

## 2014 網際網路程式設計全國大賽 國中組決賽

- 本次比賽共 7 題，含本封面共 22 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	生日快樂喵	1 秒
題目 B	半折植樹	1 秒
題目 C	奧林希克運動會	2 秒
題目 D	寧寧發糖果	5 秒
題目 E	小可魚轉轉轉	2 秒
題目 F	森林小學運動會	1 秒
題目 G	卡恩買飲料	3 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

### 生日快樂喵

執行時間限制: 1 秒

喵喵星球是個快樂的星球，生活中的各種小事都能為喵喵們帶來快樂與歡笑。而一個非負整數  $x$  能為喵喵帶來的快樂度就是  $x$  寫成十進位之後每一位數字的總和。舉例來說，514 的快樂度是  $5 + 1 + 4 = 10$ ，而 50216 的快樂度則是  $5 + 0 + 2 + 1 + 6 = 14$ 。

喵喵星球上有個特殊的傳統，當一隻喵喵過他的  $n$  歲生日的時候，他必須收集很多個「生日快樂」，使得「生日快樂」的數量之快樂度恰好為  $n$ 。例如當喵喵過 5 歲生日的時候，5 跟 14 都是可能的生日快樂個數，因為 5 跟 14 的快樂度都是 5。

現在給你喵喵要過幾歲生日，請問他至少要收集幾個「生日快樂」呢？

#### ■ 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。

每一組測試資料包含一行一個整數  $n$ ，代表喵喵要過幾歲生日。

- $1 \leq T \leq 1000$
- $0 \leq n < 1000$

#### ■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示喵喵至少要收集幾個「生日快樂」。

## ■ 範例輸入

```
4
1
2
5
14
```

## ■ 範例輸出

```
1
2
5
59
```

## ■ 範例說明

- 第一筆範例輸入中 1 的快樂度為  $1 = 1$ 。
- 第二筆範例輸入中 2 的快樂度為  $2 = 2$ 。
- 第三筆範例輸入中 5 的快樂度為  $5 = 5$ 。
- 第四筆範例輸入中 59 的快樂度為  $5 + 9 = 14$ 。

## 題目 B

### 半折植樹

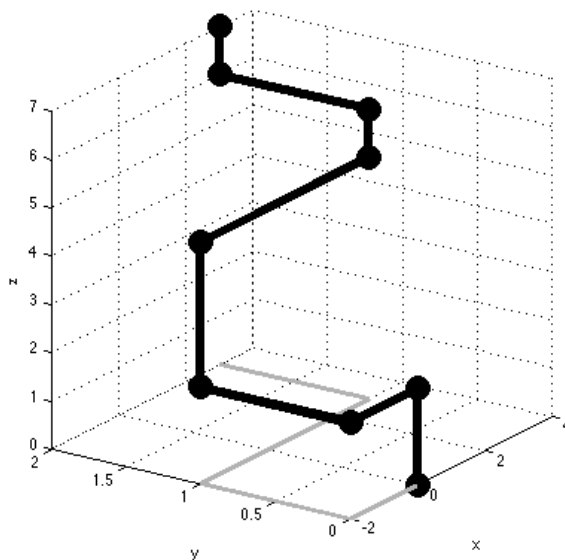
執行時間限制: 1 秒

「整樹的運作，除了桑樹以外就剩下榆樹了。」 – 不學無樹的霍光光大將軍

小可魚很喜歡種樹，除了種桑樹以外，還會用剩餘的畸零地種植榆樹。

不過最近颱風頻頻來襲，導致桑樹和榆樹都被強風豪雨襲擊，是時候重新整頓了！小可魚這次想要種的是可以防風、遮蔭而且正在特價的半折植樹。這種樹很特別，從樹根開始到樹枝的末端，有許多節點，而這些節點必須使連接的兩根樹枝相互垂直 (夾角為九十度)。除了這個條件以外，為了使得整片樹林更加美觀，小可魚希望種下的每一棵樹，其樹枝只能垂直、或平行於地平面。(請參考下圖)

Figure 1: 一棵沒有開花的半折植樹



現在給你一棵樹每一節樹枝的長度，請你幫忙小可魚算算，最高可以達到的樹頂高度為何？噢對了，半折植樹是很厲害的樹：它從樹根到樹頂，絕不會分岔。最後一節 (樹梢) 不需要垂直於地面、但是第一節 (樹根) 一定要垂直於地面。

## ■ 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一個正整數  $N$ ，代表這棵半折植樹的樹枝有幾節。第二行則包含  $N$  個正整數  $a_i$ ，代表由樹根到樹頂每一節樹枝的長度。

- $T \leq 50$
- $3 \leq N \leq 100$
- $1 \leq a_i \leq 1000$

## ■ 輸出說明

對每筆測試資料，請輸出一行，內有一個整數，表示小可魚能夠種下這棵樹其樹頂距離地面的最高高度。

## ■ 範例輸入

```
2
4
3 1 2 7
8
1 2 1 3 2 1 4 1
```

## ■ 範例輸出

```
10
8
```

## 題目 C

# 奧林希克運動會

執行時間限制: 2 秒

「奧林希克運動會」是阿森納帝國最盛大的運動會，每四年於夏季舉辦，全國各地各個運動項目的好手都會齊聚一堂，展開一場為期一個月的運動盛會。在奧林希克運動會中，每個選手都盡全力地把自己最好的能力和技術表現出來，爭取無上榮耀的獎牌。奧林希克運動會的每個項目都設立了恰好三個獎項：「金希克獎」、「銀希克獎」和「銅希克獎」，分別頒發給該項目獲得前三名的運動選手或隊伍。對於阿森納帝國的運動員而言，如果能得到一面希克獎牌，那可以說是一生無上的榮耀。

除此之外，每屆的奧林希克運動會也都是帝國各大城鎮之間的較量，運動員們除了為自身爭取榮譽，也同時希望能夠為所屬的城鎮爭光。每一屆的奧林希克運動會後，皇帝都會依照各城市的獎牌排名來決定未來四年該城市的運動補助金額，在這屆即將舉行的奧林希克運動會之前，每屆比賽的城鎮排名都是用以下方式決定的：排名按照總金希克獎數量由多至少排序；若金希克獎的數量相同，則再按照總銀希克獎數量由多至少排序；若還是數量相同，則按照總銅希克獎數量由多至少排序；如果三種獎牌數量都一樣，則並列相同名次。(例如：最多積分的城鎮有兩個，則該二城鎮並列第一名，剩下的城鎮由第三名繼續向下排名)

然而，這幾年大家開始覺得這樣子的排名依據不甚公平，一個明顯的例子便是：如果  $A$  城鎮僅拿下 1 面金希克獎牌且沒有其他獎牌，而  $B$  城鎮豪取了共 77 面希克獎牌，但獨缺一面金希克獎牌，則以原本的排名標準， $A$  城鎮會獲得較好的排名，而這顯然不太能被眾人接受。因此，從這屆開始奧林希克運動會的城鎮排名決議將使用以下新制度排序：每面金希克獎牌都價值 90 點積分；每面銀希克獎牌都價值 30 點積分；每面銅希克獎牌都價值 15 點積分。城鎮排名將以總積分做排名，而若積分相同則直接並列名次。

穗香鎮是一個民風淳樸的小鎮，由於阿森納帝國很多著名的運動家都是從這裡出身的，故鎮民們皆以該鎮的體育風氣盛行為榮。果果是一位忠實的體育迷，她對於這個新制度非常感興趣，以直覺而言，她覺得排名的結果將會因此有很大的變動。但實際上是不是如此呢？果果決定使用四年前舉辦的上一屆奧林希克運動會的比賽結果來計算，看看有多少的城鎮會因此得到較好的排名，又有多少的城鎮會因此變成較差的名次。但是因為果果怕手算會出錯，於是決定請你寫一支程式來解決這個問題。

## ■ 輸入檔說明

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一個正整數  $N$ ，代表有在奧林匹克運動會中得獎的城鎮數。接下來有  $N$  行，每一行都有三個非負整數  $g_i, s_i, b_i$  分別代表各個城鎮的總金、銀、銅希克獎牌數。

- $T \leq 200$
- $1 \leq N \leq 500$
- $0 \leq g_i, s_i, b_i \leq 1000$ ，且  $1 \leq g_i + s_i + b_i \leq 2000$ 。

## ■ 輸出檔說明

對於每一筆測試資料，請輸出兩個整數，中間以一個空白隔開。

第一個整數請輸出「因此得到較好的排名」的城鎮數；第二個整數請輸出「因此變成較差的名次」的城鎮數。

## ■ 範例輸入

```
2
3
10 8 5
9 4 7
9 2 23
5
1 0 3
0 0 512
0 0 18
0 14 16
0 14 16
```



## ■ 範例輸出

```
1 1
2 1
```

## ■ 範例說明

- 第一筆範例輸入中原先的城鎮排名依序為 1, 2, 3，新制度下的排名則依序為 1, 3, 2。
- 第二筆範例輸入中原先的城鎮排名依序為 1, 4, 5, 2, 2，新制度下的排名則依序為 5, 1, 4, 2, 2。

本頁留白。

## 題目 D

### 寧寧發糖果

執行時間限制: 5 秒

寧寧是一個非常喜歡小朋友的人。實際上，由於她強大的親和力，許多小朋友們也會自然地想要靠近她。為了能夠讓小朋友們更加地開心，寧寧總是會在身上準備一些糖果，在小朋友靠過來時寧寧就會將糖果發給他們。

然而隨著寧寧遇過的小朋友漸多，寧寧發現其實有很多的小朋友並不能被一顆糖果給滿足，因此寧寧決定將糖果分成一小袋一小袋帶出門，並一次發給小朋友一袋。

寧寧已經買好了  $N$  包糖果，為了方便起見，寧寧決定將每包糖果個別分成若干等分來裝袋。在分裝的過程中，寧寧卻發現了一件驚人的事情：每包糖果可能分出的袋數似乎不太一樣！

譬如說如果一包糖果有 20 顆，那它只可能分成 1, 2, 4, 5, 10, 20 六種可能的袋數，而如果一包糖果有 25 顆，那它就只可能分成 1, 5, 25 三種可能的袋數。

寧寧非常好奇，如果不考慮直接被裝成一袋的可能，那麼是否可能有兩包糖果被分成相同的袋數呢？

#### ■ 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一個正整數  $N$ ，代表寧寧買了幾包糖果。第二行則包含  $N$  個正整數  $S_i$ ，代表那  $N$  包糖果分別有幾顆。

- $T \leq 20$
- $1 \leq N \leq 77777$
- $1 \leq S_i \leq 1000000$

## ■ 輸出說明

對於每一筆測試資料，若是有可能有兩包糖果被分成一樣的袋數，請輸出一行「Yes」，否則請輸出「No」於一行。(不含引號)

## ■ 範例輸入

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

## ■ 範例輸出

```
Yes
No
Yes
```

## ■ 範例說明

- 第一筆範例輸入中，第一包跟第三包都可能分成兩袋。
- 第三筆範例輸入中，前兩包都可能分成三袋。

## 題目 E

### 小可魚轉轉轉

執行時間限制: 2 秒

有一天，小可魚掉進一個奇怪的空間。小可魚要回到原本的世界只有一個方法，就是要先算出他現在站的位置，然後才能找到出口。

而你現在正透過一個螢幕觀察小可魚，發現小可魚正在亂走。小可魚身上有個移動感測器，你可以獲得目前小可魚正在往前後左右哪個方向行走。另外小可魚身上也有個空間感測器，可以偵測到目前空間的變化。

你收到的訊號會有六種，且其中  $k$  為參數，表示該操作重複的次數。

1.  $r$ ：整個世界的坐標系轉  $k$  次，每次順時針轉 90 度。
2.  $l$ ：整個世界的坐標系轉  $k$  次，每次逆時針轉 90 度。
3.  $w$ ：小可魚往前方走  $k$  個單位
4.  $a$ ：小可魚往左方走  $k$  個單位
5.  $d$ ：小可魚往右方走  $k$  個單位
6.  $s$ ：小可魚往後方走  $k$  個單位

注意，小可魚自始至終面向的方向都不會改變，只有世界會旋轉。

為了方便起見，我們假設這個空間是個二維平面。而一開始小可魚在二維平面的中心，也就是是原點  $(0, 0)$ ，且面向  $y+$  的方向 (右方是  $x+$ )。

你現在拿到感測器的數據，總共有  $N$  筆訊號，已經按照發生的先後順序排列。請幫忙算出小可魚最後的位置，讓他能回到原本的世界。

## ■ 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。

每一組測試資料的第一行有個正整數  $N$ ，代表訊號的筆數。接下來有  $N$  行，每行有一個字元和一個整數  $s_i, k_i$ ，分別表示該訊號和發生次數。

- $T \leq 20$
- $1 \leq N \leq 100000$
- $1 \leq k_i \leq 100000$
- $s_i$  只會是 'r'、'l'、'w'、'a'、'd'、's' 其中之一。

## ■ 輸出說明

對於每一筆測試資料請輸出一行，包含兩個數字。第一個數字為  $x$  的座標值，第二個為  $y$  的座標值，中間以一個空白隔開。

## ■ 範例輸入

```
2
4
d 3
r 1
d 2
w 1
7
s 1
l 1
a 1
l 3
w 1
l 1
d 1
```

## ■ 範例輸出

```
2 2
0 0
```

## ■ 範例說明

- 第一筆測試資料移動路徑為：

1. d 3:  $\rightarrow (3, 0)$
2. r 1: 整個世界順時針旋轉 90 度，所以目前小可魚面向  $x-$
3. d 2:  $\rightarrow (3, 2)$
4. w 1:  $\rightarrow (2, 2)$

- 第二筆測試資料移動路徑為：

1. s 1:  $\rightarrow (0, -1)$
2. l 1: 整個世界逆時針旋轉 90 度，所以目前小可魚面向  $x+$
3. a 1:  $\rightarrow (0, 0)$
4. l 3: 整個世界逆時針旋轉 90 度 3 次，所以目前小可魚面向  $y+$
5. w 1:  $\rightarrow (0, 1)$
6. l 1: 整個世界逆時針旋轉 90 度，所以目前小可魚面向  $x+$
7. d 1:  $\rightarrow (0, 0)$

本頁留白。



## 題目 F

### 森林小學運動會

執行時間限制: 1 秒

森林小學一年一度的運動會開始了。

跳遠一直都是運動會缺少不可的項目。然而，今年的趣味競賽，推出了嶄新的項目。新項目改自以往的跳遠比賽，改稱之為「策略跳遠」。

策略跳遠的規則是這樣的。

- 比賽場地在一塊平地，可以視為一個二維平面。
- 起點在  $(0, 0)$ ，終點在  $(x, 0)$ 。選手必須正確的落在終點上才算抵達終點。
- 所有選手每次跳躍的距離都必須符合一個指定的規律。這個規律可以用一個短數列  $a_1, a_2, \dots, a_n$  來表示。如果違規，則必須回到起點重新來過，跳躍的規律也必須重置。舉例來說，如果數列  $a = \langle 2, 5 \rangle$ ，選手每次跳躍的距離就必須是  $2, 5, 2, 5, 2, 5, 2, 5 \dots$ 。
- 本比賽採計時制，根據抵達終點的時間來計算。
- 比賽前四小時會公佈比賽所規定的規律，以及終點的位置。

策略跳遠剛公佈的時候受到大家的歡迎，但在大家開始深入研究這個項目後，才發現看似簡單的規則其實暗藏不少陷阱。例如光是要跳指定的長度，就已經非常困難。動物們經過反覆地練習，才勉強能夠按照該規律連續地跳躍。

接下來大家就發現，這個遊戲的策略成分很重，畢竟每個選手可以自由的選擇要跳躍的方向，只要跳躍的距離符合規定即可。

距離比賽只剩下四個小時，大會剛公佈指定的規律。一時之間，同學們幾乎都同時消失無蹤。估計是躲在某個角落擬定策略，忙得很。

教室空無一人，你默默的趴在窗頭，遠望著操場。雖然身為一隻兔子，可是運動神經不太發達，導致運動會幾乎沒有你發揮的空間。

「嗨！」身後的門被推開，嚇了你一跳。你回頭一看，大批的兔朋友都擠在了門口。

「你……想參與嗎？」為首的一個高壯的兔子站了出來，迫切地看著你。

原來，雖然你的運動能力不佳，但數理能力可是樂勝其他的兔子，於是今天參賽的兔子們找上了你，今日運動會終於有你發揮的舞台。你下定了決心，絕對要讓兔族的榮耀繼續在運動場上發揚下去。

於是你被給予了規律表  $a_1, a_2, \dots, a_n$  和終點的位置  $(x, 0)$ ，你要求出最少可以在幾步時跳到終點。可是你發現規律表有點長，因此你想要借助電腦來幫你規劃策略。

噢，對了！如果使用該規律無論如何都無法抵達終點，你希望你的程式輸出  $-1$ 。這時候你要趕快跟大會回報這件事了，避免整個比賽無法進行。

## ■ 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。

每一組測試資料有兩行。第一行有兩個用空白隔開的整數  $n, x$ ，分別代表數列的長度和終點位於  $(x, 0)$ 。第二行則是規律表， $n$  個以空白隔開的整數  $a_i$ 。

- $T \leq 217$
- $1 \leq n \leq 50$
- $-10^9 \leq x \leq 10^9$
- $1 \leq a_i \leq 10^9$

## ■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示在最佳策略之下，要跳幾步才有辦法落在終點上。如果無論如何都無法跳到終點上，請輸出  $-1$ 。

## ■ 範例輸入

```
6
10 15
1 2 3 4 5 6 7 8 9 10
1 5
5
1 1
10
7 -10
2 3 4 500 6 7 8
1 -1000000000
1
1 0
19911120
```

## ■ 範例輸出

```
5
1
2
11
1000000000
0
```

## ■ 範例說明

- 在第一筆測試資料可以發現只跳 4 步是不夠遠的。5 步的話則可以這樣跳： $(0, 0) \rightarrow (1, 0) \rightarrow (3, 0) \rightarrow (6, 0) \rightarrow (10, 0) \rightarrow (15, 0)$ 。
- 在第二筆測試資料只跳 1 步就夠了。 $(0, 0)$  到  $(5, 0)$  的距離剛好是 5。
- 在第三筆測試資料則只需要跳 2 步。一種策略是： $(0, 0) \rightarrow (0.5, \sqrt{10^2 - 0.5^2}) \rightarrow (0, 1)$ 。

本頁留白。

## 題目 G

### 卡恩買飲料

執行時間限制: 3 秒

卡恩最喜歡喝飲料了！一日不喝飲料便覺得面目可憎。上課喝，下課也喝；午餐喝，晚餐也喝。傳說中卡恩版本的生命三要素便是：「飲料、飲料、飲料」。喜歡的程度簡直跟胖胖天喜歡吃大薯的程度不相上下。

然而每天都要買飲料的他，常常為了錢包裡堆積如山的零錢而感到苦惱。為了減輕錢包的重量，他在買飲料的時候會盡可能付出最多的硬幣和紙鈔，而且總是會付剛好的金額以避免麻煩的找零。如果沒辦法湊出剛好的金額，表示當下跟飲料沒有緣分，卡恩就會黯然離去。可是他實在太有錢又太常喝飲料又太可愛了，因此希望聰明的你可以寫個程式幫幫他。

#### ■ 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。

每一組測試資料包含一行八個整數  $p, c_1, c_5, c_{10}, c_{50}, c_{100}, c_{500}, c_{1000}$ ，依序代表飲料的價錢以及卡恩錢包中一元、五元、十元、五十元、一百元、五百元、一千元的硬幣和紙鈔數量。

- $1 \leq T \leq 100000$
- $0 \leq p \leq 2000000$
- $0 \leq c_1, c_5, c_{10}, c_{50}, c_{100}, c_{500}, c_{1000} \leq 1000$

#### ■ 輸出說明

對於每一筆測試資料請輸出一行，包含一個整數表示買  $p$  元的飲料最多可以付出多少個硬幣和紙鈔。如果沒辦法恰好湊出  $p$  元的話請輸出「OAQ」（不含引號）。

## ■ 範例輸入

```
4
20 5 1 4 5 1 4 0
65 5 1 4 5 1 4 0
514 0 0 0 0 0 0 999
1000 1000 0 0 0 0 0 1
```

## ■ 範例輸出

```
7
7
OAQ
1000
```

## ■ 範例說明

- 第一筆測試資料中，卡恩會付五個一元、一個五元跟一個十元。
- 第一筆測試資料中，卡恩會付五個一元、一個十元跟一個五十元。
- 第三筆測試資料中，雖然卡恩非常富有，但因為湊不出 514 元，只好黯然離去。
- 第四筆測試資料中，雖然卡恩有千元鈔，但還是會付一千個一元，飲料店店員表示難過。