

2014 網際網路程式設計全國大賽

國中組初賽

- 本次比賽共 6 題，含本封面共 14 頁。
- 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。
- 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，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 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。
- 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。
- 每一題的執行時間限制如下表所示。
執行期間該電腦不會有別的動作，也不會使用鍵盤或滑鼠。

	題目名稱	執行時間限制
題目 A	棒球中的統計學	1 秒
題目 B	雨澤的鞋櫃	7 秒
題目 C	新 · $3N+1$	1 秒
題目 D	貓熊羽哲	1 秒
題目 E	等差數列	2 秒
題目 F	小可魚寫作業	1 秒

2014 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：

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%I64d %I64d", &a, &b);
10        printf("%I64d\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

題目 A

棒球中的統計學

執行時間限制: 1 秒

俗話說：「你怎麼能不愛棒球呢？」身為棒球迷，除了球賽本身的內容以外，「球賽數據」也往往是大家關注的焦點。近年來，除了打擊率 (AVG, batting average)、打點 (RBI, run batted in) 以及自責分率 (ERA, earned run average) 等等大家熟知的基本棒球數據以外，每局被上壘率 (WHIP, walks plus hits per inning pitched) 以及長打率 (SLG, slugging percentage) 等等進階數據，也越來越為人所知。

事實上，這些統計數據並不只有給粉絲或專家分析用，這幾年的職業比賽也導入了基於分析數據而產生的戰術，如防守佈陣 (shifts) 等等。從美國、日本到台灣，轉播單位也慢慢地讓觀眾看到各式各樣的統計數據，而大家也逐漸地感受到這些數據確實有在球賽中反映其統計意義。

在棒球統計的進階領域中，「賽伯計量學 (Sabermetrics)」是運動科學中最成功且最著名的學門之一，如著名的比爾詹姆斯 (Bill James) 便是賽伯統計的支持者之一。而其中最著名的指數，莫過於「攻擊指數 (OPS, on-base plus slugging)」了。OPS 的算法就如同它的英文名稱一樣：上壘率 (OBP, on-base percentage) 加長打率。若我們將各項數據展開，則 OPS 的公式會變成

$$\text{OPS} = \frac{\text{AB} \times (\text{H} + \text{BB} + \text{HBP}) + \text{TB} \times (\text{AB} + \text{BB} + \text{SF} + \text{HBP})}{\text{AB} \times (\text{AB} + \text{BB} + \text{SF} + \text{HBP})},$$

- AB: 打數 (at bats)
- H: 安打數 (hits)
- BB: 四壞球保送次數 (base on balls)
- HBP: 觸身球次數 (times hit-by-pitch)
- TB: 總壘打數 (total bases)
- SF: 高飛犧牲打次數 (sacrifice flies)

現在，你從網路上搜集到了一些穗香聯盟於草創期的打擊記錄片段。很不幸地，由於年代久遠，你手上的記錄並沒有包含攻擊指數，只有一些基本的數據。身為一個棒球迷，你很想知道當時最厲害的打者是誰，所以決定寫一支計算攻擊指數的程式。

為了簡化題目，我們保證給定的數據只包含計算 OPS 會用到的項目。

■ 輸入檔說明

輸入的第一行有一個正整數 N ，代表共有幾位球員。

接下來共有 N 行，每一行都有六個整數，分別依序代表打數、安打數、四壞球保送次數、觸身球次數、總壘打數和高飛犧牲打次數。

- $N \leq 200$
- 所有的數值都是非負整數且不會超過 100000
- 保證 AB 至少為 1。

■ 輸出檔說明

對於每一筆輸入的球員資料，請輸出一行，其中包含一個實數表示該球員的攻擊指數。請將數值四捨五入後輸出至小數點下第三位。

■ 範例輸入

```
3
539 155 94 3 299 2
644 181 19 12 302 7
561 188 70 4 317 6
```

■ 範例輸出

```
0.950
0.780
0.974
```

題目 B

雨澤的鞋櫃

執行時間限制: 7 秒

雨澤有句名言：「無論再怎麼難看的款式，仍是會有人喜歡，所以不怕賣不出去。」

雨澤是一個鞋店老闆，他有一個兩層的櫃子，上層放著一排高級鞋，下層放著一排山寨鞋。為了簡化問題，每一雙高級鞋都可以用一個正整數來表示它的種類，而山寨鞋是仿冒高級鞋而成，所以每一雙山寨鞋會各別與某一雙高級鞋同種類。

雨澤是一個很有愛心的人，他想要捐給慈善團體一雙高級鞋與若干雙山寨鞋，雖然他也有順便清理鞋櫃的意思。但在捐鞋之餘，他有個特殊的要求，他希望在捐完鞋子後，高級鞋與山寨鞋的數量相同。而且在不改變剩餘鞋子的順序下重新排好後，第 i 雙高級鞋與第 i 雙山寨鞋是同種類。

舉個例子，假設一開始上排鞋櫃的鞋子有 3 雙高級鞋，種類依序為 $\langle 5, 1, 1 \rangle$ ，而下排鞋櫃有 6 雙山寨鞋，種類依序為 $\langle 3, 1, 4, 1, 5, 9 \rangle$ ，雨澤唯有捐第 1 雙高級鞋以及第 1, 3, 5, 6 雙山寨鞋才能達成他的要求，此時上下排剩下的鞋子數皆為 2 雙，上排鞋子種類依序為 $\langle 1, 1 \rangle$ ，下排鞋子種類依序亦為 $\langle 1, 1 \rangle$ 。

雨澤很聰明，若他知道要捐哪雙高級鞋可以達到他的要求，則可以很快地找出他要捐出哪些山寨鞋。因此雨澤希望如果存在一種捐法可以達到他的要求，請你告訴他該捐出第幾雙高級鞋。

■ 輸入說明

輸入的第一行有一個正整數 T ，代表測試資料的筆數。

每一筆測試資料的第一行有兩個以空白隔開正整數 N, M ，表示高級鞋有 N 雙且山寨鞋有 M 雙。第二行有 N 個正整數 A_i ，表示高級鞋的種類，第三行有 M 個正整數 B_i ，表示山寨鞋的種類。

- $T \leq 100$
- $2 \leq N, M \leq 100000$
- $1 \leq A_i, B_i \leq 100000$

■ 輸出說明

對於每一筆測試資料，若雨澤可以達成目的，請輸出他該捐出第幾雙高級鞋。若有多種答案，請告訴他最靠左的。若無法滿足他的要求則輸出 -1 。

■ 範例輸入

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

■ 範例輸出

```
2
1
1
```

■ 範例說明

- 第一筆範例中，雨澤捐第 2 雙與第 3 雙高級鞋都可以達到目的。
- 第二筆範例中，雨澤捐第 1 雙與第 2 雙高級鞋才可以達到目的。
- 第三筆範例中，雨澤捐哪雙都可以達到目的，因此捐第 1 雙高級鞋。

題目 C

新 · $3N+1$

執行時間限制: 1 秒

寧寧是個對程式極有興趣的國中生。有天，當她在網路上悠遊時看到了一個令她覺得神奇的問題： $3N+1$ 問題。

$3N+1$ 問題是這樣的：給一個正整數 N ，如果它現在是偶數就除以二，否則乘以三加一，重複一直做下去，問你要做幾步才會變成一。

寧寧雖然覺得不可思議，也質疑是否所有的數字到最後都會變成一，但聰穎的她仍很快的解出了這題。然而身為一個精益求精的好學生，她並不滿足於此。「題目規定只能乘以三加一，那如果改成乘以 A 加 B 會怎麼樣呢？」寧寧問著自己。

為了挑戰自己的智慧，她定義了新 · $3N+1$ 問題：給一個正整數 N ，請找到最佳的正整數 A, B 使得將原問題中的乘以三加一換成乘以 A 加 B 後數字最快變成一。

不幸的，這個問題變得太難，以至於寧寧無法快速解出來。你有辦法幫助她嗎？

■ 輸入說明

輸入的第一行有一個正整數 T ，代表測試資料的筆數。

每一筆測試資料包含一個正整數 N ，代表新 · $3N+1$ 問題中給定的 N 。

- $T \leq 300000$
- $N \leq 10^9$

■ 輸出說明

對於每一筆測試資料請輸出一行，包含兩個最佳的正整數 A, B 。

若有多組解，請輸出 A 最小的。若仍有多組，請輸出 B 最小的。

■ 範例輸入

```
2
4
5
```

■ 範例輸出

```
1 1
1 3
```

■ 範例說明

- 第一筆範例中，不論 A, B 是多少， N 都會一直除以二，故直接輸出最小的 A, B 即可。
- 第二筆範例中，最短的過程為 $5 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 。

題目 D

貓熊羽哲

執行時間限制: 1 秒

貓熊羽哲是一隻很愛吃棉花糖的熊貓，在某次旅行中，他來到了一塊很像綠豆糕的田地，上面散落著許多棉花糖。

	1	2	3	4
1	6	1	1	1
2	9	5	1	1
3	2	2	7	1

田地長得就像稿紙，方方正正好像棋盤似的。每一格上面有著若干棉花糖，我們以 $V_{i,j}$ 表示第 i 行第 j 列的棉花糖數量。羽哲每次可以將他所在格內的棉花糖吃完，但不幸的是羽哲無法將田地中的所有棉花糖吃完，因為他的體力只夠走 3 步，每步只能往上下左右其中一個方向走一格。

羽哲很想知道如果他可以從任意一格開始，他最多可以吃到幾個棉花糖？充滿雅量的你可以幫他解決他卑微的困惑嗎？

■ 輸入說明

輸入的第一行有一個正整數 T ，代表測試資料的筆數。

每一筆測試資料的第一行有兩個以隔開正整數 N, M ，接下來有 N 行，每行有 M 個的正整數 $V_{i,j}$ ，表示這一行每格棉花糖的數量。

- $T \leq 150$
- $2 \leq N, M \leq 50$
- $1 \leq V_{i,j} \leq 10000$

■ 輸出說明

對於每一筆測試資料，請輸出一行包含一個整數，表示羽哲最多可以吃到幾個棉花糖。

■ 範例輸入

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

■ 範例輸出

```
23
32
```

■ 範例說明

- 第一筆範例如敘述所示，羽哲會從 (2,1) 開始，吃 (2,1), (2,2), (3,2), (3,3) 格，共吃 23 單位棉花糖。
- 第二筆範例中，羽哲會從 (1,1) 開始，吃 (1,1), (2,2), (2,2), (2,3) 格，共吃 32 單位棉花糖。

題目 E

等差數列

執行時間限制: 2 秒

卡恩自從學到等差數列後就情不自禁地愛上它，無論他看到什麼數列都會忍不住想要把它變成等差數列。但是改得面目全非就失去意義了，因此他想要在修改儘量少的數字下完成這件事。現在他面前有一個數列 $\langle a_1, a_2, \dots, a_n \rangle$ ，請問他至少要修改其中幾個數字才能把它變成等差數列呢？

如果你忘記什麼是等差數列的話，卡恩很難過但還是會提醒你。我們說一個數列是等差數列，若且唯若它任何相鄰兩項的差相等。

舉例來說， $\langle 3, 5, 7, 9, 11, 13 \rangle$ 、 $\langle 9, 8, 7, 6 \rangle$ 、 $\langle 1.23, 1.46, 1.69 \rangle$ 都是等差數列，而 $\langle 5, 1, 4 \rangle$ 不是等差數列因為 $1 - 5 \neq 4 - 1$ 。

■ 輸入說明

輸入的第一行有一個正整數 T ，代表測試資料的筆數。

每一筆測試資料包含兩行。第一行會有一個整數 n ，代表卡恩要修改的原始數列的長度。第二行會有 n 個整數 a_1, a_2, \dots, a_n ，依序代表數列中每項的數值。

- $1 \leq T \leq 500$
- $1 \leq n \leq 200$
- $1 \leq a_i \leq 10^6$

■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示卡恩至少要修改幾個數字才能把原數列變成等差數列。

■ 範例輸入

```
3
3
1 2 3
3
5 1 4
6
3 1 4 1 5 9
```

■ 範例輸出

```
0
1
3
```

■ 範例說明

- 第一筆範例的數列不需修改。
- 第二筆範例的數列可修改為 $\langle -2, 1, 4 \rangle$ 。
- 第三筆範例的數列可修改為 $\langle 3, 3.5, 4, 4.5, 5, 5.5 \rangle$ 。

題目 F

小可魚寫作業

執行時間限制: 1 秒

小可魚今天雖然只有一項回家作業，但他今天很懶，不想寫作業，所以他希望有個程式可以幫忙算出作業的答案。

而作業題目是這樣的：

有一個由字母和數字所組成的字串 s 。請計算出 s 有幾個子字串是合法的數字字串？

我們定義 s 的一個子字串 $s(i, j)$ ，則 $s(i, j) = s_i, s_{i+1}, \dots, s_{j-1}, s_j$ 。

如果兩個子字串的起點或終點的位置不同，則視為不同子字串。也就是說，如果 $i \neq i'$ 或 $j \neq j'$ 則 $s(i, j)$ 和 $s(i', j')$ 是不同子字串。

一個合法的數字字串，必須全部由純數字字元組合，且不能以 0 開頭。注意，0 本身不算是以 0 開頭，是合法的數字字串。

■ 輸入說明

輸入的第一行有一個正整數 T ，代表測試資料的筆數。

每一筆測試資料有兩行，第一行有個正整數 n ，代表 s 的長度。第二行則是字串 s 。

- $T \leq 30$
- $1 \leq n \leq 100000$
- s 是由小寫英文字母「a」到「z」及數字「0」到「9」所組成。

■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個數字，代表總共有幾個合法的數字子字串。

■ 範例輸入

```
2
7
101a103
6
sn00py
```

■ 範例輸出

```
10
2
```

■ 範例說明

以下假設索引從 1 開始。

- 第一筆範例的答案為：

1. $s(1, 1) = 1$
2. $s(1, 2) = 10$
3. $s(1, 3) = 101$
4. $s(2, 2) = 0$
5. $s(3, 3) = 1$
6. $s(5, 5) = 1$
7. $s(5, 6) = 10$
8. $s(5, 7) = 103$
9. $s(6, 6) = 0$
10. $s(7, 7) = 3$

- 第二筆範例的答案為：

1. $s(3, 3) = 0$
2. $s(4, 4) = 0$