

2012 網際網路程式設計全國大賽 模擬競賽

- 題目：本次比賽共六題（含本封面共 16 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）。
- 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`。
- `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	北極熊大遷徙	5 秒
題目 B	南極企鵝大遷徙	5 秒
題目 C	賓果遊戲	5 秒
題目 D	三角形金磚	5 秒
題目 E	田忌賽馬	5 秒
題目 F	河川改道之術	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, a, b, i;
    scanf("%d", &cases);
    for(i = 0;i < cases;i++){
        scanf("%d %d", &a, &b);
        printf("%d\n", a+b);
    }
    return 0;
}
```

C++ 程式範例：

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

題目 A

北極熊大遷徙

執行時間限制: 5 秒

因為全球暖化的關係，北極各處的浮冰正在慢慢融化之中。部份北極熊所在的浮冰已經融化到不堪居住的程度，於是這些北極熊興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 隻北極熊，牠們現在打算遷徙到有 b 隻北極熊居住的浮冰 B 。你要回答的是：經過北極熊大遷徙以後，浮冰 B 上總共會有多少隻北極熊。

■ 輸入檔說明

輸入檔第一行有一個數字 T 代表總共有幾筆測試資料。

每一筆測試資料有兩個整數 a 和 b ($1 \leq a, b \leq 50$)，代表有 a 隻北極熊即將從浮冰 A 遷徙到原本有 b 隻北極熊的浮冰 B 。

■ 輸出檔說明

針對每一筆測試資料，輸出浮冰 B 上最後會有多少隻北極熊。

■ 範例輸入

```
3
24 47
33 20
11 30
```

■ 範例輸出

```
71
53
41
```

本頁留白。

題目 B

南極企鵝大遷徙

執行時間限制: 5 秒

因為全球暖化的關係，南極各處的浮冰正在慢慢融化之中。部份企鵝所在的浮冰已經融化到不堪居住的程度，於是這些企鵝興起遷徙的念頭。

已經融化到不堪居住的浮冰 A 上有 a 公斤的企鵝，牠們現在打算遷徙到有 b 公斤的企鵝居住的浮冰 B 。你要回答的是：經過企鵝大遷徙以後，浮冰 B 上總共會有多少公斤的企鵝。

■ 輸入檔說明

輸入檔第一行有一個數字 T 代表總共有幾筆測試資料。

每一筆測試資料有兩個浮點數 a 和 b ($1 \leq a, b \leq 50$)，代表有 a 公斤的企鵝即將從浮冰 A 遷徙到原本有 b 公斤重的企鵝的浮冰 B 。

■ 輸出檔說明

針對每一筆測試資料，輸出浮冰 B 上最後會有多少公斤的企鵝，四捨五入至小數第二位。

■ 範例輸入

```
3
24.23 47.33
33.43 20.92
11.38 30.74
```

■ 範例輸出

```
71.56
54.35
42.12
```

本頁留白。

題目 C 賓果遊戲

執行時間限制: 5 秒

在整理櫃子的時候，創創和守守發現了一疊賓果卡（5 x 5 的方格紙上不按順序地印著 01 到 25 的數字），所以他們還沒整理完櫃子就開始玩起賓果遊戲來了。

<u>03</u>	<u>21</u>	05	<u>15</u>	<u>08</u>
14	<u>10</u>	17	<u>23</u>	04
<u>18</u>	<u>06</u>	<u>22</u>	<u>12</u>	<u>09</u>
<u>24</u>	<u>20</u>	<u>13</u>	07	<u>16</u>
<u>11</u>	<u>25</u>	01	<u>19</u>	02

22	<u>12</u>	<u>15</u>	<u>03</u>	<u>16</u>
<u>11</u>	<u>23</u>	<u>20</u>	14	<u>19</u>
02	17	09	05	07
<u>24</u>	04	<u>10</u>	<u>21</u>	01
<u>06</u>	<u>13</u>	<u>18</u>	<u>25</u>	<u>08</u>

04 14 07 22 02 01 09 17 05 18 19 12 08 21 15 06 10 16 03 24 20 25 13 23 11

創創和守守玩賓果遊戲的方式可能和我們有點不一樣：他們會一人先抽一張賓果卡，再使用從網路上找到的某個程式輸出一組數列（這個數列包含 25 個數字，每個數字都在 01 到 25 之間，已經出現過的數字就不會再重複出現），然後按照數列裡數字出現的順序把賓果卡上的數字劃掉，直到其中一人的賓果卡上出現一條長度為 5 的直線、橫線或斜線為止，達成連線的人就是贏家；如果兩個人同時達成連線，則視為平手。

仔細想想，每次都要尋找數字、再把它劃掉、還有判斷是否有人勝利，這一連串的動作其實冗長又重複，所以創創和守守現在只想知道是誰贏了遊戲，而且他們還要回去整理櫃子呢！你可以在他們整理完櫃子之前幫忙寫一個程式，判斷誰抽到的賓果卡會贏得遊戲嗎？

■ 輸入檔說明

包含多組測試資料，第一列有一個整數 T 表示接下來有 T 組測試資料。

每組測試資料包含兩張賓果卡和一組數列：前 10 列每列有 5 個數字，第 1 列到第 5 列代表第 1 張賓果卡，第 6 到第 10 列代表第 2 張賓果卡；第 11 列是由 25 個數字所組成的數列，每個數字都在 01 到 25 之間，出現過的數字不會重複出現。

■ 輸出檔說明

對於每組測試資料輸出哪張賓果卡會贏得遊戲，”1” 代表第 1 張，”2” 代表第 2 張，”0” 代表平手。

■ 範例輸入

```
2
03 21 05 15 08
14 10 17 23 04
18 06 22 12 09
24 20 13 07 16
11 25 01 19 02
22 12 15 03 16
11 23 20 14 19
02 17 09 05 07
24 04 10 21 01
06 13 18 25 08
04 14 07 22 02 01 09 17 05 18 19 12 08 21 15 06 10 16 03 24 20 25
13 23 11
03 21 05 15 08
14 10 17 23 04
18 06 22 12 09
24 20 13 07 16
11 25 01 19 02
22 12 15 03 16
11 23 20 14 19
02 17 09 05 07
24 04 10 21 01
06 13 18 25 08
04 14 07 22 02 01 09 17 13 05 18 19 12 08 21 15 06 10 16 03 24 20
25 23 11
```

■ 範例輸出

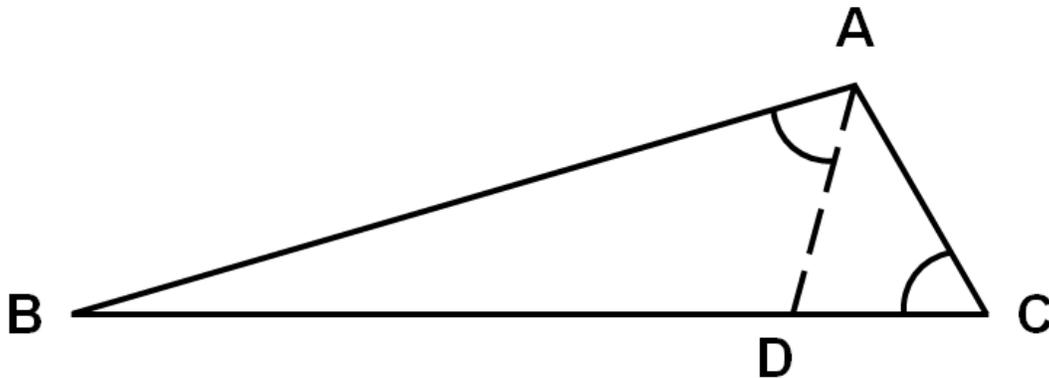
```
2
0
```

題目 D 三角形金磚

執行時間限制: 5 秒

塔克拉瑪干沙漠的別名是「死亡之海」，身為考古探險隊隊長的亞特想要聘請一位嚮導，幫助考古探險隊展開在「死亡之海」的旅程。他上次在塔克拉瑪干沙漠考察的時候認識了一個嚮導，這個嚮導自稱能夠帶領亞特前往傳說中的沙漠古城，條件是從沙漠古城帶出來的部份寶藏要分給嚮導。

根據嚮導所言，這個沙漠古城中的寶藏都是厚度一致的鈍角三角形金磚，而且兩個銳角必定不相等。為了方便起見，我們將金磚的鈍角稱為 A 、比較小的銳角稱為 B 、比較大的銳角稱為 C ，再把金磚從鈍角 A 分為兩個三角形，並且將新的頂點稱為 D ，分割金磚的條件是讓 $\angle BAD = \angle ACD$ 。嚮導要求的報酬就是和 $\triangle ACD$ 金磚等價的金錢或寶物。



雖然亞特並不是那麼相信嚮導口中的沙漠古城，但是亞特想要探險的地方風險實在太大，沒有其他嚮導願意陪同，嚮導人選因此決定。因為亞特不太相信會從這趟旅程帶回傳說中的寶物，所以他不想花時間計算嚮導要求的報酬佔了所有寶物的多少比例，你能代替他算出來嗎？

■ 輸入檔說明

包含多組測試資料，第一列有一個整數 T 表示接下來有 T 組測試資料。

每組測試資料包含兩個整數，分別代表 \overline{AB} 的長度和 \overline{BC} 的長度。

■ 輸出檔說明

對於每組測試資料輸出 $\triangle ACD$ 佔 $\triangle ABC$ 的比例，四捨五入至小數點第三位。

■ 範例輸入

```
3  
4 6  
12 14  
24 26
```

■ 範例輸出

```
0.556  
0.265  
0.148
```

題目 E 田忌賽馬

執行時間限制: 5 秒

齊使者如梁，孫臏以刑徒陰見，說齊使。齊使以為奇，竊載與之齊。齊將田忌善而客待之。忌數與齊諸公子馳逐重射。孫子見其馬足不甚相遠，馬有上、中、下、輩。於是孫子謂田忌曰：「君弟重射，臣能令君勝。」田忌信然之，與王及諸公子逐射千金。及臨質，孫子曰：「今以君之下駟與彼上駟，取君上駟與彼中駟，取君中駟與彼下駟。」既馳三輩畢，而田忌一不勝而再勝，卒得王千金。於是忌進孫子於威王。威王問兵法，遂以為師。

--- 『史記。孫子吳起列傳第五』

千年以前，孫臏靠著過人的智謀，巧妙的調整比賽順序，讓三戰皆墨的田忌翻身成兩勝一敗的贏家，也為自己贏得尊敬和重用。千年以後的今日，賽馬依然是熱門的活動，不過今天你要面對的是更困難的問題。

你和對手各有 N 匹馬，要進行 N 場比賽。一匹馬只限出場一次，同場比賽中速度較快的馬獲勝。若兩匹馬速度一樣，則算平手。

你可以決定你的馬匹的出場順序；而你的對手，就如同齊王，會在第一場比賽出速度最快的馬，第二場出次快的馬， \dots ，第 N 場出速度最慢的馬。

除此之外，你還可以決定比賽的時間，全部 N 場比賽都會在你選的這一天進行。在比賽之前，勤勞的你每天都會訓練你的每一匹馬；而你的對手自我感覺非常良好，因此不會訓練他的馬。每一匹馬的素質不同，我們用 a_i 來表示第 i 匹馬的速度，用 b_i 來表示第 i 匹馬的成長率。經過 m 天的訓練，你的第 i 匹馬在第 $m + 1$ 天的速度就會是 $a_i + m \cdot b_i$ 。對手的第 j 匹馬在每一天的速度都是 a_j 。

現在你有你和對手共 $2N$ 匹馬的資料，請決定訓練的天數 M ，使得在第 $M + 1$ 天比賽的時候，你有一個出場順序可以贏得 N 場比賽中的至少 K 場（不包含平手）。

■ 輸入檔說明

第一行有一個整數 T ($T \leq 100$)，代表接下來有幾組測試資料。

每一組測試資料的第一行有兩個數字， N 和 K 。

接下來 N 行是你的馬匹的資料，每一行有兩個整數， a_i 和 b_i ，代表馬匹的速度和成長率。

再下來 N 行是對手的馬匹的資料，每一行有一個整數 a_j ，代表馬匹的速度。 $(1 \leq K \leq N \leq 10000, 0 \leq a_i, a_j \leq 100000000, 0 \leq b_i \leq 100)$

■ 輸出檔說明

對每筆測試資料輸出一個非負整數 M ，代表訓練 M 天後在第 $M + 1$ 天舉行賽馬你可以贏得至少 K 場。

如果有不只一個 M 滿足條件，請輸出最小的 M 。如果沒有任何 M 滿足條件，請輸出 -1 。

■ 範例輸入

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

■ 範例輸出

```
1
-1
```

題目 F 河川改道之術

執行時間限制: 5 秒

歪歪國的河川都是歪歪的。歪歪國的人們每次出外旅行的時候總是會讓其他人留下深刻印象，其他人會說：啊哈，我會記得你們這些歪國人。

不過，歪歪國的人民很自豪地說，我們是歪中有序。儘管所有的河川都是歪歪扭扭的，但國土可是方方正正，而且可以分成 $M \times N$ 個小方形區域，每一個小方形區域都恰有兩條河道經過。更有趣的是，如果只看這兩條河道經過的方式，會發現其實只有兩種形態：



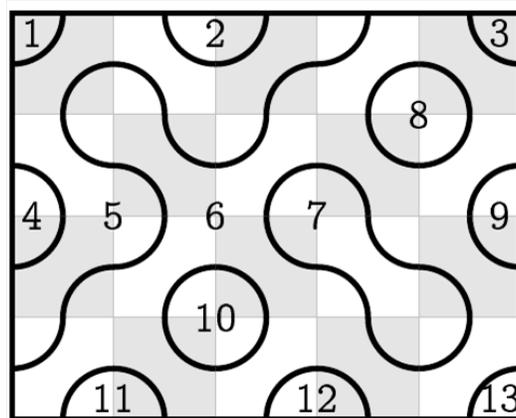
(a) A 形態



(b) B 形態

而實際的國土例子可能是：

ABAAB
BBABA
AABBB
ABABA



(a) 國土的顯示範例

(b) 範例對應的實際樣貌

歪歪國由於河道眾多，所有土地可能會被河道切成了很多塊。自古以來，歪歪國便依照國土分塊的狀況制定行政區域，如上圖右邊的數字編號。換句話說，只要國土上任兩點之間有一條完全落在國土上，而且不跨越任何河道的路線連結，它們就會屬於同一個行政區域。

不過，由於當前的行政區域過於眾多，歪歪國想要改變河道行進的方式，使得以相同定義新劃分的行政區域總數降至最低。

史歪哩 (Swyly) 和科西 (Kshi) 是歪歪國裡面著名的魔法師，他們被歪歪國的水利大臣球球 (Kel-kel) 和皮皮 (Pi-pi) 聘請來整治歪歪國的國土。經過了許久的研究，為了不打破歪歪國的國土風格，他們決定分次施以控制力 (Ctrl-Z)，每次可將一塊 A 形態的小方格上面的水道轉換成 B 形態的小方格，或將 B 形態的小方格轉換成 A 形態的小方格。

「這，究竟給不給力啊？」歪歪國的審查大臣文文 (Vim-vim) 歪著頭，看著魔法師提出的國土修正方案。

為了省力，史歪哩和科西打算用最少的施力次數，使得新劃分出來的行政區域數量降至最低。不僅如此，他們還希望找出盡可能美觀的方式，所謂盡可能地美觀，就是將修正後的國土每個小方格的形態寫出來以後一列一列接成一個字串時，是所有可能施以最少次控制力的做法當中，字典順序最小的一個方案。

時候不多了，趕快幫兩位魔法師找出整治歪歪國的最佳方案吧！

■ 輸入檔說明

輸入的第一列有一個正整數 T ($T \leq 100$) 表示測試資料的數量。每一筆測試資料的第一列有兩個正整數 M, N ($1 \leq M, N \leq 500$) 表示歪歪國的國土長寬，接下來的 M 列每一列有一個長度為 N 的字串，第 i 列的第 j 個字元代表的就是 (i, j) 這一格的河道分布形態。 ($1 \leq i \leq M; 1 \leq j \leq N$)

■ 輸出檔說明

對於每一筆測試資料，第一列請輸出兩個以一個空白隔開的非負整數 R, K ，其中 R 表示所能夠達到的最低行政區域數量，而 K 表示至少要進行幾次名為控制力的河川改道之術。由於直接輸出最佳方案資料量太過巨大，我們採用下面的方式將答案雜湊後輸出，此函式與解題無關，純粹是方便輸出使用：

我們首先定義答案矩陣 A 為：

以 M 列每一列有 N 個字元用以表示盡可能美觀的最佳方案。若第 (i, j) 格不需要施力，以點 '.' 表示；若需要施力，則以星號 '*' 表示。

現在我們把答案矩陣 A 的每一列依序接成一個長度恰好是 $M \times N$ 的字串 S ，然後利用以下的函式跑出我們要的答案：

```
unsigned int get_hash(const char *S) {
    unsigned int i, h = 1315423911;
    for (i = 0; *S; S++, i++)
        h ^= ((h << 5) + (*S) + (h >> 2));
    h &= 0x7FFFFFFF;
    return h;
}
```

請你將這個函式回傳的值在每筆測資的第二列輸出。

■ 範例輸入

```
3
3 5
AAAAA
AAAAA
AAAAA
2 2
AB
BA
4 5
ABAAB
BBABA
AABBB
ABABA
```

■ 範例輸出

```
9 0
1804696164
5 1
1131664733
10 3
91716038
```

■ 範例輸出說明

第一筆範例測試資料的答案矩陣長得像這樣：

```
.....  
.....  
.....
```

第二筆範例測試資料的答案矩陣長得像這樣：

```
. *  
..
```

第三筆範例測試資料的答案矩陣長得像這樣：

```
.....*  
.....*.  
...*..  
.....
```