq 1000$ 。

■ 輸出檔說明

為了輸出方便，我們先定義一個雜湊函數 $hash$，能夠將一個長度為 $size$ 的整數序列轉換成一個整數。

```
int hash(int numbers[], int size){
    int value = 0, i;
    for(i = 0; i < size; i++){
        value ^= (numbers[i] + i + 1);
    }
    return value;
}
```

對於每一筆測試資料，請輸出中間用一個空白分隔的兩個整數 A, B。A 代表有幾個積木若被規定不准放到積木塔上，會使得能疊到積木塔上的最多積木數量減少。B 代表將那 A 個積木的索引值 (依據輸入順序從 1 開始編號到 N) 從小排到大後雜湊的結果。

■ 範例輸入

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

■ 範例輸出

```
0 0
4 4
3 14
```

■ 範例說明

第三筆測試資料中，2 3 5 6 或 2 3 4 6 是唯二能疊四個積木的疊法。但拔掉重量為 4 或 5 的積木都還會剩下一種疊法，所以只有第 1、2、5 個積木是重要的。因此答案為將 $[1, 2, 5]$ 這個序列雜湊後的結果，即 $(1 + 1) \oplus (2 + 2) \oplus (5 + 3) = 14$。

題目 C
蚯蚓的遺跡保衛戰 2

執行時間限制: 10 秒

有一款遊戲叫做遺跡保衛戰。最近出了第二集，轟動了蚯蚓界，而蚯蚓界霸者 — 老蚯，也不能跟不上流行。但不幸的是，雖然身為蚯蚓界的霸者，但仍是遊戲界的菜鳥。於是他要他的謀士，也就是你，幫他規劃一下玩遊戲的策略。

這遊戲始創於胖胖天界，後來傳到蚯蚓界的時候，規則有做了一些修改，畢竟這是蚯蚓的世界。首先遊戲主角每秒鐘會得到一塊錢，可以用來買裝備加強自身的能力。但是每件裝備都會有不同的價格和增益效果 (例如提昇力量、加強攻擊力 ...)，為了方便起見，我們把增益效果都量化成一個整數，稱之為增益值。所以，第 i 件裝備的價格是 p_i 以及增益值是 b_i。

有趣的是同類型的裝備們會融合，例如你買了兩雙鞋子，增益值分別是 b_1, b_2。他們會自己組合成一雙增益值為 $b_1 + b_2$ 的裝備。所以你不用考慮買太多裝備的問題，買就對了！

話說回來，這遊戲有個限制，要求你必須先買下 ‘‘老蚯的祝福’’，才可以開始購買其他裝備。而 ‘‘老蚯的祝福’’ 這件裝備，價值 1 塊錢且增益值為 1。

接下來，他想要盡快買齊他想要的 n 件裝備。然而，雖然有很多種方式都能在最短的時間買齊所有裝備，但有些成效很差。舉例來說有時候直接存錢買比較貴的裝備不一定比較好，因為這樣做的話，可能會導致遊戲前期沒辦法買太好的裝備，那麼前期的裝備太弱會玩得很辛苦。

於是老蚯希望你幫他挑選一個買裝備的策略，使得從遊戲一開始 (時間 0) 到買齊所有裝備這段時間的平均增益值最大。因為這樣的話，可以讓遊戲過程過的比較平滑，比較不會太崩潰。另外，我們定義，在某一秒鐘的增益值，為主角當前所有已買的裝備增益值總和。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 10$)，表示接下來總共有幾筆測試資料。

每筆測試資料的第一行有一個整數 n ($0 \leq n \leq 10^5$)，表示老蚯想買 n 件裝備。接下來會有 n 行，第 i 行有兩個整數 p_i 和 b_i ($0 \leq p_i, b_i \leq 10000$)，分別表示第 i 件裝備的價格和增益值。注意這 n 件裝備不含 ‘‘老蚯的祝福’’。

■ 輸出檔說明

對於每一筆測試資料請輸出中間用一個空白隔開的兩個整數 p, q，分別為平均值的分子和分母。注意這個分數必須是最簡分數，也就是說 p, q 要互質，即 p 與 q 的最大公因數為 1。如果 $p = 0$，請輸出 $\frac{0}{1}$。

■ 範例輸入

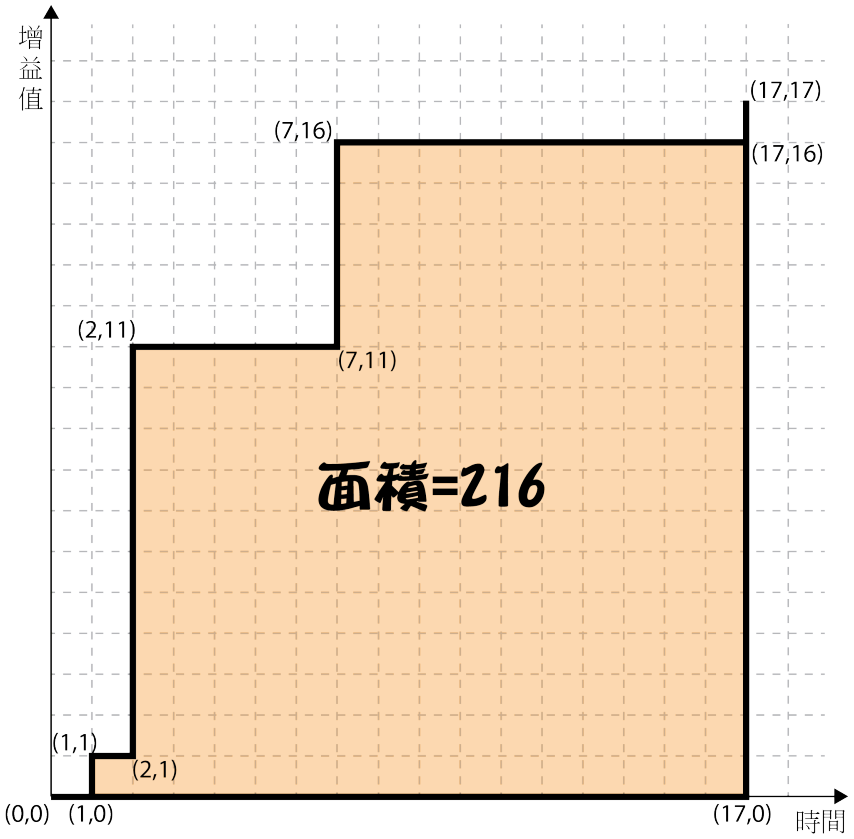
```
2
3
1 10
5 5
10 1
3
1 1
5 5
10 10
```

■ 範例輸出

```
216 17
81 17
```

■ 範例說明

第一筆測試資料中，最佳策略中增益值的增長如下圖所示。



題目 D
小可魚兒向上游

執行時間限制: 10 秒

江江江江，有一條江耶！

經過長時間的觀察，老姜發現這條江有許多支流，且整個河系可以用一個以源頭為根的樹狀結構來表示。除此之外老姜還發現有許多小可魚循著某種規律在江河中悠游穿梭，每當一隻小可魚開始逆流而上時，牠會先在起點做個記號，之後開始奮發向上游阿游阿游，直到源頭或是遇到其他小可魚在牠開始前做的記號為止。

聰明的老姜已經先寫了一個程式來預測每隻小可魚最後會停在哪裡，但謹慎的他還是請你也寫一份來確認他寫的程式是否正確。但為了肉眼比對方便，聰明的老姜決定比較結果的雜湊值就好，也就是假設小可魚最後停留的位置依序為 $p_1, p_2, \ldots, p_k$，則我們定義其雜湊值為 $(p_1 + 1) \oplus (p_2 + 2) \oplus \cdots \oplus (p_k + k)$，其中 $\oplus$ 代表 xor 運算。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 514$)，表示接下來總共有幾條江的觀察記錄。

每個觀察記錄開始有兩個正整數 N, M ($M < N \leq 10^5$)，代表這條江的樹狀結構有 N 個節點，M 隻小可魚。節點編號為 $0, 1, \ldots, N-1$，其中 0 為源頭。下一行有 $N-1$ 個整數，代表每個節點（支流）$1, 2, \ldots, N-1$ 的上游編號。再下一行有 M 個相異整數，代表每隻小可魚的起點，輸入順序為小可魚們開始向上的順序。

■ 輸出檔說明

對於每筆觀察紀錄，請輸出其對應的雜湊值。若對雜湊值之運算有所疑惑可以參考題目 B 所附之程式碼。

■ 範例輸入

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

■ 範例輸出

```
7
6
```

■ 範例說明

第一筆測試資料中的江系：$0 \leftarrow 1 \leftarrow 2 \leftarrow 3$

1. 一開始有隻小可魚在 3 號節點做記號並開始往上游，直到源頭為止，路線為 $3 \rightarrow 2 \rightarrow 1 \rightarrow 0$，停在 0。
2. 接下來有隻小可魚在 1 號節點做記號並開始往上游，直到源頭為止，路線為 $1 \rightarrow 0$，停在 0。
3. 接下來有隻小可魚在 2 號節點做記號並開始往上游，直到 1 號節點為止，路線為 $2 \rightarrow 1$，停在 1。

最後輸出的雜湊值為 $(0+1) \oplus (0+2) \oplus (1+3) = 7$。

第二筆測試資料中的江系：$0 \leftarrow 2 \leftarrow 1, 3, 4$

1. 一開始有隻小可魚在 1 號節點做記號並開始往上游，直到源頭為止，路線為 $1 \rightarrow 2 \rightarrow 0$，停在 0。
2. 接下來有隻小可魚在 2 號節點做記號並開始往上游，直到源頭為止，路線為 $2 \rightarrow 0$，停在 0。
3. 接下來有隻小可魚在 3 號節點做記號並開始往上游，直到 2 號節點為止，路線為 $3 \rightarrow 2$，停在 2。

最後輸出的雜湊值為 $(0+1) \oplus (0+2) \oplus (2+3) = 6$。

題目 E
胖胖天天天胖

執行時間限制: 10 秒

胖胖天，天天胖！

身為一個專業飲料愛好者，胖胖天經常研究飲料相關的問題並樂此不疲。現在有許多飲料在胖胖天面前排成一排，每個飲料都有其購買的成本。他想從特定區段中挑出盡量多瓶飲料，使得總成本不超過他錢包裡的錢，請問他最多可以挑幾瓶呢？

聰明的胖胖天已經先寫了一個程式來輔助他喝遍天下的飲料，但謹慎的他還是請你也寫一份來確認他寫的程式是否正確。但為了肉眼比對方便，聰明的胖胖天決定比較結果的雜湊值就好，也就是假設對每次詢問所能購買的瓶數依序為 $p_1, p_2, \ldots, p_k$，則我們定義其雜湊值為 $(p_1 + 1) \oplus (p_2 + 2) \oplus \cdots \oplus (p_k + k)$，其中 $\oplus$ 代表 xor 運算。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 514$)，表示接下來總共有幾筆測試資料。

每一筆測試資料第一行有兩個正整數 N, Q ($N, Q \leq 10^5$)，代表總共有 N 瓶飲料，胖胖天將問你 Q 個問題。飲料編號由左到右為 $1, 2, \ldots, N$。下一行有 N 個非負整數，依序代表每瓶飲料的購買成本，飲料的成本不會超過 10^4。再接下來有 Q 行，每行有三個整數 L, R, S ($1 \leq L \leq R \leq N, 0 \leq S \leq 10^9$)，代表胖胖天想問你如果他錢包裡剛好有 S 元，他最多可以買幾瓶編號在 L 到 R 之間的飲料呢？

■ 輸出檔說明

對於每筆測試資料，請輸出其對應的雜湊值。若對雜湊值之運算有所疑惑可以參考題目 B 所附之程式碼。

■ 範例輸入

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

■ 範例輸出

```
3
6
```

■ 範例說明

第一筆測試資料中，胖胖天最多可購買成本為 1 和 4 的飲料各一瓶，最後輸出的雜湊值為 $(2+1)=3$。

第二筆測試資料中，第一個問題胖胖天最多可購買成本為 1 的飲料兩瓶，第二個問題胖胖天最多可購買成本為 1 的飲料三瓶，最後輸出的雜湊值為 $(2+1)\oplus(3+2)=6$。

題目 F
史萊姆森林

執行時間限制: 10 秒

傳說中，在 NPSC 大陸的某片森林深處住著一群史萊姆。這群史萊姆非常好戰，每當有兩隻史萊姆相遇時，他們總會打 10^{18} 回合，或是打到有其中一方戰死才肯罷休。身為一個史萊姆生態研究員，蚯蚓決定前往這片森林研究這些特別的史萊姆。

在觀察後，蚯蚓發現一個特別的現象：兩隻史萊姆打架時，常常分不出勝負，而且輸贏跟兩隻史萊姆的大小似乎沒什麼關係！在更進一步研究後，蚯蚓發現這似乎與史萊姆特殊的打架方式有關。

假設有兩隻大小為 a, b $(a > b)$ 的史萊姆在打架，則大小為 b (比較小) 的那隻會發動攻擊，並且在攻擊完後將 a 的一部分吸收。這一回合後兩隻史萊姆的大小會變為 $a - b, b + b$ ，再由比較小的那隻史萊姆再度發動攻擊，以此類推。若當兩隻史萊姆的大小變得一樣時，兩方會發動最後的攻擊，而速度較快的一方就會擊敗對方、將對方完全吸收，而若在打了 10^{18} 回合後都沒有一方戰敗，兩隻史萊姆就會休戰。

蚯蚓很好奇，在眾多的史萊姆中，究竟有幾對史萊姆相遇時是會分出勝負的？

■ 輸入檔說明

輸入檔的第一行有一個正整數 T $(T \leq 10)$，表示接下來總共有幾筆測試資料。

每一筆測試資料的開頭有一個正整數 N $(N \leq 10^5)$，代表森林中有幾隻史萊姆。接下來有 N 個正整數 Si 代表該隻史萊姆的大小，史萊姆的大小不會超過 500000。

■ 輸出檔說明

對每筆測試資料請輸出一個整數，代表有幾對史萊姆相遇時是會分出勝負的。

■ 範例輸入

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

■ 範例輸出

```
0
1
```

■ 範例說明

第二筆測試資料中，只有 $(1, 3)$ 能分出勝負。

題目 G
古代巨塔之謎

執行時間限制: 5 秒

在綠草如茵、一望無際的賽姆利亞平原上，座落著數座神祕的高塔，根據考古學家的調查，這些「古代巨塔」似乎都是在數千年前的同一時期所建造的，儘管他們的高度和現今的樣貌不盡相同，但整體而言我們都可以感受到這些古代塔宇所帶給我們的震撼和難以形容的壓迫感。

對於這些古代巨塔，許多專家學者紛紛提出了自己的假說，但令人困惑的是，目前在賽姆利亞平原上都沒有找到一些看起來和這些高塔有關的古物或是遺跡，而這也讓這些古代巨塔的背景始終難以被確認。至今為止沒有人知道這些高塔的用處，這些塔宇也沒有看似窗戶的空洞，每座塔都只有唯一的入口以及通往塔頂的階梯。

艾絲蒂爾是現在正就讀於利貝爾學園的二年級學生，有一天她在歷史課中得知了這古代巨塔之謎。最近，她也在另一門課中聽到了一個關於神祕古代帝國埃雷波尼亞的傳說。突然之間，艾絲蒂爾產生了一個想法：「如果那些古代巨塔是埃雷波尼亞帝國的建築的話 ⋯⋯ ?」但是，有什麼方法可以稍微確認這個假說的正確與否呢？

於是，艾絲蒂爾便去做了一些調查，她得知有一些古代的文獻之中，記載著疑似埃雷波尼亞帝國的領土面積。於是她便想道：「如果這些古代巨塔都是在當時埃雷波尼亞帝國的領土內的話，就可以用這些塔來推測出當時帝國的領土面積了！」而如果推測出的面積跟文獻中記載的帝國領土面積相近，那麼或許這個假說就有可能是正確的了！

那些相關文獻中同時也提到埃雷波尼亞帝國的領土是一個完整的矩形 (即長方形)，故艾絲蒂爾便想要算出可以包含住這些古代巨塔的最小矩形面積 (如果假說是正確的，則帝國的領土面積至少這麼大)。然而，在她從圖書室中找出標示著古代巨塔座標的地圖的當下，她就楞住了。「這 ⋯⋯ 這也太多座塔了吧！?」艾絲蒂爾絕望地看著手上的地圖，可能的矩形實在是太多了，要怎麼找出可以包含住這些古代巨塔的最小矩形面積呢？請你寫一個程式幫幫艾絲蒂爾，找出最小的可能面積。

■ 輸入檔說明

輸入檔的第一行有一個正整數 T ($T \leq 50$)，代表總共有幾組測試資料。

每組測試資料的第一行有一個正整數 N ($3 \leq N \leq 3000$)，代表在地圖上總共標示著多少座古代巨塔。接下來有 N 行，每一行都有兩個整數 x_i, y_i，分別代表一座古代巨塔的 x, y 座標。所有 x, y 座標的絕對值都不會超過 10^4。

此外，為了簡化題目，我們可以視每座塔皆為一半徑為 0 的單點 (從高空俯瞰時塔的大小相較於一單位的座標距離而言極小，故我們在此忽略它的半徑。)。

■ 輸出檔說明

對於每一組測試資料，請輸出最小可以包含住這些古代巨塔的矩形面積。輸出請四捨五入至小數點後第四位。

■ 範例輸入

```
3
5
−1 0
−1 1
−1 −1
1 1
1 −1
5
8 9
10 −2
−2 3
−4 8
−5 5
8
7328 10000
7329 9999
7329 3797
7328 3796
1126 3796
1125 3797
1125 9999
1126 10000
```

■ 範例輸出

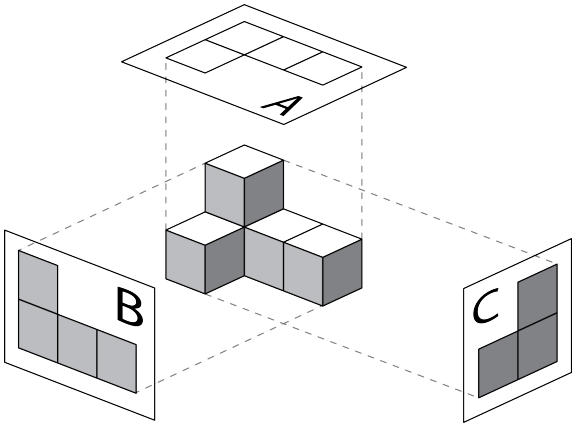
```
4.0000
157.3728
38489616.0000
```

題目 H
好大一座金礦

執行時間限制: 5 秒

老姜是一個快樂的礦工，每天他都會提著他的鐵鍬與他的六個好朋友一起出門去尋找新的礦脈。老姜和他的朋友們都很喜歡黃金，每次找到了新的金礦，他們都會很開心地邊唱歌邊把黃金通通挖出來。

有一天，老姜在一座山中發現了新的黃金礦脈。他快速地呼叫他的朋友們來開挖，但是這座山太大了，需要大量的成本才能進行挖掘。因此他們打算先估計這座山中究竟有多少金礦再決定之後該怎麼開採，為此老姜請了專家對整座山從三個方向進行掃描，並得出了三張結果圖 A, B, C。但是老姜和他的朋友們看完結果圖後還是不太懂山裡有多少金礦，請你寫一個程式來幫助他們估計。



假設山裡存在如上圖的一塊金礦，則三張結果圖 A, B, C 分別如上圖所示，結果圖 A 表示由正上方掃描後發現該單位面積投影所對應的空間中是否存在金礦，你可以假設金礦都是由一單位的立方體所構成的。

專家們都是很專業的，不會給老姜不合理的結果圖，意即一定可以算出山裡可能有多少金礦。

■ 輸入檔說明

檔案的第一行有一個正整數 T ($T \leq 100$)，表示接下來總共有幾筆測試資料。

每組測試資料的第一行有三個整數 L, W, H 表示整座山的尺寸，而三張結果圖的尺寸分別是 $L \times W, W \times H, L \times H$。接下來有 W 行，每行有 L 個字元表示結果圖 A，接下來有 H 行，每行有 W 個字元表示結果圖 B，接下來有 H 行，每行有 L 個字元表示結果圖 C，每張結果圖中的 "." 表示該單位面積投影所對應的空間中不存在金礦，其他字元則表示存在金礦。($1 \leq L, W, H \leq 100$)

■ 輸出檔說明

對每筆測試資料輸出一行，每行包含一個整數，代表山裡存在的金礦總體積。答案可能有多組解，請輸出其中最大的那一組。

■ 範例輸入

```
3
2 3 3
AA
.A
.A
...
B..
BBB
..
.C
CC
3 3 1
.A.
A.A
.A.
BBB
CCC
2 3 3
AA
.A
AA
BBB
B..
BBB
CC
C.
CC
```

■ 範例輸出

```
5
4
11
```