

商管程式設計 (106-1)

第一次期中考

題目設計：孔令傑

國立臺灣大學資訊管理學系

請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為每一題各上傳一份 Python 3.6 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。作弊被確認者，整門課的成績將直接被評定為 F，沒有第二種可能 (也不能停修)。

所有的分數都根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、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。在所有的題目中，你都可以使用任何方法 (包括課堂上沒教過的)。這次考試的截止時間是 10 月 16 日下午 12 點 10 分。

注意：因為一些不明原因，在 PDOGS 上使用 list 的 `split(' ')`，可能會被錯誤地執行而達不到應有的效果。因此在這次考試中，請使用 `split()` 而不要用 `split(' ')`。

第一題

(20 分) 給定一個四位整數 $x = a_3 \times 10^3 + a_2 \times 10^2 + a_1 \times 10 + a_0$ ，其中 $a_3, a_2, a_1, a_0 \in \{0, \dots, 9\}$ ，我們想知道 x 是不是他中間那兩位數 $y = 10a_2 + a_1$ 的平方。舉例來說，若 $x = 2500$ ，則 $y = 50$ ，此時 $x = y^2$ ；若 $x = 1234$ ，則 $y = 23$ ，此時 $x \neq y^2$ 。請注意 x 的千分位數 a_3 可以是 0。在極端狀態下， $x = 0$ 也是可以的，此時 $y = 0$ ，我們也有 $x = y^2$ 。

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，會有四個相連的整數 a_3, a_2, a_1, a_0 ，其中 $a_i \in \{0, \dots, 9\}$ ，合起來代表一個介於 0 到 9999 之間的整數 x 。讀入 x 之後，請輸出他的 $y = 10a_2 + a_1$ ，接著輸出一個空白字元，再接著輸出一個整數 1 或 0：如果 $x = y^2$ 則輸出 1，反之則輸出 0。

舉例來說，如果輸入是

2500

則輸出應該是

50 1

如果輸入是

1234

則輸出應該是

23 0

第二題

(20 分) 給定一個含有 n 個整數 x_1, x_2 直到 x_n 的數列，以及兩個整數 y 和 z ，我們想要從這個數列中把其值為 y 的數字去掉，然後把這個數列根據 z 的值順著或反著印出來：如果 $z = 1$ 則照原本的順序印出來， $z = 0$ 則反著印出來。舉例來說，如果 $n = 7, x = (7, 5, 3, 3, 9, 6, 7), y = 7, z = 1$ ，則印出 $5, 3, 3, 9, 6$ ；如果 n 和 x 不變而 $y = 0, z = 0$ ，則印出 $7, 6, 9, 3, 3, 5, 7$ 。

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，第一列是 n 個整數 x_1, x_2 直到 x_n ，其中 $2 \leq n \leq 100$ ，並且我們知道 $-100 \leq x_i \leq 100$ 。第二列是兩個整數 y 跟 z ，其中 $-100 \leq y \leq 100$ 且 $z \in \{0, 1\}$ 。任兩個數字之間被一個空白鍵隔開。讀入這些資料之後，請照題目規定印出一個數列，任兩個數字之間用一個空白字元隔開。如果沒有任何數字應該被印出，就印出一個空字串。

舉例來說，如果輸入是

```
7 5 3 3 9 6 7
7 1
```

則輸出應該是

```
5 3 3 9 6
```

如果輸入是

```
7 5 3 3 9 6 7
0 0
```

則輸出應該是

```
7 6 9 3 3 5 7
```

第三題

(30 分) 你想要從你家出發去搭火車從 A 站到 B 站。你已知你從你家到 A 站要 x 分鐘、從 B 站到你最終目的地要 y 分鐘，以及你必須在距離現在 z 分鐘內抵達目的地。你手上有一份電子火車時刻表，上面列著 n 個車次，車次 i 記錄著一個最多四位數字的車次編號 q_i 、在 A 站的發車時間（以距離現在有幾分鐘記錄） s_i ，以及在 B 站的到達時間（以距離現在有幾分鐘記錄） t_i 。單一車次的 A 站發車時間一定早於在 B 站的到達時間，但在 A 站發車較早的車次未必比發車較晚的車次早到達 B 站。已知所有車次在 A 站的發車時間都不一樣，且由早到晚排列。請幫你自己在你趕得上且能讓你及時抵達目的地的車次中，找出最晚從 A 站發車的車次編號，以及你應該幾分鐘後從家裡出發才能在出發的 x 分鐘後恰好抵達 A 站。

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 $n + 1$ 行，第一行包含四個整數 n 、 x 、 y 、 z ，第二行至第 $n + 1$ 行的第 $i + 1$ 行包含 q_i 、 s_i 到 t_i 等三個整數。同一行的任意兩個整數之間被一個空白隔開。已知 $1 \leq n \leq 100$ 、 $1 \leq q_i \leq 9999$ 、所有其他數字都介於 0 到 1440 之間，以及 $s_1 < s_2 < \dots < s_n$ 。讀入資料後，請先印出最晚從 A 站發車的能讓你趕上且準時抵達目的地的車次編號，接著印出一個空白字元，然後印出你應該幾分鐘後從家裡出發。如果沒有任何一個班次能讓你準時抵達，就印出一個字元 0 就好。

舉例來說，如果輸入是

```
6 15 10 200
624 50 80
516 67 100
513 92 98
9877 99 160
11 120 190
81 155 191
```

則輸出應該是

```
11 105
```

請注意 $190 + 10 = 200$ ，恰好在最後一刻抵達目的地是可以的。又例如如果輸入是

```
2 15 10 200
624 12 80
516 150 195
```

則輸出應該是

```
0
```

因為班次 624 你趕不上，而班次 516 又來不及抵達目的地。

第四題

(30 分) 在之前的作業中，我們介紹過一個讓銷售員選擇移動路線的問題。在本題中，我們要基於當時的問題來探討一個新的路線選擇方法。底下我們把整個問題（的精簡版）重新描述一次。

在一個城市裡，有一個銷售員，他要拜訪此城市裡的 n 個地點。地點 i 和地點 j 之間的最短距離已知為 D_{ij} 。我們假設從地點 i 到地點 j 的距離和從地點 j 到地點 i 一樣，因此如果我們收集 D_{ij} 以組成一個 $n \times n$ 的方形矩陣 D ，則 D 是對稱的。舉例來說，假設共有四個客戶，且

$$D = \begin{bmatrix} 0 & 4 & 6 & 3 \\ 4 & 0 & 8 & 5 \\ 6 & 8 & 0 & 2 \\ 3 & 5 & 2 & 0 \end{bmatrix},$$

就表示地點 1 跟 2 之間的距離是 4、地點 2 跟 3 之間的距離是 8，依此類推。在本題中，我們假設三點之間的距離不一定符合三角不等式，也就是 $D_{ij} + D_{jk} < D_{ik}$ 是有可能的。

銷售員的任務是從地點 1 拜訪每個地點恰好一次，然後回到地點 1。銷售員決定使用兩個簡單的演算法。方法一是作業寫過的「每次都在沒去過的中去離現在位置最近的；平手就選編號最小的。」因為這個方法未必最佳，所以銷售員決定也試試第二種方法：「每次都在沒去過的中去離現在位置第二近的；平手就選編號最小的。」給定一個矩陣 D ，銷售員決定用兩個方法都排一條路線，然後挑總距離比較短的那一條去走。以上面的 D 為例，方法一會得到的移動順序是 1、4、3、2，方法二則是會得到 1、2、3、4。因為兩條路線的總距離都是 17，銷售員可以選任意一條。需注意所謂「第二近」是要跳過所有同樣都最近的選項。舉例來說，如果現在有五個未拜訪客戶，和銷售員的距離分別是 2、2、4、4、7，則我們定義第二近的客戶為距離為 4 的那兩位，我們應該從這兩位中選擇客戶編號較小的去拜訪。

在本題中，資料輸入的格式和那次作業的輸入格式一模一樣。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 $n + 1$ 行，第一行包含一個整數 n ，第二行至第 $n + 1$ 行的第 $i + 1$ 行包含 $D_{i,1}$ 到 $D_{i,n}$ 等 n 個整數。同一行的任意兩個整數之間被一個空白隔開。已知 $2 \leq n \leq 1000$ 、 $1 \leq D_{ij} \leq 10000$ 、 $D_{ii} = 0$ 。

讀入資料後，請按照題目指定的規則，在第一行依序印出路線一中銷售員拜訪的地點編號（從地點 1 開始），並在最後印出總移動距離。任兩個數字之間用一個空白字元隔開。接著在第二行依序印出路線二中銷售員拜訪的地點編號（從地點 1 開始），並在最後印出總移動距離。任兩個數字之間用一個空白字元隔開。最後，請在第三行印出比較短的那條路線的編號，路線一短則印出 1，路線二短則印出 2，平手則印出 0。舉例來說，如果輸入是

```
4
0 4 6 3
4 0 8 5
6 8 0 2
3 5 2 0
```

則輸出應該是

```
1 4 3 2 17
1 2 3 4 17
0
```