

臺北市九十八學年度高級中等學校電腦程式設計競賽決賽試題

(高中組)

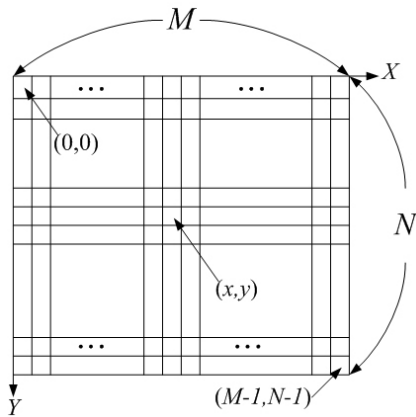
說明：

1. 本試卷共有四題，每題 25 分。
2. 請記得隨時備份自己的程式。

試題：

1. 數位影像處理

一張數位影像一般可視為是由寬為 M 個像素(Pixel)，長為 N 個像素的二維矩陣所構成，座標原點 $(0,0)$ 在左上角(如圖一所示)。此解析度(Resolution) $M \times N$ 的數位影像，任一個像素 (x,y) 是由一個位元組(Byte)，數值只可由 0 到 255，代表此像素的灰階亮度(Gray Intensity)，圖二為一個 5×5 範例影像。



圖一、數位影像的座標關係

67	100	109	55	109
137	185	92	56	132
158	174	62	89	154
174	182	99	107	133
189	181	119	125	184

圖二、一個 5×5 範例影像

在數位影像處理的過程中，水平及垂直鏡像(Mirror)、畫面平移(Translation)等影像座標轉換是常用的功能。原有的數位影像灰階內容，在經過特定座標轉換計算後，將影像灰階內容置於新的座標位置，而可以產生一張新的影像。轉換方式是：在原始影像中任一個座標在 (x,y) 位置的像素 ($0 \leq x \leq M-1$, $0 \leq y \leq N-1$)，其水平鏡像後的新座標為 (x',y') ，可由公式(1)中的轉換矩陣運算獲得。

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & M-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

同樣的，畫面平移 (T_x, T_y) 後的新座標 (x',y') ，也就是 X 方向平移 T_x 位置及 Y 方

向平移 T_y 位置，可由公式(2)中的轉換矩陣運算獲得。轉換後未被設定內容的未定義像素，內容以 0 表示，若新座標超出影像範圍的像素，則予以忽略。

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

以下圖為例，圖三為圖二中的範例影像，經過水平鏡像矩陣轉換後的 5×5 結果影像，圖四為經過平移矩陣轉換 (1, 2) 位置後的 5×5 結果影像。此兩個範例中，

水平鏡像及平移的轉換矩陣分別為 $\begin{bmatrix} -1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 及 $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$ ， $M=N=5$ 。

189	181	119	125	184
174	182	99	107	133
158	174	62	89	154
137	185	92	56	132
67	100	109	55	109

圖三、水平鏡像結果影像

0	0	0	0	0
0	0	0	0	0
0	67	100	109	55
0	137	185	92	56
0	158	174	62	89

圖四、畫面平移結果影像

試寫出能解決此兩類影像轉換之一組共同 **通用程式**，符合上述矩陣運算，以及未定義像素、邊界像素檢查之功能，當 3×3 轉換矩陣內容不同時，能輸出不同的轉換結果。資料輸入方式，是先輸入固定為 3×3 的轉換矩陣各個參數值，再輸入影像的動態解析度 M 和 N ，接著是影像的像素灰值，每一個數字間以空白間隔。因此，前述水平鏡像例子的資料輸入方式應該是：(由 sample1.txt 檔中讀入)

```
-1  0  4
0   1  0
0   0  1
5   5
67  100 109 55  109
137 185 92  56  132
158 174 62  89  154
174 182 99  107 133
```

189 181 119 125 184

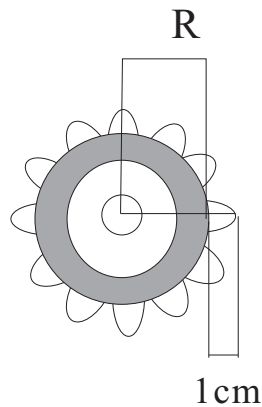
螢幕上依序輸出結果影像內容(每一個數字間以逗號及空白做為間隔)：

189, 181, 119, 125, 184,
174, 182, 99, 107, 133,
158, 174, 62, 89, 154,
137, 185, 92, 56, 132,
67, 100, 109, 55, 109,

☆測試結果時，除現有兩組範例外，會由其他的數個 sample 檔案中讀入影像內容檢測，測試用的 M 、 N 值、轉換矩陣內容及檢測影像像素灰值，可能是任何合理的數字組合。

2. 齒輪運轉

一個齒輪的規格如下圖，首先他有個半徑 R ，然後每個牙齒是一公分長。下圖的齒輪總共有 12 個牙齒。齒輪的中心點為 (x, y)



齒輪一旦彼此相連會彼此帶動。如下圖



假設有兩個齒輪 A，B，兩個齒輪的速度關係為

$$(\text{Speed A} * \text{A 的牙齒個數}) = (\text{Speed B} * \text{B 的牙齒個數})$$

假設這些齒輪都放置在一個 2D 的平面中。其中只有一個齒輪有動力。給你一群齒輪（註：齒輪與齒輪之間不會重疊，但也不見得相連結），以及某一個齒輪 X ，請根據齒輪彼此互相接連的狀況來計算出 X 的旋轉方向 (clock or clockwise) 以及旋轉的速度為何。

輸入格式

一組測試資料以一個整數 N 作為開頭。 N 是齒輪的個數。接著是 N 列的齒輪資料，每個齒輪的資料為 (x, y, R, n) 分別以空白隔開。 (x, y) 代表齒輪中心在 2D 平面上的座標， R 代表齒輪的半徑（不包含一公分的牙齒），而 n 則是牙齒的個數。第一個齒輪編號為 1，以此類推往後遞增。

齒輪的資料完畢之後，是 1 號齒輪的旋轉方向 (D) 以及旋轉速度 (V)。所有齒輪中，只有 1 號齒輪具備有動力。旋轉方向以 1 或 -1 來表示。1 代表順時鐘，-1 代表逆時鐘。另外旋轉速度以一秒鐘幾轉來表示。例如 $V=5$ 代表第一個齒輪每一秒鐘轉 5 圈。

測試資料的最後是一個整數 X ($1 < X \leq N$)。 X 代表我們要查詢的齒輪編號。

輸出格式

請輸出齒輪 X 的旋轉方向 (D) 以及速度 (V)，中間以空白隔開。旋轉方向一樣以 1 或 -1 來表示。 V 的值如果有小數點，請輸出小數點以下兩位即可。若齒輪 X 沒有能夠接收任何齒輪 1 號傳來的動力，請輸出 0 0。

輸入範例

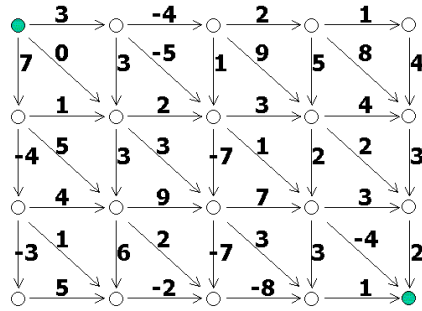
```
2
0 0 10 10
16 0 5 5
1 20
2
```

輸出範例

```
-1 40
```

3. 路徑選擇問題

在一個線上遊戲裡有如下圖的路徑，每個角色由左上角出發，目標在右下角，每一步可以向右、向右下、或是向下走，在每一步路徑上有可以加分的寶物或是扣分的陷阱，加減分數標在該路徑上，起始分數為 0，請注意一旦累加分數低於 0，該路徑就不能繼續走，請尋找一條分數一直保持在 0 以上且分數最高的路徑



以下是上圖的參考測試資料檔案內容，數字間以空白分隔

4 5	共 4 列 共 5 行
3 -4 2 1	第一列向右的分數
0 -5 9 8	一二列間斜線上的分數
7 3 1 5 4	第一列到第二列向下的分數
1 2 3 4	第二列向右的分數
5 3 1 2	二三列間斜線上的分數
-4 3 -7 2 3	...
4 9 7 3	
1 2 3 -4	
-3 6 -7 3 2	
5 -2 -8 1	

請撰寫程式，範例輸出如下：

```
maximum score = 33    path length = 6
(0,0,0) (1,0,7) (2,1,12) (2,2,21) (2,3,28) (2,4,31) (3,4,33)
```

上述第一列中 33 為最高分路徑的分數，路徑的長度為 6，第二列為路徑座標: (0, 0), (1, 0), (2, 1), (2, 2), (2, 3), (2, 4), (3, 4)，其中 (0, 0) 代表左上角, (3, 4) 為右下角，每一個括號中第三個數字代表由 (0,0) 至該座標部份路徑所得到的分數，

下圖為另一參考測試資料 (撰寫時請盡量一般化，實際測試資料中可能有數百列，數百行的資料)

```
5 5
-40 0 3 -50
1 -10 4 4
-30 -10 0 0 -40
0 -40 2 3
-30 0 3 2
-10 -30 2 2 -30
-40 3 4 3
0 0 2 4
1 2 0 4 3
3 -20 2 -30
2 2 -10 -40
2 2 1 0 -10
-30 4 4 4
```

```
maximum score = 13    path length = 6
(0,0,0) (1,1,1) (2,2,1) (2,3,5) (3,3,9) (4,3,9) (4,4,13)
```

4. 物體排序

王老師教過排序的方法後，要學生撰寫程式來對 n 個物體排序，然後王老師會對學生得到的結果評分。整個問題具體描述如下：有 n 個物體以 x_1, x_2, \dots, x_n 表

示，每個物體都有不同的權重，假設 $w(x_1)$ 表示 x_1 的權重， $w(x_2)$ 表示 x_2 的權重，...，依此類推。王老師要學生設計程式來對這 n 個物體依其權重由小至大來排序，假設排序後的結果為 r_1, r_2, \dots, r_n ，顯然正確的排序結果必須滿足 $w(r_1) < w(r_2) < \dots < w(r_n)$ 。但現在王老師要對學生所得到的排序結果訂定一個較寬鬆的給分標準，並不是要完全正確才給分，而是可以有部份分數，為清楚描述這個給分標準，先定義兩個名詞：由 r_1, r_2, \dots, r_n 中依序選取 m 個物體 ($1 \leq m \leq n$)，令其為 s_1, s_2, \dots, s_m ，則 s_1, s_2, \dots, s_m 稱為「子序列」，而若 $w(s_1) < w(s_2) < \dots < w(s_m)$ ，則其稱為「正確子序列」。如此一來，給分的標準為：這排序結果中可以找到的正確子序列之最大長度即為所得之分數。例如：有四個物體 ($n=4$) 其權重分別為 1 2 3 4 (即 $w(x_1)=1, w(x_2)=2, w(x_3)=3, w(x_4)=4$)，若排序的結果若是 2 1 3 4 (即 x_2, x_1, x_3, x_4) 就得 3 分，因為 2 3 4 ($w(x_2) < w(x_3) < w(x_4)$) 或 1 3 4 ($w(x_1) < w(x_3) < w(x_4)$) 為可以找到的最長正確子序列，而其長度是 3。

現在請你撰寫程式依王老師給的這個標準，來評定學生所得到的排序結果之分數。

輸入：第一列為物體個數；第二列依序為每個物體的權重(以空格隔開)；第三列為學生設計的程式所產生的排序結果(以空格隔開)。

輸出：學生得到的分數，後面接著以括號列出所有最大長度的正確子序列。

範例 1：

輸入：

4

4 2 3 1

3 2 1 4

輸出：

2 (3 1, 2 1)

範例 2：

輸入：

5

3 1 2 4 5

1 2 3 4 5

輸出：

4 (1 2 4 5)