

程式設計 (113-1)

作業二

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 **9 月 17 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黃芷榆。

在你開始前，請閱讀課本的第 6.1–6.4 節 (關於陣列)¹。第 6.7 和 6.8 節也有幫助。

本次作業滿分為 110 分，得幾分就算幾分，若行有餘力就請多拿分吧！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) 給定一群 n 個人，若我們挑出其中兩人 i 跟 j ，就可以問他們並得知 i 是否認識 j 以及 j 是否認識 i 。 n 個人之間一共有最多 $n(n-1)$ 個「認識」。給定 m 個「認識」資訊，我們想找出被最多人認識的人是被多少人認識，以及認識最多的人認識多少人。顯然這兩個數字最大都只能是 $n-1$ ，最小則可能是 0 (如果全部的人都互不相認識)。

舉例來說，若現在有四個人 1、2、3、4，並且我們知道 1 認識 2、1 認識 3、1 認識 4、4 認識 3，則被最多人認識的人是 3，他被兩個人認識，而認識最多的人是 1，他認識三個人。

輸入輸出格式

系統會提供一共 10 組測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中會有 $m+1$ 列，第一行包含兩個整數 n 和 m ，之後 m 列每一列有兩個不相同的整數 i 和 j ，表示 i 認識 j 。每一列的兩個整數之間被一個空白隔開。已知 $2 \leq n \leq 50$ 、 $0 \leq m \leq n(n-1)$ 、 $i \in \{1, 2, \dots, n\}$ 、 $j \in \{1, 2, \dots, n\}$ ，且最後 m 列中沒有任兩列是一樣的。

讀入資料後，請按照題目指定的規則，先印出被最多人認識的人是被多少人認識，接著印出一個逗點，再印出認識最多的人認識多少人。舉例來說，如果輸入是

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

則輸出應該是

```
2,3
```

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

如果輸入是

```
50 0
```

則輸出應該是

```
0,0
```

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) 本題將給定一個不大於 9999 的正整數，請將其看作四位數字（亦即其若不大於 999 則前面補 0），並找出將數字重新排序後，能組成的最大整數。舉例來說，若給定數字為 1323，經過重新排列，能組成的最大數字為 3321；若原先給定的數字非四位數，如：25，則需看成 0025 這四位數字去做排列，重組後最大數字為 5200。本題給定的數字不會四位皆相同，例如不會是 9999。

在本題中，除了寫出對的程式之外，我們也要求你用「函數」來完成這一題。請想像你要跟你的一位夥伴一起寫這一題，而你的夥伴已經把 main function 寫好了，要求你按照他的設計寫一個函數；他會把讀入的資料存入對應的變數或陣列，接著把這些變數和陣列傳進你寫的函數，讓你做計算並且回傳指定的結果。具體來說，他定義了這個函數的 prototype 為

```
int getBiggestByReordering(int theValue);
```

其中 theValue 是一個不大於 9999 且四個數字不完全一樣的正整數，回傳值則應該是照前述規則產生的重新排列後的最大的數，例如若 theValue 為 1323，則回傳值應該是 3321。

他寫的 main function 如下（如附件 PD113-1_hw02_main02.cpp）：

```
#include <iostream>
using namespace std;

// This is the prototype of the function that you need to implement
int getBiggestByReordering(int theValue);

int main(){
    int aValue = 0;
```

```

    cin >> aValue;

    int biggest = getBiggestByReordering(aValue);

    cout << biggest;

    return 0;
}

// PDOGS will copy and paste the code you upload to this place
// and compile the resulting program

```

在這題之中，助教已經在 PDOGS 上設定好上面的「你的夥伴」寫的程式了。你需要完成一個完整的 `getBiggestByReordering` 函數，自己測試的時候當然需要結合上面的 `main function`，但在繳交到 PDOGS 時請只上傳這個 `getBiggestByReordering` 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！

特別提醒：本題開發的函數有很高的機率會在下一題被用到，而下一題是不准使用沒教過的語法的，所以雖然這一題我們不管，但建議大家在寫這一題時還是不要使用沒教過的語法，不然到下一題你就得重寫一次這個函數了！

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。在每個檔案中會有一行，包含一個正整數 n ，其中 $1 \leq n \leq 9999$ ，且四位數字不會皆為同樣的數字。請依題目規則重新排列數字，並印出最大之數字組合。

舉例來說，如果輸入是

1323

則輸出應該是

3321

如果輸入是

25

則輸出應該是

5200

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) 仔細看看 6174 這個數字。乍看之下沒什麼特別之處，但是，自從 1949 年以來，它一直令數學家 and 數字控抓狂與癡迷。為什麼呢？因為人們發現，當我們任意選擇一個四位數字後，只要這個四位數字沒有四個數字都一樣，那麼如果先將數字由大到小排序，得出大數，再由小到大排序，得出小數，並把所得出的大數減小數，並且不斷重複這幾個步驟，最終都會得到 6174 這個神祕的數字。舉例來說，如果輸入是 3087，我們在第一輪將 3087 重新排列後的大數是 8730，小數是 0378 亦即 378，大數減小數是 $8730 - 378 = 8352$ ；在第二輪將 8352 重新排列後的大數是 8532，小數是 2358，大數減小數是 $8532 - 2358 = 6174$ 。果然是 6174！

我們已經確定任何四個數字不完全相同的介於 1 到 9999 的整數經過上面的流程後，都會得到 6174 了。姑且不論為什麼，本題我們將要請你實作這個過程。

若要完成這一題，顯然使用第二題已經寫好的 `getBiggestByReordering` 函數應該很有幫助。你應該也會想要另外寫一個 `getSmallestByReordering` 函數，長得跟前者有八成七像。如果兩個函數都有了，那以 3087 為例，我們先呼叫第一個函數得到重新排列後能得到的最大的數 8730，再呼叫第二個函數得到 378，然後相減，就可以進入下一輪了。

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。在每個檔案中會有一行，包含一個正整數 n ，其中 $1 \leq n \leq 9999$ ，且四個數字不會皆為同樣的數字。請依題目規則重複相同步驟，並印出每輪計算後的結果，印出的數字兩兩之間以一個逗號隔開，直到印出 6174 後結束。

舉例來說，如果輸入是

1324

則輸出應該是

3087,8352,6174

如果輸入是

2005

則輸出應該是

5175,5994,5355,1998,8082,8532,6174

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！

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運算邏輯、可讀性（包含排版、變數命名、註解等等）、可擴充性，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）此題我們將實作一個簡單版的 Maximum Bipartite Matching 演算法。在這個題目中，我們將尋找一組一對一的配對方式來最大化總配對權重（weight）。這個演算法在實務上的應用有學校和學生的配對、公司和求職者的配對、捐腎者與受贈者的配對等等，以下我們將簡述此演算法。為了講解方便，我們假設有 m 個群組一的人與 n 個群組二的人要配對，任意兩個來自不同群組的人配對在一起後都會產生一個配對權重（效益），而此權重已經被量化了。讓我們用以下表格來舉例：

4	2	0	3
5	1	5	2
3	2	0	3

在這個例子中，有 3 個人在群組一、4 個人在群組二，任兩人配對的權重記錄在表格中，例如群組一的第 2 人和群組二的第 4 人配對的權重為 2。

由於要尋找一對一的配對，因此在一個合法的配對方案中，每一列最多只能對應到一欄，反之亦然。因此，以下的配對是不行的：

4			
5		5	

下方表格的配對情境皆為一對一的配對，因此是可行的，但在此題中並非最佳解，因為其權重總和並非所有解法中最大的：

4			
			2
		2	

以下配對為此題的最佳解：

4			
		5	
			3

此題中我們將不強求找出最佳解，只請你使用以下的方法求出還不錯的解。假設表格共有 m 列及 n 欄，且以 i 作為列的索引， j 作為欄的索引。若 m 小於等於 n ，則從第一列開始往第 m 列做處理，針對第 i 列，去該列中找尋尚未被配對給任一列且能使 w_{ij} 最大的第 j 欄。如果有數欄都使 w_{ij} 最大，我們選編號 j 最小的將之稱為第 j_i^* 欄，並將之配對給第 i 列。循此流程，我們將求得 j_1^* 、 j_2^* 直到 j_m^* ，即為一組一對一的配對。這組配對的總配對權重即為

$$\sum_{i=1}^m w_{i,j_i^*}。$$

若 n 小於 m ，則改由第一欄開始往第 n 欄做處理，針對第 j 欄，去該欄中找尋尚未被配對給任一欄且能使 w_{ij} 最大的第 i 列。如果有數列都使 w_{ij} 最大，我們選編號 i 最小的將之稱為第 i_j^* 列，並將之配對給第 j 欄。循此流程，我們將求得 i_1^* 、 i_2^* 直到 i_n^* ，即為一組一對一的配對。這組配對的總配對權重即為

$$\sum_{j=1}^n w_{i_j^*,j}。$$

以前述 3×4 的表格為例，因為 $m = 3 < n = 4$ ，因此我們以列為單位看。第一列中，由於沒有其他欄的值大於 4（在第一欄），因此 $j_1^* = 1$ 。接下來看第二列，由於第一欄已經被使用過了，因此我們選擇這一行中除了第一欄外最大的值（在第三欄）， $j_2^* = 3$ 。最後看第三列，撇除第一欄及第三欄外最大的值為第四欄，因此 $j_3^* = 4$ 。將總權重加起來，總共為 $4 + 5 + 3 = 12$ ，即為這組配對的總配對權重。

在本題中，請使用本題指定的演算法，針對給定的輸入求得一組配對，並計算其總配對權重。

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，總共有 $m + 1$ 行。第一行有兩個正整數，分別為 m 、 n ，分別代表列數及欄數。第二行至第 $m + 1$ 行之中，第 $i + 1$ 行有 n 個整數 $w_{i,1}$ 到 $w_{i,n}$ ，任意兩個整數之間以空白隔開。已知 $1 \leq m \leq 100$ 、 $1 \leq n \leq 100$ 、 $0 \leq w_{mn} \leq 100$ 。讀入這些數字之後，請依上述規則，當 $m \leq n$ 時，輸出 j_1^* 、 j_2^* 到 j_m^* ，任兩個整數之間以逗號隔開，

最後再輸出一個分號後輸出總配對權重。當 $n < m$ 時，輸出 i_1^* 、 i_2^* 到 i_n^* ，任兩個整數之間以逗號隔開，最後再輸出一個分號後輸出總配對權重。

舉例來說，如果輸入是

```
3 4
4 2 0 3
5 1 5 2
3 0 2 3
```

則輸出應該是

```
1,3,4;12
```

如果輸入是

```
3 4
5 1 5 2
4 2 0 3
3 0 2 3
```

則輸出應該是

```
1,4,3;10
```

如果輸入是

```
4 3
5 1 5
4 2 0
3 0 2
6 1 3
```

則輸出應該是

```
4,2,1;13
```

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 20 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。