

程式設計 (113-1)

作業四

作業設計：孔令傑
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS (<http://pdogs.ntu.im/>) 為第一、二、三、四題各上傳一份 C++ 原始碼 (以複製貼上原始碼的方式上傳)。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。這份作業的截止時間是 **10 月 01 日早上八點**。為這份作業設計測試資料並且提供解答的助教是葉又銘。

在你開始前，請閱讀課本的第 5.20–5.22 (關於遞迴) 和第 19 章 (關於搜尋和排序)¹。

本次作業滿分為 110 分，得幾分就算幾分。若整學期有 n 份作業，則學期的作業總成績即為 n 份作業的總分除以 n (不論超過 100 與否)。

第一題

(20 分) 奧運會上各國之間會有獎牌排名，雖然有各種排名方式，但最常見的是「先比金牌數，多者排前；如果金牌數一樣，則比銀牌數，多者排前；如果銀牌數一樣，則比銅牌數，多者排前」。已知在某一天賽事開始前，沒有任兩個國家的三種獎牌數都一樣，且照此規則排序後， n 個國家中排名第 i 的國家的金、銀、銅牌數各是 g_i 、 s_i 和 b_i 。若在該天賽事中排名第 k 的國家多獲得了 G 、 S 、 B 面金、銀、銅牌，且其它國家的獎牌數都不變，我們想知道這個有獲得獎牌的國家的排名會上升到第幾名。如果這個國家獲得獎牌後的各種獎牌數剛好跟另一個國家的一模一樣，規定上是剛獲得獎牌的這個國家要被排名在後。

舉例來說，如果原本有 $n = 5$ 個國家，每個國家的排名和獎牌數如表 1 所示。則若原本第 $k = 3$ 名的國家多得到 $G = 2$ 面金牌、 $S = 0$ 面銀牌、 $B = 0$ 面銅牌，它的排名將升到第二名；若原本第 $k = 3$ 名的國家多得到 $G = 2$ 面金牌、 $S = 4$ 面銀牌、 $B = 0$ 面銅牌，它的排名將升到第一名；若原本第 $k = 4$ 名的國家多得到 $G = 0$ 面金牌、 $S = 5$ 面銀牌、 $B = 3$ 面銅牌，它的排名將維持在第四名。

排名	金牌數	銀牌數	銅牌數
1	10	6	1
2	10	2	8
3	8	2	9
4	7	5	4
5	7	5	3

Table 1: 獎牌排名

給定以上資訊，請求出有獲得獎牌之國家的新排名。

¹課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $n + 1$ 列，第一列裝著五個正整數，依序是 n 、 k 、 G 、 S 、 B ；在從第二列到第 $n + 1$ 列中，第 $i + 1$ 列存了三個非負整數，依序是 g_i 、 s_i 和 b_i 。已知 $1 \leq n \leq 100$ ，其它所有整數都介於 0 和 50（包含 0 和 50），且不存在不同的 i 和 j 同時滿足 $g_i = g_j$ 、 $s_i = s_j$ 和 $b_i = b_j$ 。每一列的任兩個相鄰的整數間以一個空白字元隔開。

讀入這些資訊後，請輸出原本排名第 k 的國家獲得更多獎牌後的新排名，若新排名為第 r 名則輸出 r 。舉例來說，如果輸入是

```
5 3 2 0 0
10 6 1
10 2 8
8 2 9
7 5 4
7 5 3
```

則輸出應該是

```
2
```

如果輸入是

```
5 4 0 5 3
10 6 1
10 2 8
8 2 9
7 5 4
7 5 3
```

則輸出應該是

```
4
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。

第二題

(20 分) 有一個物流公司每週會配送食材到各個市場，我們令 $I = \{1, \dots, m\}$ 為市場的集合。每個市場都有三個參數，而市場 i 的參數為 X_i 、 Y_i 和 q_i ，其中 X_i 和 Y_i 是市場 i 在笛卡兒座標系的位置座標， q_i 是市場 i 每週的需求量。

食材配送是從物流中心出發的。我們令 $J = \{1, \dots, n\}$ 為候選地點的集合，每個物流中心候選地點也有三個參數，而候選地點 j 的參數為 X_j 、 Y_j 、 f_j ，其中 X_j 和 Y_j 是候選地點 j 的位置座標， f_j 是在該地點設置物流中心後每週的營運成本。從市場 i 到物流中心候選地點 j 的距離用曼哈頓距離計算，亦即兩地點間的距離為 $d_{ij} = |X_i - X_j| + |Y_i - Y_j|$ 。已知從物流中心運送一單位食材到市場時，一單位距離的成本是 C 。從總部到物流中心的補貨成本因為是有效率的大宗運補，所以可以忽略不計。

該公司原本就已經有若干個早已建設好、已經營運中的物流中心了。我們用 A_1 、 A_2 到 A_n 表示各候選地點上是否已經有物流中心了，如果 $A_j = 1$ 表示候選地點 j 上已經有物流中心， $A_j = 0$ 則表示沒有。公司現在希望能在還沒有物流中心的候選地點中，選擇一個地點去新建一個物流中心，以最小化每週的總成本，如果有複數個地點都可以最小化總成本，就選編號最小的。請注意因為新增物流中心會帶來額外的營運成本，不無可能在任意一個可行的候選地點上新建物流中心都讓總成本上升。若是如此，公司就應該選擇不新建物流中心。

舉例來說，假設有 $m = 3$ 個市場，市場 1 位在 (2, 3) 且單週需求為 20，市場 2 位在 (5, 6) 且單週需求為 30，市場 3 位在 (8, 9) 且單週需求為 25；有 $n = 3$ 個物流中心候選地點，地點 1 位在 (1, 2) 且單週營運成本為 100，地點 2 位在 (4, 5) 且單週營運成本為 120，地點 3 位在 (7, 8) 且單週營運成本為 150。如果目前沒有已經建設好的物流中心（亦即 $A_1 = A_2 = A_3 = 0$ ），且 $C = 5$ ，則公司可以依序考慮三個地點。如果在地點 1 建立物流中心，單週補貨總成本是 $5 \times (20 \times 2 + 30 \times 8 + 25 \times 14) = 3150$ ，單週營運成本為 100，因此單週總成本為 3250；如果在地點 2 建立物流中心，單週總成本為 1820；如果在地點 3 建立物流中心，單週總成本為 2000。結論是在地點 2 新建物流中心可以讓總成本最小，單週總成本為 1820。

前一個例子中原本是有沒有已經建好的物流中心的，所以讓我們再看一個例子。假設還是那三個市場和那三個候選地點，但是在候選地點 2 已經有物流中心了。那麼公司的考量將會是：

- 如果在地點 1 建立物流中心，市場 1、2、3 應該分別由在地點 1、地點 2、地點 2 的物流中心補貨，三個市場的單週補貨成本將依序是 $5 \times 20 \times 2 = 200$ 、 $5 \times 30 \times 2 = 300$ 、 $5 \times 25 \times 8 = 1000$ ，合計 1500，而單週營運成本為 $100 + 120 = 220$ ，因此單週總成本為 1720，比原本的 1820 小，是一個可以考慮的選擇。
- 如果在地點 3 建立物流中心，市場 1、2、3 應該分別由在地點 2、地點 2、地點 3 的物流中心補貨，三個市場的單週補貨成本將依序是 $5 \times 20 \times 4 = 400$ 、 $5 \times 30 \times 2 = 300$ 、 $5 \times 25 \times 2 = 250$ ，合計 950，而單週營運成本為 $120 + 150 = 270$ ，因此單週總成本為 1220，也比原本的 1820 小，是一個可以考慮的選擇。

結論是在地點 2 已經有一個物流中心的情況下，在地點 3 再新建一個物流中心可以讓總成本最小，新的總成本是 1220。

在本題中，助教們已經把 main function 寫好了，你只要寫一個「函數」，在函數內做計算並且回傳最後的答案。具體來說，首先我們定義全域常數

```
const int MAX_MKT_CNT = 20;
```

```
const int MAX_LOC_CNT = 10;
```

其中 `MAX_MKT_CNT` 代表市場個數上限，`MAX_LOC_CNT` 代表物流中心候選地點個數上限。你要實做的函數的 prototype 為

```
int getBestFacility(int mktCnt, int locCnt, int unitShippingCost, const int
    distances[][MAX_LOC_CNT], const int oprtCosts[], const int demands[],
    const bool curDCs[]);
```

其中 `mktCnt` 代表市場總數；`locCnt` 代表物流中心候選地點總數；`unitShippingCost` 代表從物流中心運送食材到市場食，運送一單位食材移動一單位距離的成本；`distances` 是長度為 `MAX_MKT_CNT` 乘以 `MAX_LOC_CNT` 的二維陣列，其中的第 $i-1$ 列的第 $j-1$ 個元素是市場 i 和物流中心 j 的距離；`oprtCosts` 是長度為 `MAX_LOC_CNT` 的一維陣列，其中的第 $j-1$ 個元素是在地點 j 營運一個物流中心一週的成本；`demands` 是長度為 `MAX_MKT_CNT` 的一維陣列，其中的第 $i-1$ 個元素是市場 i 一週的需求量；`curDCs` 是長度為 `MAX_LOC_CNT` 的一維陣列，其中的第 $j-1$ 個元素的值表示地點 j 是否已經有物流中心了，`true` 表示有，`false` 表示沒有。函數應該回傳能最小化總成本的新建一個物流中心的地點編號，如果新建任何一個都比不新建帶來更高的成本，則回傳 0。

特別注意：在這題之中，助教們已經在 PDOGS 上設定好 main function 了（雖然沒給你看，但他們會寫個正確的）和那兩個全域常數了。你需要完成一個完整的 `getBestFacility` 函數，繳交時也只需要到 PDOGS 上傳這個 `getBestFacility` 函數，PDOGS 會自動把你上傳的函數跟已經在 PDOGS 上的程式拼起來去編譯。換言之，在本題你被迫必須要實作本題指定的函數；如果你上傳了任何帶有你寫的 main function 的程式，你會無法得到分數的！

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有八列，第一列裝著三個正整數，依序是 n 、 m 和 C ；第二列存了 n 個整數，依序是第 1 個、第 2 個直到第 n 個候選地點的 x 座標；第三列存了 n 個整數，依序是第 1 個、第 2 個直到第 n 個候選地點的 y 座標；第四列存了 m 個整數，依序是第 1 個、第 2 個直到第 m 個市場的 x 座標；第五列存了 m 個整數，依序是第 1 個、第 2 個直到第 m 個市場的 y 座標；第六列存了 n 個整數，依序是 f_1 、 f_2 直到 f_n ，分別是各候選地點上營運物流中心的單週營運成本；第七列存了 m 個整數，依序是 q_1 、 q_2 直到 q_m ，代表每個市場的單週需求量；第八列存了 n 個非零則一的整數，依序是 A_1 、 A_2 直到 A_n ，代表候選地點是否已經有物流中心了，1 表示有，0 表示沒有。已知 $1 \leq n \leq 10$ 、 $1 \leq m \leq 20$ 、 $1 \leq C \leq 10$ 、所有的座標值都落在 0 和 100 之間（包含 0 和 100）、 $1 \leq f_j \leq 10000$ 、 $1 \leq q_i \leq 100$ 、 $A_j \in \{0, 1\}$ 。

讀入這些資訊後，你的程式（PDOGS 上的 main function 加上你實作的程式）應該輸出一個整數，代表應該被新建物流中心的候選地點編號。舉例來說，如果輸入是

```
2 5 5
35 26
25 17
25 53 25 27 47
67 27 55 61 67
16 45
19 18 69 70 75
```

0 0

則輸出應該是

1

如果輸入是

```
5 11 5
35 36 25 25 46
32 62 46 24 36
24 26 36 36 36 46 36 36 25 64 25
37 84 25 74 25 84 57 47 46 38 48
28 26 47 47 78
72 83 16 48 28 25 37 94 37 27 36
1 0 1 0 1
```

則輸出應該是

2

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。

第三題

(50 分) 承上題，現在我們要幫公司解決他們真正的問題：他們目前只有在地點 1 有一個物流中心，剩下的 $n - 1$ 個地點都還沒有，他們要在那 $n - 1$ 個候選地點中挑若干個（可能超過一個、一個或零個）蓋物流中心，以最小化總成本。你要使用的演算法很簡單：執行若干輪挑選，每一輪都用第二題的規則挑一個可以讓總成本最小的地點去新建一個物流中心，然後就視同剛剛挑的地點上已經有物流中心了，再執行下一輪的挑選，直到所有候選地點都被選過了，或者剩餘的候選地點中已經沒有可以讓總成本降低的了。當演算法停止時，請輸出你找到的建設方案（可能不是最佳，但通常也很不錯了）。

假設有 $m = 3$ 個市場，市場 1 位在 (2,3) 且單週需求量为 20，市場 2 位在 (5,6) 且單週需求量为 30，市場 3 位在 (8,9) 且單週需求量为 25；有 $n = 3$ 個候選地點，地點 1 位在 (1,2) 且單週營運成本為 100，地點 2 位在 (4,5) 且單週營運成本為 120，地點 3 位在 (7,8) 且單週營運成本為

150。已知 $C = 2$ ，且地點 1 已經有一個物流中心了，因此在我們考慮新建任何物流中心前，總成本是 $2 \times (20 \times 2 + 30 \times 8 + 25 \times 14) + 100 = 1360$ 。

公司在演算法的第一輪會依序考慮地點 2 和地點 3：

- 如果在地點 2 建立物流中心，市場 1、2、3 應該分別由在地點 1、地點 2、地點 2 的物流中心補貨，三個市場的單週補貨成本將依序是 $2 \times 20 \times 2 = 80$ 、 $2 \times 30 \times 2 = 120$ 、 $2 \times 25 \times 8 = 400$ ，合計 600，而單週營運成本為 $100 + 120 = 220$ ，因此單週總成本為 820，比原本的 1360 小，是一個可以考慮的選擇。
- 如果在地點 3 建立物流中心，市場 1、2、3 應該分別由在地點 1、地點 3、地點 3 的物流中心補貨，三個市場的單週補貨成本將依序是 $2 \times 20 \times 2 = 80$ 、 $2 \times 30 \times 4 = 240$ 、 $2 \times 25 \times 2 = 100$ ，合計 420，而單週營運成本為 $100 + 150 = 250$ ，因此單週總成本為 670，也比原本的 1360 小，是一個可以考慮的選擇。

結論是在地點 1 已經有一個物流中心的情況下，在地點 3 再新建一個物流中心可以讓總成本最小，新的總成本是 670。

接著進入演算法第二輪，公司在已經選了地點 1 和 3 的前題下，將再接著考慮最後剩餘的地點 2。如果在地點 2 再建一個物流中心，市場 1、2、3 應該分別由在地點 1、地點 2、地點 3 的物流中心補貨，三個市場的單週補貨成本將依序是 $2 \times 20 \times 2 = 80$ 、 $2 \times 30 \times 2 = 120$ 、 $2 \times 25 \times 2 = 100$ ，合計 300，而單週營運成本為 $100 + 120 + 150 = 370$ ，因此單週總成本為 670。由於找不到能讓總成本更小的候選地點，演算法將停止，最終找到的建設計畫就是在地點 3 新建一個物流中心。

在這一題，我們順便介紹一下如何用數學規劃（mathematical programming）去描述這個問題，大家也可以試著透過數學式子的描述對問題有更精準的理解²。我們可以定義一個決策變數

$$y_j = \begin{cases} 1 & \text{若我們有在候選地點 } j \text{ 建立物流中心} \\ 0 & \text{若沒有} \end{cases}。$$

給定建設的決策後，我們理所當然地也會知道應該由哪個物流中心去幫哪個市場補貨（就是最近的那個），亦即我們可以決定

$$x_{ij} = \begin{cases} 1 & \text{若市場 } i \text{ 是由物流中心 } j \text{ 補貨} \\ 0 & \text{若不是} \end{cases}。$$

有了以上這些資訊，我們就可以建構這家物流公司的最佳化問題為

$$\begin{aligned} \min \quad & C \sum_{i \in I} q_i \sum_{j \in J} d_{ij} x_{ij} + \sum_{j \in J} f_j y_j \\ \text{s.t.} \quad & \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \\ & x_{ij} \leq y_j \quad \forall i \in I, j \in J \\ & y_1 = 1 \\ & x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \\ & y_j \in \{0, 1\} \quad \forall j \in J, \end{aligned}$$

²如果真的看不懂這個最佳化問題的數學表達式，可以跳過，不影響大家做這一題，但我們還是建議大家花時間看看，因為將來有許多課程在許多地方都會用數學符號描述一個要被求解的問題，畢竟數學某方面來說是人類社會最精確的語言。

其中第一條限制式要求每個市場都由恰好一個物流中心補貨，第二條限制式要求如果沒有在地點 j 蓋物流中心，就不能由該地點對市場 i 補貨，第三條限制式表示地點 1 上的物流中心必須持續存在，第四和第五條限制式要求建設決策只有蓋或不蓋，不能蓋一半，而補貨選擇也只能選或不選，不能一半一半。最後，目標式中的第一項為單週總補貨成本（正比於運算數量和運送距離），第二項為單週總營運成本，目標是要最小化單週總成本。

輸入輸出格式

系統會提供一共 15 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有七列，第一列裝著三個正整數，依序是 n 、 m 和 C ；第二列存了 n 個整數，依序是第 1 個、第 2 個直到第 n 個候選地點的 x 座標；第三列存了 n 個整數，依序是第 1 個、第 2 個直到第 n 個候選地點的 y 座標；第四列存了 m 個整數，依序是第 1 個、第 2 個直到第 m 個市場的 x 座標；第五列存了 m 個整數，依序是第 1 個、第 2 個直到第 m 個市場的 y 座標；第六列存了 n 個整數，依序是 f_1 、 f_2 直到 f_n ，分別是各候選地點上營運物流中心的單週營運成本；第七列存了 m 個整數，依序是 q_1 、 q_2 直到 q_m ，代表每個市場的單週需求量。已知 $1 \leq n \leq 10$ 、 $1 \leq m \leq 20$ 、 $1 \leq C \leq 10$ 、所有的座標值都落在 0 和 100 之間（包含 0 和 100）、 $1 \leq f_j \leq 10000$ 、 $1 \leq q_i \leq 100$ 。

讀入這些資訊後，你的程式應該輸出 n 個非零即一的整數，兩個整數之間用逗號隔開，代表每個地點是否建立物流中心，1 表示要建立，0 表示不要。舉例來說，如果輸入是

```
4 3 5
46 26 12 32
27 27 36 33
17 37 37
47 59 57
47 49 48 38
38 96 37
```

則輸出應該是

```
1,0,1,1
```

如果輸入是

```
6 11 5
32 26 25 36 36 26
36 26 46 18 28 58
14 47 27 26 27 27 26 37 36 31 27
47 59 48 48 58 48 58 58 58 48 48
46 37 75 26 27 46
48 37 38 38 48 27 26 37 25 19 49
```

則輸出應該是

```
1,0,1,0,0,1
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**不可以**使用上課沒有教過的方法：

- 確定可以使用的語法包含 `if-else`、`for`、`while`、陣列、函數、`<climits>` 裡面所有的東西、`<iomanip>` 裡面所有的東西、`<cmath>` 裡面的 `abs()` 和 `sqrt()`、`sizeof()`、`static_cast()`、`constants` 等。
- 確定不可以使用的語法包含 `printf`、`scanf`、`max`、`min`、`<cmath>` 裡面除了 `abs()` 和 `sqrt()` 以外的函數、動態配置記憶體等等。

請注意正面表列的固然是都確定可以用，但沒有被負面表列的不表示可以用喔！

此外，請幫你的程式寫適當的函數，例如你第二題已經寫好的函數，或者你自己設計的函數，但總之你的程式應該要藉由使用函數來實現理想的程式結構。

評分原則

- 這一題的其中 30 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。
- 這一題的其中 20 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的結構、運算邏輯、可讀性（包含排版、變數命名、註解等等）、可擴充性、模組化程度，以及是否使用了還沒教過的語法。請寫一個「好」的程式吧！

第四題

(20 分) 在一個國家裡有 n 個小鎮，某些小鎮間有路相連，總共有 m 條路，每條路都是起自一個小鎮也結束於一個小鎮，路上沒有別的小鎮。小鎮間的道路關係可以用一個 $n \times n$ 的對稱矩陣 R 表示。以

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

為例，表示 $n = 5$ 、 $m = 6$ ，小鎮 1 跟 2 之間有一條路、小鎮 1 跟 5 之間有一條路，依此類推。在每個小鎮上住有居民 h_i 人。國王想要看看各小鎮發展的情況，也拜訪民眾，因此想要從小鎮 1（其實是首都）出發，經過 q 個不同的小鎮後回到小鎮 1。他擬定了路線 $(1, p_1, p_2, \dots, p_q, 1)$ ，表示先從小鎮 1 走到小鎮 p_1 ，再從小鎮 p_1 走到小鎮 p_2 ，依此類推，最後從小鎮 p_q 回到小鎮 1。舉例來說， $(1, 5, 2, 1)$ 就是一個合情合理的路線，沿路可以拜訪 $h_1 + h_5 + h_2$ 這麼多個民眾（小鎮 1 雖然被經過兩次，但被拜訪的民眾數只被計算一次）。

以上一切都很完美，唯一就是國王不太擅長擬定路線，因此國王把路線交給你，請你檢查他擬的路線上是否每條路都確實存在。舉例來說，如果國王擬的路線是 $(1, 5, 4, 2, 1)$ ，這個路線就行不通，因為

從小鎮 5 沒有路通往小鎮 4。國王給你的任務是，如果給定的路線是可行的，就計算可以拜訪的民眾數；如果給定的路線不可行，就依照路線上的順序依序列舉不存在的路段。舉例來說，如果國王給的路線是 $(1, 5, 2, 1)$ ，就輸出 $h_1 + h_5 + h_2$ ；如果是 $(1, 5, 4, 2, 1)$ ，就輸出 $(5, 4)$ 和 $(4, 2)$ 。

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $n + 3$ 列，第一列裝著三個正整數，依序是 n 、 m 、 q ；在從第二列到第 $n + 1$ 列中，第 $i + 1$ 列存了 n 個非零則一的整數，依序是 $R_{i,1}$ 、 $R_{i,2}$ 直到 $R_{i,n}$ ；第 $n + 2$ 列存了 n 個正整數，依序是 h_1 、 h_2 直到 h_n ；第 $n + 3$ 列存了 q 個正整數，依序是 p_1 、 p_2 直到 p_q 。已知 $1 \leq n \leq 100$ 、 $1 \leq m \leq 4950$ 、 $1 \leq q \leq n - 1$ 、 $R_{ij} \in \{0, 1\}$ 、 $1 \leq h_i \leq 1000$ 、 $p_i \in \{2, \dots, n\}$ 且 $p_1 \neq p_2 \neq \dots \neq p_q$ 。每一列的任兩個相鄰的整數間以一個空白字元隔開。

讀入這些資訊後，如果指定路線上的每個路段都存在，就輸出一個整數代表路線上會拜訪的總民眾數；反之則按照路線順序輸出所有不存在的路段，每當要輸出一個路段時，先輸出該路段的起點小鎮編號，接著輸出一個逗點，再輸出該路段的終點小鎮編號，如果後面還有下一個路段，兩個路段間用一個分號隔開。舉例來說，如果輸入是

```
5 6 2
0 1 0 0 1
1 0 1 0 1
0 1 0 1 1
0 0 1 0 0
1 1 1 0 0
300 500 200 100 300
5 2
```

則因為 $(1, 5, 2, 1)$ 路線上每個路段都存在，輸出應該是

```
1100
```

如果輸入是

```
5 6 3
0 1 0 0 1
1 0 1 0 1
0 1 0 1 1
0 0 1 0 0
1 1 1 0 0
300 500 200 100 300
5 4 2
```

則因為 $(1, 5, 2, 1)$ 路線上 $(5, 4)$ 、 $(4, 2)$ 路段不存在，輸出應該是

```
5,4;4,2
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。