

102 學年度高級中學資訊學科能力競賽決賽

上機程式設計競賽注意事項

- 一、本競賽採電腦線上自動評分，程式必須依規定上傳至評分主機。請**嚴格遵守**每一題目所規定之**原始程式、輸入檔、輸出檔等檔名及格式**。若未遵守，該題將可以 0 分計算。
- 二、本次競賽採取全面回饋機制，程式上傳至評分主機後，將自動編譯並進行測試。視等待評分題數多寡，該題測試結果及該題得分數將可於短時間內得知。程式可重複上傳及評分，但同一題**兩次上傳之間必須間隔二分鐘**以上。每題最終分數以該題所有單次評分結果之最高分計算。
- 三、本次競賽**每一題皆有不同難易度**的測試資料，詳細配分及限制條件請詳各題題目說明。程式執行時，針對每組測試資料**執行時間以 2 秒為限**（以評分主機執行時間為準）。執行時間超過 2 秒者視同未完成，該組測試資料得分將以 0 分計算。每題可使用記憶體空間，以 512MB 為限。
- 四、本次競賽學生**需上傳原始程式碼** (.c, .cpp, .pas)，輸入資料則自“input.txt”檔讀取，計算結果需輸出至“output.txt”檔。注意：**所有檔案的讀寫都在執行檔的工作目錄下進行**，請勿自行增修輸入檔的檔名或路徑，若因此造成評分程式無法評分，該次評分結果將以 0 分計算。
- 五、線上評分系統使用的編譯器與試場電腦所安裝的編譯器版本不一定一致，**以評分系統為主**。本次競賽題目**無需使用超長整數** (long long) 變數，請勿使用 long long 宣告變數。若因為使用該資料型態造成程式編譯或執行錯誤，導致評分程式無法正確評分，該次評分結果將以 0 分計算。
- 六、請用承辦單位提供之**個人電腦內硬碟或網路儲存空間**備份原始程式碼。若由於任何原因需更換電腦時，僅能將備份之原始程式碼複製至新電腦，或下載已上傳至評分主機的程式碼。
- 七、對考題有任何疑義，請以評分系統之提問功能 (Communication) 提問（請參考評分系統之使用手冊）或填寫「提問單」交付監試人員轉送命題委員提出問題。**監試人員不負責解答任何有關試題的問題**。
- 八、競賽時間結束後，參賽選手應留在位置上等待評審前來共同確認評審結果。
- 九、試場守則：
 - 甲. 試場內不得**與其他選手討論、上傳與解題無關檔案**或**惡意干擾線上自動評分系統運作**，承辦單位得依據違規狀況予以警告或**取消比賽資格**。
 - 乙. 考試進行期間，選手可至試場外飲水、用餐、如廁，但不得**與他人研討試題**，違者得**取消比賽資格**。
 - 丙. 離開試場，返回試場時，均須於試場入口填寫「離場休息登記表」。
 - 丁. 離開試場時間，除午餐為 1 小時外，皆不可超過**10 分鐘**。
 - 戊. 考試時間用餐點、如廁均不可**離開工程三館三樓電梯北側區域**。
 - 己. 11:30~12:30，三樓電梯門口將供應午餐便當。
 - 庚. 午餐便當如未能及時食用，請於 14:30 前至三樓電梯門口領取，可使用微波爐加熱服務。
 - 辛. 參賽中途如遇緊急事件（如停電等），因應措施得依據命題及評審委員會決議處理。

(本頁空白)

1. 平衡數列

時間限制:2 秒

問題敘述:

一個長度至少為 2 的正整數列 $A = (a_1, a_2, \dots, a_n)$ ，如果存在某個 $a_i (1 \leq i \leq n)$ ，使得我們把整個數列視為一個翹翹板，並以 a_i 所在的位置作為支點撐起來時，支點兩端的力矩值相同（將數列中的每一個數字視為一個重量，且相鄰兩數字間的距離為 1），則我們稱此數列為**平衡數列**。一個平衡數列的例子如下圖 1 所示。

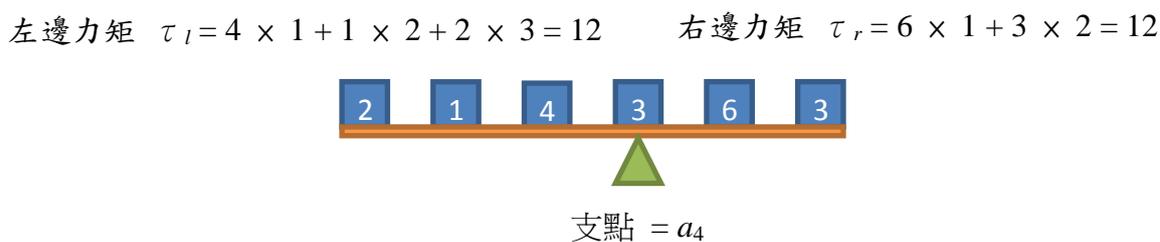


圖 1. 一個平衡數列 $A = (2, 1, 4, 3, 6, 3)$

如果 A 不是平衡數列，我們可以找出一組 a_i 和 a_{i+1} ，使得當我們把支點放置於 a_i 時，支點右側的力矩值較大；而一旦支點移動至 a_{i+1} ，會讓支點左側的力矩值較大，在這種情況下，我們將 a_i 與 a_{i+1} 稱為這個非平衡數列的**平衡交錯點**。一個非平衡數列的例子如下圖 2 所示。在這個例子中，交錯點為 a_3 和 a_4 。

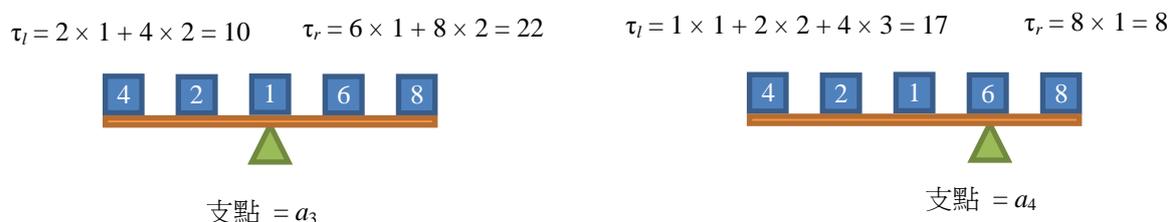


圖 2. 一個非平衡數列 $A = (4, 2, 1, 6, 8)$

對於給定的一個正整數數列 A ，我們可以由左到右嘗試將數列中的每一個元素當作支點並計算其兩側的力矩值，來判斷它是否為一個平衡數列。一個具體的例子如下圖 3 所示。

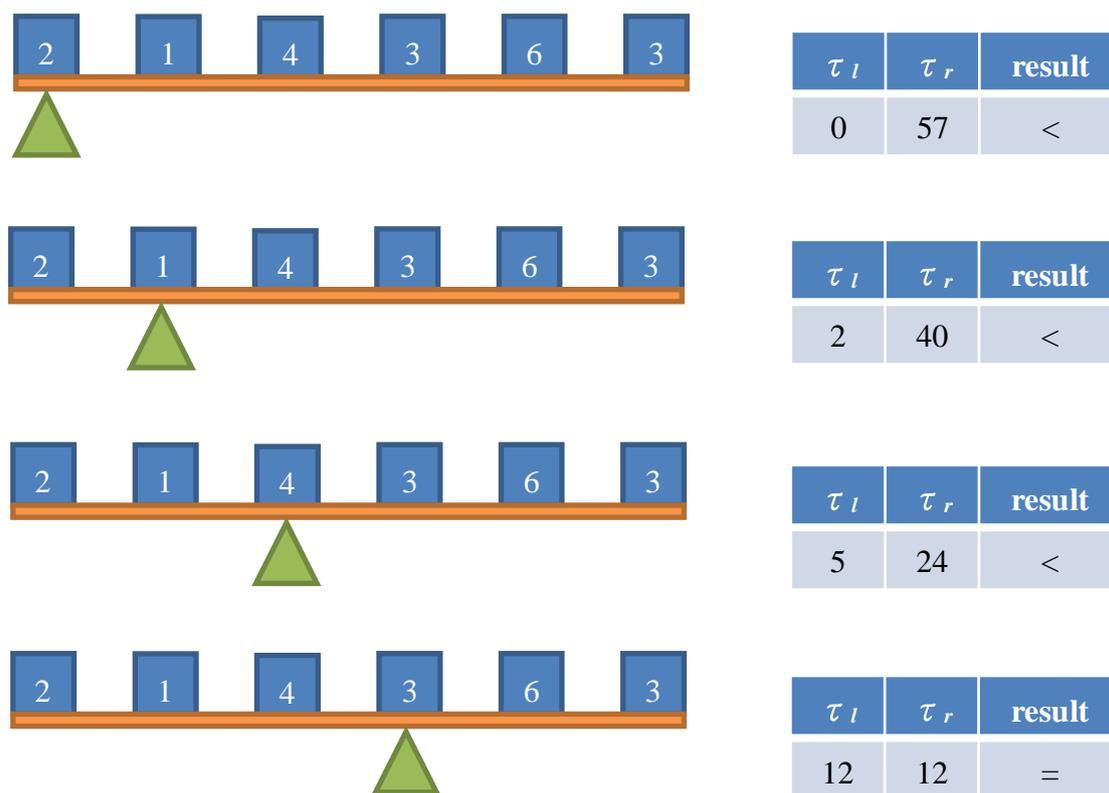


圖 3. 檢驗各個位置找出支點的過程

在圖 3 中，當檢查 a_4 時，我們發現兩邊的力矩 $\tau_l = \tau_r$ ，所以 $A = (2, 1, 4, 3, 6, 3)$ 是一個平衡數列，支點的位置為 4。圖 4 是另外一個例子，過程中並沒有出現兩邊相等的位置，所以 $A = (4, 2, 1, 6, 8)$ 不是一個平衡數列。在這個例子中， τ_l 在以 a_4 當支點時才第一次超過 τ_r ，所以交錯點的位置為 3 和 4。

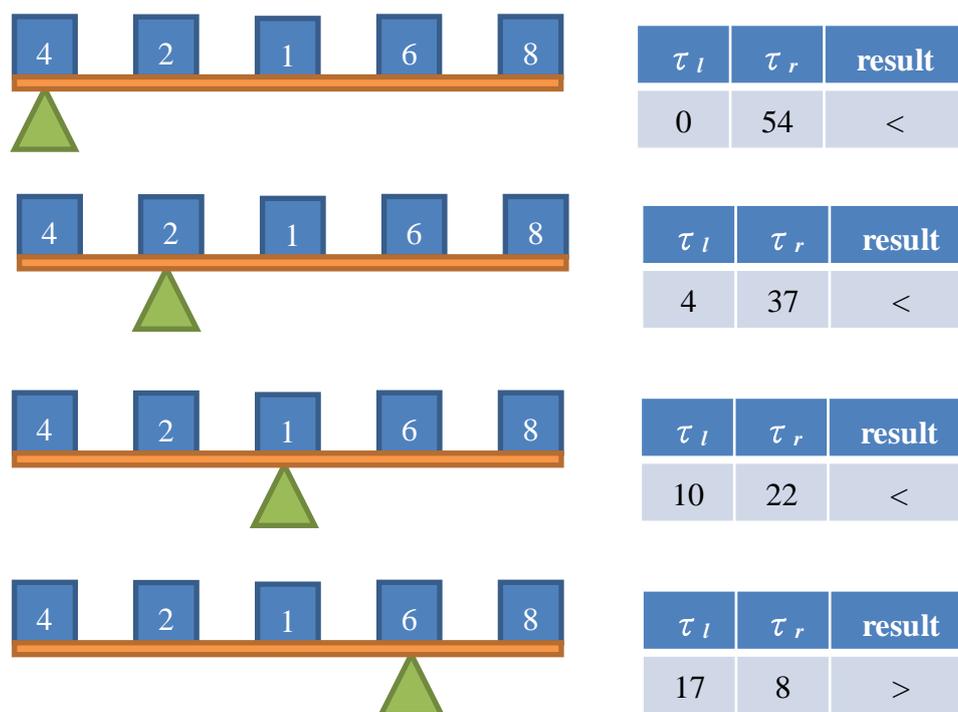


圖 4. 尋找支點時發現無法平衡的過程

當然，我們也可以利用物理的質心公式來快速判斷一個數列是否為平衡數列，並計算支點與交錯點的位置，不過寫程式時要稍微小心留意，才不會犯錯。

輸入說明：

輸入的第一行有一正整數 n ，表示正整數列的長度，其中 $2 \leq n \leq 1000$ 。第二行會有 n 個正整數，依序代表 a_1, a_2, \dots, a_n ，其中 $1 \leq a_i \leq 1000$ ，整數間以一個以上的空白字元隔開。

輸出說明：

輸出由包含一或兩個整數的一行所組成。若輸入的正整數數列為一平衡數列，請輸出支點的位置。否則，輸出兩個交錯點的位置(以單個空白字元隔開，較小者先出現)。

輸入範例 1：

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

輸出範例 1：

```
4
```

輸入範例 2：

```
5  
4 2 1 6 8
```

輸出範例 2：

```
3 4
```

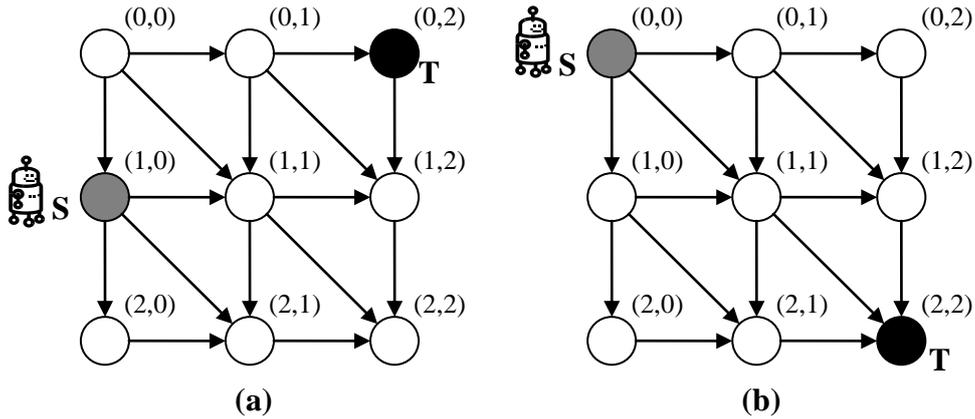
(本頁空白)

2. 機器人走迷宮

限制時間：2 秒

問題敘述：

H 公司發展了一個新型機器人，為了測試這個機器人，H 公司建構了一個矩形的網格迷宮，這個迷宮由 $(M+1) \times (N+1)$ 個位置由整數對 $\{(i, j) \mid 0 \leq i \leq M \text{ and } 0 \leq j \leq N\}$ 所構成。 $(i, j) \rightarrow (i', j')$ 表示有一個單向通道從 (i, j) 通往 (i', j') ，其充分及必要條件為 $i' = i + 1$ 或 $j' = j + 1$ 。圖一表示 $M=2$ 和 $N=2$ 的網格迷宮。



圖一

機器人從位置 S 走到位置 T 代表從 $S(=X_0)$ 出發經過一系列的位置 $X_1, X_2, \dots, X_{k-1}, X_k(=T)$ 到達位置 T ，其中， $X_1 \rightarrow X_2, X_2 \rightarrow X_3, \dots, X_{k-1} \rightarrow X_k$ 皆為單向通道。給定起始位置 S 和終止位置 T ($S \neq T$)，請寫一個程式計算機器人從 S 走到 T 共有幾種不同的走法，如果 S 走不到 T 則輸出 0。注意：兩種 S 到 T 的走法不同代表中間走過的位置個數不同，或是中間走過的位置個數相同但存在某個整數 p 使得兩種走法的第 p 個位置彼此不同。例如：圖一(a)的網格迷宮中，從 $S=(1, 0)$ 到 $T=(0, 2)$ 共有 0 種走法；圖一(b)的網格迷宮中，從 $S=(0, 0)$ 到 $T=(2, 2)$ 共有 13 種走法。如下：

- $(0, 0) \rightarrow (0, 1) \rightarrow (0, 2) \rightarrow (1, 2) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (0, 1) \rightarrow (1, 2) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (1, 2) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 0) \rightarrow (2, 1) \rightarrow (2, 2)$
- $(0, 0) \rightarrow (1, 0) \rightarrow (2, 0) \rightarrow (2, 1) \rightarrow (2, 2)$

規格說明：

(1) $0 \leq M \leq 12$ 。

(2) $0 \leq N \leq 12$ 。

輸入說明：

第一行有 2 個數字，代表 M 值與 N 值，中間以空白隔開。第二行有兩個數字，表示起始位置 S 的 X 座標值和 Y 座標值，中間以空白隔開。第三行有兩個數字，表示終止位置 T 的 X 座標值和 Y 座標值，中間以空白隔開。

輸出說明：

針對所輸入的資料，輸出一個數字代表不同的走法數。每行輸出一個數字。

輸入範例 1：

2 2

0 0

2 2

輸出範例 1：

13

輸入範例 2：

2 2

1 0

0 2

輸出範例 2：

0

3. 路由器轉送量

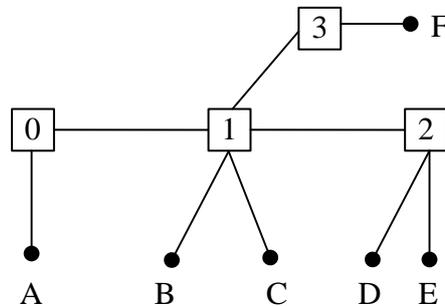
限制時間：2 秒

問題敘述：

電腦網路由一些路由器將電腦相互連結，每一台電腦都會連在一台路由器上，而一台路由器會與若干台路由器相連，兩台電腦傳輸資料時，都需透過路由器的轉送。兩台電腦如果連在同一台路由器上，其傳輸只需經過一次轉送。但如果是接在不同路由器上的電腦要傳輸時，就必須透過一台以上的路由器多次轉送，而網路的機制會確保每一次傳送不會重覆經過相同的路由器。在本題要處理的網路中，路由器係以樹狀結構連結，也就是說任兩台電腦之間均恰有一條路徑相連。

假設每一台電腦到任何一台其他電腦都有相同的一單位傳輸需求量，本題的目的是希望你可以找出各路由器中最高的轉送量。

例如，在下圖中，方塊代表路由器，黑點代表電腦，這個網路中一共有四台路由器（編號由 0 至 3），分別接了一台(A)、兩台(B,C)、兩台(D,E)及一台(F)電腦。



對電腦 A 而言，如果資料傳到 B 或 C 都需經過 0 和 1 這兩台路由器，如果要傳送到 D 就需要經過(0,1,2)三台路由器了；而對 D 而言，傳送到 E 只需要經過路由器 2，如果要送到 B 則需要兩台路由器(2,1)。經過仔細的計算，你會發現路由器 0 的轉送量是 10，其傳送接收端分別是(AB), (AC), (AD), (AE), (AF), (BA), (CA), (DA), (EA), (FA)。另外一種觀察方式是 A 與其他五台電腦中的任何一台的傳輸都需經過 0，因為來回都計算所以轉送需求量是 10。以下我們來計算路由器 1 的轉送量：

- A 出發到其他五台電腦都需要路由器 1。
- B 出發到其他五台電腦都需要路由器 1。
- C 出發到其他五台電腦都需要路由器 1。
- D 出發到 ABCF 四台電腦都需要路由器 1。
- E 出發到 ABCF 四台電腦都需要路由器 1。
- F 出發到其他五台電腦都需要路由器 1。

加總起來路由器 1 的轉送量是 $5+5+5+4+4+5=28$ 。用相同的方式可算出編號 2 是 18；編號 3 是 10。因此本題答案是 28。

輸入說明：

共有若干個網路需要計算，網路個數不超過 5。每一個網路的資料有三行，第一行是一個整數 n 代表有 n 台路由器（編號 $0 \sim n-1$ ）， $0 \leq n \leq 50000$ 。第二行有 n 個不超過 10 的非負整數，依序代表編號 0、編號 1、...編號 $n-1$ 的路由器所接的電腦數。第三行有 $n-1$ 個整數，第 i 個數字 $p(i)$ ， $1 \leq i \leq n-1$ ，是

一個小於 i 的非負整數代表編號 i 路由器與編號 $p(i)$ 路由器之間有網路線相連，每行的兩個數字都以一個空白隔開。當 $n=0$ 時表示測資結束，不需要處理此筆資料。

如果你可以正確處理「路由器連成一條 path」的情形或是「路由器總數比較小」的情形，將有機會得到本題的部分分數。

輸出說明：

依序每行輸出一個網路的最大轉送量。你可以假設答案不會超過 2 的 32 次方。

輸入範例 1：

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

輸出範例 1：

```
18
60
```

輸入範例 2：

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

輸出範例 2：

```
28
24
```

4. 食材爭奪戰

限制時間：2 秒

問題敘述：

小當家是史上最年輕的特級廚師，自從在神前刀工對決裡取得傳說中的七廚具之一 --- 永靈刀之後，他與夥伴七星刀雷恩、鋼棍解師傅、嘟嘟以及四郎走遍各地，開啟與黑暗料理界對抗的旅程。途中經歷了火燒樓麟艦以及敦煌之戰，這次又與黑暗料理界開啟了新的對決。對決的內容是材料庫裡有很多樣的食材，一開始雙方都有一些金錢來爭奪好的食材，每種食材都有美味程度以及所需金錢量。每次雙方輪流選擇食材，每回合至多選一樣(但已被選擇食材不能再被選擇)，若食材被選完即停止，由小當家團先選。小當家團每輪必須用金錢購買所選食材(金錢不足即不能買)，但由於黑暗料理界的行徑，使得他們可以不受金錢限制來取得任何的食材；同時因為黑暗料理界精準的預測，又可以知道每一步怎麼選可以對小當家一行人造成最大的傷害，使得小當家一行人非常的頭痛。在這個時候，小當家從及第師傅口中得知閣下擁有非常聰明的腦袋以及超強的計算能力，因為前來尋求你拯救他們。

為了挽救料理界的未來，閣下是否能幫小當家一行人算出他們最多可以拿到多少美味程度的食材，使得他們能在此情況下做出美好的料理來呢？

簡單例子說明：

假設有 5 種食材，其美味程度為 7、5、4、3、2，而所需金錢剛巧與美味程度相等。另一方面，小當家一行人所擁有的金錢數量為 9。若小當家先取美味程度為 7 的食材，則黑暗料理界只需把美味程度為 2 的食材拿掉，便使小當家無法拿取其餘食材(因金錢不足)；但若小當家起始時先取美味程度為 5 的食材，則無論黑暗料理界如何挑選，小當家必能在下一輪拿到美味程度至少為 3 的食材。在雙方使用最佳策略的情形下，我們可發現小當家能拿到美味程度總計為 8 的食材。

輸入說明：

第一行有一整數 T ($1 \leq T \leq 25$)，代表有 T 組測試資料。接下來依序為 T 組測試資料的內容，每組均以下列格式表示：

- 各組的第一行為一整數 N ($1 \leq N \leq 15$)，代表共有 N 種食材
- 接下來的 N 行分別代表 N 種食材的資訊，每行有兩個整數，以一個空白字元隔開。第一個整數為此食材的美味程度 D ($1 \leq D \leq 1000$)，而第二個整數則為此食材的所需金錢 G ($1 \leq G \leq 200$)
- 最後一行為一整數 M ($0 \leq M \leq 200$)，代表小當家一行人所擁有的金錢數量

輸出說明：

依序對每一組測試資料，輸出一行結果(即共 T 行)。每行結果為一正整數，代表在雙方使用最佳策略的情形下，小當家一行人最多能拿到的食材美味程度總和。

輸入範例 1：

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

輸出範例 1：

8

輸入範例 2：

2
5
8 6
5 5
5 5
4 4
2 2
9
5
8 6
5 5
5 5
4 4
2 2
22

輸出範例 2：

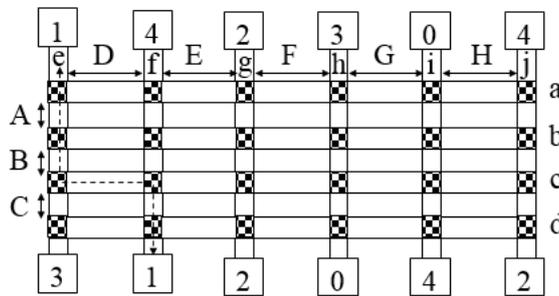
9
15

5. 鼠窩間的地道

時間限制：2 秒

問題敘述：

有 K 隻地鼠在地下挖出了一個棋盤狀地道系統，由 N 條垂直地道與 K 條水平地道構成。在每個垂直地道的上方和下方，地鼠們也各自挖了一個地洞，作為鼠窩之用。如圖一所示，有 4 隻地鼠挖出像棋盤狀的 4 條水平地道(由上到下編號為 a, b, c, d)與 6 條垂直地道(由左到右編號為 e, f, g, h, i, j)，並且在每個垂直地道的上方與下方都各挖了一個洞。挖完地洞與地道之後，每隻地鼠至少都會分到兩個地洞當作自己的鼠窩，但有些地洞可能不會被地鼠選為鼠窩。地鼠的編號是 1 到 K 的整數，地洞如果被地鼠 k 選為鼠窩，其編號就是 k ；如果地洞沒有被任何地鼠選為鼠窩，則其編號為 0。圖一中，上下方各有一個地洞沒被選為鼠窩，所以編號為 0。



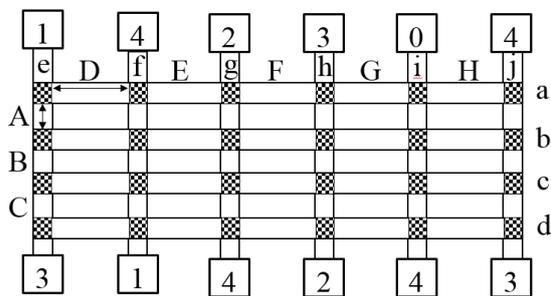
圖一

地鼠會沿著地道，在自己的鼠窩間移動。地鼠們在地道行走有下列規則：

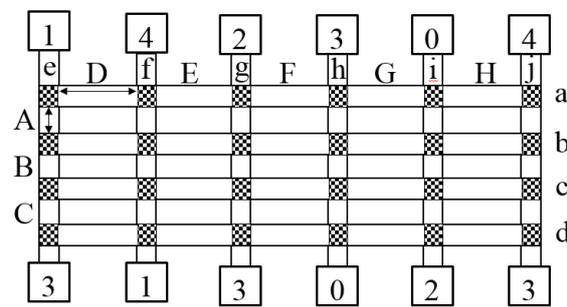
1. **地鼠不喜歡跟其他地鼠走在相同的地道上。**例如水平地道 d 的水平 E 段落，如果地鼠 1 走過，其他地鼠就不會再走這一段路，但是 d 的其他段落的路，如 D, F, G, H ，還是會有其他地鼠經過，垂直地道也有同樣的規則，例如垂直地道 g 的 C 段落如果被地鼠 2 走過，其他地鼠就不會再經過 g 的 C 段落。
2. **只有在兩隻地鼠都不改變行走方向的時候，才允許兩隻地鼠穿越相同的十字路口。**例如地鼠 2 垂直經過 g 與 c 的交叉路口，而地鼠 4 則可以水平經過同一個十字路口。如果一隻地鼠在十字路口改變方向，則其他地鼠就不能再經過此十字路口了。如地鼠 2 由下方地洞沿著 g 往上走並右轉走 c ，則其他地鼠就不能再經過 g 與 c 構成的十字路口。
3. **地鼠不喜歡轉彎。**在鼠窩間移動時，地鼠會選擇走**最少轉折的路徑**。地鼠在兩個鼠窩間移動時，不是沒有轉折(兩個鼠窩在同一條垂直地道上時)，就是只會有兩個轉折(兩個鼠窩不在同一條垂直地道上時)，如圖一中地鼠 1 走到下方的另一個窩的路徑就是只有兩個轉折(虛線雙箭頭所示)。

為了分配好每隻地鼠在地道行進的路徑，地鼠們開會後決定每隻地鼠都會有自己的一條專用水平地道。當每隻地鼠想走到自己的另一個鼠窩，需要水平移動的時候，必須要走自己專用的那一條水平地道。只要兩隻地鼠會走過的水平地道範圍(含端點)沒有重疊，他們是可以共用同一條水平地道的；例如，在圖一中，地鼠 1 與地鼠 2 的最左邊與最右邊鼠窩的水平地道範圍並沒有重疊，所以他們是可以共同使用相同的水平地道。首先需要解決的問題是，**是否存在一個合理的水平地道分配方式**，讓地鼠們不用違反規則，便能在自己的鼠窩之間移動。圖二即是一個沒有合理分配方式的例子。

接下來，我們要考慮的地鼠們分配到的水平地道間的上下關係。雖然圖一的例子有多種合理的水平地道分配方式，但只要合理，那麼地鼠 4 被分配到的水平地道一定是最高的。並不是每一種有合理地道分配答案的問題中，被分配到最高水平地道的地鼠是唯一解，如圖三的例子，在合理的水平地道分配方式中，地鼠 4 或者是地鼠 2 都可以被分配到最高的水平地道中。如果存在合理的水平地道分配，請找出所有可以被分配到最高水平地道的地鼠。



圖二



圖三

我們用 (k,a) 代表地鼠 k 被分配到的專屬水平地道是 a 。圖一的例子中， $(1,b), (2,b), (3,c), (4,a)$ 是一種合理的水平地道分配方式。雖然地鼠 1、2、4 所分配到的水平地道 $b、b、a$ 都高於地鼠 3 的 c ，但地鼠 2 分配到水平地道即使改成 d ，仍然是一個合理的分配。以圖一的例子，對於任一個合理的水平地道分配，地道位置一定在地鼠 3 水平地道上方的地鼠就只有地鼠 1 與 4 了。最後，在有合理水平地道分配的前提下，給定 k ，請算出有多少隻地鼠分配到的水平地道一定比地鼠 k 分配到的水平地道高？

輸入說明：

輸入檔案的第一行有一個數字 $T \leq 20$ ，代表有多少筆測試資料。每一個測試資料的第一行有兩個數字以空白字元隔開，先後分別代表地鼠的數目 $K \leq 2000$ 與垂直地道的數目 $N \leq 4000$ 。第二行有 N 個由空白隔開的數字，由左到右代表著上方地洞由左到右的編號。第三行也有 N 個以空白隔開的數字，由左到右代表著下方地洞由左到右的編號。第四行為一個數字 $Q \leq 2000$ ，第五行有 Q 個以空白隔開的數字 k_1, \dots, k_Q ，代表要分別計算出有多少隻地鼠分配到的水平地道一定比地鼠 k_1, \dots, k_Q 分配到的高。

部分分數：

第一筆測試資料 $K \leq 10、N \leq 15$ ，第二筆測試資料 $K \leq 60、N \leq 160$ ，之後 K 逐筆變大。

輸出說明：

對第 t 筆測試資料，第一行請印出「Case t:」，如第 1 筆請印「Case 1:」。如果該筆測試資料沒有合理的水平地道分配，第二行印出「no solution」即可。如果該筆測試資料有合理的水平地道分配，第二行印出所有可以分配到最高水平地道的地鼠編號，每個編號以一個空白字元隔開，編號請以由小到大的順序列印。第三行至第 $Q + 2$ 行則依序印出多少隻地鼠分配到的水平地道一定比地鼠 k_1, \dots, k_Q 分配到的水平地道高。

輸入範例：

```
3
4 6
1 4 2 3 0 4
3 1 2 0 4 2
2
2 3
4 6
1 4 2 3 0 4
3 1 4 2 4 3
2
2 3
4 6
1 4 2 3 0 4
3 1 3 0 2 3
2
2 3
```

輸出範例：

```
Case 1:
4
1
2
Case 2:
no solution
Case 3:
2 4
0
3
```

(本頁空白)

6. 圈地問題

限制時間：2 秒

問題敘述：

阿朗·伍德是一位農夫，他擁有一片他非常自豪的圓形牧場。他自己休息的農舍則設置在這圓形牧場的正中央。然而好景不常，最近在附近森林的狼群肆虐下，他豢養於牧圈中的牛群飽受騷擾。為此他決定派他的獵犬來抵禦狼群的騷擾。方便起見，讓我們假設阿朗的牧場是一個以原點 $(0,0)$ 為圓心， $R = 1$ 為半徑的正圓；而阿朗的農舍則座落在原點上。

阿朗在原先這個牧圈的圍牆（也就是圓周）上找了 N 個適合讓獵犬鎮守的點，我們姑且稱他們為「哨站」。然而他只養了 K 條獵犬，因此並不是所有哨站都能被獵犬鎮守——他只能選擇其中的 K 個哨站防守。更明確地說，阿朗會選擇其中的 K 個哨站讓獵犬防守，並按照他們在圓周上的順序將所有相鄰的兩個哨站用直線圍籬相互連接。對於這樣一個配置，我們稱其「有效防守面積」為被圍籬圍起的總面積。要注意的是，一個合法的配置必須確定阿朗休息的農舍（即原點）被包含在圍籬中或是恰好在圍籬上。

給定牧圈圓周上的 N 個點，幫阿朗決定在合法的最佳配置下，「有效防守面積」最大可以是多少？

輸入說明：

輸入中將至多有 20 筆測試資料，並以 EOF 結尾。

每組輸入的第一行是兩個整數 N 與 K ($3 \leq K \leq N \leq 1000$)，依序代表哨站的數量及獵犬的數量。接下來 N 行各有一個介在 $[0, 2\pi]$ 之間的浮點數，代表這 N 個哨站在圓周上的位置：該浮點數代表這個哨站與圓心連線和 X 軸的夾角。輸入保證任兩個哨站的位置（角度）都不相同。

輸出說明：

對於每筆測試資料，請輸出一個浮點數四捨五入至小數點下四位，代表最佳配置下可以達到的最大有效防守面積。若沒有任何合法的配置存在，則請輸出 0.0000。

輸入範例：

4 3
1.5708
0.0000
3.1416
4.7124
4 4
1.5708
0.0000
3.1416
4.7124

輸出範例：

1.0000
2.0000