

# 商管程式設計 (110-2)

## 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 Python 3.9 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 **2021 年 10 月 18 日晚上九點**。在你開始前，請閱讀課本的第五、七章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是吳恩淇。

### 第一題

(20 分) 餐飲外送公司 Foodsoda 為提升其員工之工作效率，設立了以下的獎勵機制：完成一筆訂單的運送可得  $x$  元報酬，單日累積滿  $n$  單，自  $n + 1$  單起，每單可得  $y$  元報酬，其中  $y \geq x$ 。更精確地說，若  $r$  為某外送員完成的訂單數，若  $r \geq n$ ，則總報酬為  $n \times x + (r - n) \times y$  元。舉例來說，若公司規定每單報酬為 50 元，單日滿 10 單後，每單可得 100 元，則該外送員若當日接了 10 單，可得 500 元報酬；若該外送員接了 20 單，則總報酬為  $50 \times 10 + 100 \times 10 = 1500$  元。

### 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案含有 4 行，每行皆含有一個整數，依序是  $n$ 、 $x$ 、 $y$ 、 $r$ 。已知  $0 \leq n \leq 100$ 、 $1 \leq x \leq 100$ 、 $1 \leq y \leq 500$ 、 $1 \leq r \leq 500$ 。讀入這些資訊後，請輸出該送貨員單日可得總報酬。

舉例來說，如果輸入是

```
10
50
100
10
```

則輸出應該是

```
500
```

如果輸入是

```
10
50
100
20
```

<sup>1</sup>課本是 A. Downey 所著的 *Think Python 2*，在 <http://greenteapress.com/wp/think-python-2e/> 可以下載。

則輸出應該是

1500

如果輸入是

6  
90  
130  
15

則輸出應該是

1710

如果輸入是

15  
59  
200  
19

則輸出應該是

1685

## 你上傳的原始碼裡應該包含什麼

你的.py 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 Python 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 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。

## 第二題

(30 分) 承第一題，現在新增一條獎勵規定，單週累積滿  $m$  單，可額外獲得獎金  $b$  元。舉例來說，若同樣一單報酬為 50 元，單日滿 10 單後，每單可得 100 元，且單週滿 100 單，可額外獲得獎金 200 元，若該外送員在星期一到日分別接下 12、9、11、5、17、20 和 12 單，則該週累積總報酬為  $(10 + 9 + 10 + 5 + 10 + 10 + 10) \times 50 + (2 + 1 + 7 + 10 + 2) \times 100 = 5400$  元；若隔週該外送員奮發圖強，星期一到日依序接下 15、9、14、8、21、20 和 13 單，達到單週 100 單的規定，則該週總報酬為  $(10 + 9 + 10 + 8 + 10 + 10 + 10) \times 50 + (5 + 4 + 11 + 10 + 3) \times 100 + 200 = 6850$  元。

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案含有 12 行，每行皆含有一個整數。依序是  $n$ 、 $m$ 、 $x$ 、 $y$ 、 $b$ 、 $r_1$ 、 $r_2$ 、 $r_3$ 、 $r_4$ 、 $r_5$ 、 $r_6$ 、 $r_7$ 。已知  $0 \leq n \leq 100$ 、 $1 \leq m \leq 500$ 、 $1 \leq x \leq 100$ 、 $1 \leq y \leq 500$ 、 $1 \leq b \leq 1000$ 、 $1 \leq r_i \leq 500$ 。讀入這些資訊後，請輸出該送貨員單週可得總報酬。

舉例來說，如果輸入是

```
10
100
50
100
200
12
9
11
5
17
20
12
```

則輸出應該是

```
5400
```

如果輸入是

```
10
100
50
100
200
15
9
14
8
21
20
13
```

則輸出應該是

```
6850
```

如果輸入是

```
6
```

```
70
90
130
150
5
7
10
6
15
20
13
```

則輸出應該是

```
8390
```

如果輸入是

```
15
97
59
200
386
14
10
15
11
7
18
21
```

則輸出應該是

```
6933
```

## 你上傳的原始碼裡應該包含什麼

你的.py 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 Python 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 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。

## 第三題

(50 分) 你的商店賣一個商品，而你必須幫這一個商品定價。除此之外，你也要決定這個商品的包裝方式。你一共有  $n$  種方式可以選。假設用第  $i$  種包裝方式且價格為  $p$ ，則需求數量為

$$D_i(p) = \max\{a_i - b_i p, 0\},$$

其中  $a_i > 0$  和  $b_i > 0$  各是此商品在包裝方式  $i$  下的基本需求量和價格敏感度。用第  $i$  種包裝方式的單位生產成本是  $c_i > 0$ 。

在本題中，你將被給定  $n$ 、 $a_1$ 、 $b_1$ 、 $c_1$  直到  $a_n$ 、 $b_n$ 、 $c_n$ ，而你的任務是找出能最大化商品利潤的最佳包裝方法  $i^*$ 、最佳整數價格  $p^*$ ，以及被最大化的總利潤。在一種包裝方法中，如果有複數個價格都能得到最高的利潤，請選擇比較大的價格；如果兩種包裝方法的最佳價格能得到一樣高的利潤，請選擇用編號比較小的包裝方式。

### 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案含有  $3n + 1$  行，每行含有一個正整數，依序是  $n$ 、 $a_1$ 、 $b_1$ 、 $c_1$ 、 $a_2$ 、 $b_2$ 、 $c_2$  一直到  $a_n$ 、 $b_n$ 、 $c_n$ 。已知任一個數字都介於 1 和 1000 之間。請注意不無可能有包裝方式是完全無法獲利的（例如  $c_i > a_i$ ），此時請把此種包裝方式的最佳價格設為 1000、利潤設為 0。

讀入這些數字之後，請依照題目指定的規則，印出最佳包裝方式、最佳整數價格與得到的總利潤，共三個數字。在包裝方式方面，如果最佳解是用方案  $i$ ，請輸出  $i$ 。任意兩個整數之間用一個逗點隔開。

舉例來說，如果輸入是

```
2
80
3
20
100
2
40
```

則輸出應該是

```
2,45,50
```

如果輸入是

```
3
10
2
100
10
2
100
```

```
10
2
100
```

則輸出應該是

```
1,1000,0
```

## 你上傳的原始碼裡應該包含什麼

你的.py 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 Python 3.9 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 `for`、`while`，以及之前作業說過可以使用的語法。
- 確定不可以使用的語法包含清單、自定義函數、tuple、dictionary、利用 `print` 的任何格式化輸出法（例如百分比、f-string、`str.format()`）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。本題共有 15 組測試資料，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可讀性（包含排版、變數命名、註解等等）。請寫一個「好」的程式吧！

## 第四題（加分題）

（20 分）你經營一家公司賣加工玩具車，旗下有兩個員工，第一個員工負責把買進來的原始玩具車加上你獨門研發的飛行器，第二個員工負責包裝、放進紙箱、貼郵寄標籤。你今天收到了  $n$  張訂單，而對第  $i$  張訂單，我們知道它訂購的玩具車輛數  $x_i$  以及該張訂單的總價  $p_i$ ，表示客戶  $i$  想要用  $p_i$  元購買  $x_i$  輛車。你的每個員工每天各可以工作 8 小時，每張訂單都要依序經過兩個員工。第一個員工從早上八點工作到下午四點，第二個員工從下午四點工作到午夜十二點，所以沒有任何工作銜接（員工二因為要等員工一而閒置）的問題。員工一需要  $q_1$  分鐘處理一輛玩具車，員工二則需要  $q_2$  分鐘。

因為總工時有限，你未必能接受所有訂單，所以你設定了一個挑選訂單的方法。你將這  $n$  張訂單依照收到的時間先後編號為 1、2 直到  $n$ ，接著從第 1 張訂單開始逐一挑選。針對第  $i$  張訂單，如果被訂購的玩具車單價低於  $t$ ，就一定不挑選；如果單價等於或高於  $t$ ，就看兩位員工是否都還有時間完成這張訂單，若有則接受，任何一位員工的剩餘工時無法完成這張訂單則不挑選。你想知道面對眼前的  $n$  張訂單，照這個規則你應該接受哪幾張、員工最後各剩下多少時間不用工作，以及公司總營收是多少。

舉例來說，假設  $t = 100$ 、 $q_1 = 5$ 、 $q_2 = 10$ 、 $n = 5$ ，且訂單資訊依序是  $(x_1, p_1) = (20, 3000)$ 、 $(x_2, p_2) = (15, 1500)$ 、 $(x_3, p_3) = (10, 800)$ 、 $(x_4, p_4) = (15, 3000)$  和  $(x_5, p_5) = (5, 6000)$ 。則訂單挑選依序如下：

- 第 1 張：兩位員工各剩 480 分鐘。訂單中車子單價  $\frac{3000}{20} \geq 100$ ，且 20 輛車要花掉兩位員工各 100、200 分鐘，時間還夠，所以挑選這張訂單。
- 第 2 張：兩位員工各剩 380 與 280 分鐘。訂單中車子單價  $\frac{1500}{15} \geq 100$ ，且 15 輛車要花掉兩位員工各 75、150 分鐘，時間還夠，所以挑選這張訂單。
- 第 3 張：兩位員工各剩 305 與 130 分鐘。訂單中車子單價  $\frac{800}{10} < 100$ ，所以不挑選這張訂單。
- 第 4 張：兩位員工各剩 305 與 130 分鐘。訂單中車子單價  $\frac{3000}{15} \geq 100$ ，且 15 輛車要花掉兩位員工各 75、150 分鐘，時間不夠，所以不挑選這張訂單。
- 第 5 張：兩位員工各剩 305 與 130 分鐘。訂單中車子單價  $\frac{6000}{5} \geq 100$ ，且 5 輛車要花掉兩位員工各 25、50 分鐘，時間還夠，所以挑選這張訂單。

最終依序挑選訂單 1、2、5，兩位員工各剩下 280 與 80 分鐘，公司總營收  $3000 + 1500 + 6000 = 10500$ 。

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案含有  $2n + 4$  列，每列皆含有一個整數。依序是  $n$ 、 $t$ 、 $q_1$ 、 $q_2$ 、 $x_1$ 、 $p_1$ 、 $x_2$ 、 $p_2$  直到  $x_n$ 、 $p_n$ 。已知  $0 \leq n \leq 100$ 、 $1 \leq t \leq 1000$ 、 $1 \leq q_1 \leq 600$ 、 $1 \leq q_2 \leq 600$ 、 $1 \leq x_i \leq 100$ 、 $1 \leq p_i \leq 100000$ 。讀入這些資訊後，請輸出兩列數字，第一列是由小到大排序的應該被接受的訂單編號，兩兩之間以一個逗點隔開；第二列是三個數字，依序是員工一剩下不用工作的分鐘數、員工二剩下不用工作的分鐘數，以及公司總營收。如果所有訂單都不挑選，第一列請印出 -1，第二列請照既有規則印出。

舉例來說，如果輸入是

```
5
100
5
10
20
3000
15
1500
10
800
15
3000
5
6000
```

則輸出應該是

```
1,2,5
280,80,10500
```

如果輸入是

```
5
100
600
600
20
3000
15
1500
10
800
15
3000
5
6000
```

則輸出應該是

```
-1
480,480,0
```

如果輸入是

```
5
100
5
10
20
3000
15
1500
10
1200
15
3000
5
6000
```

則輸出應該是

```
1,2,3
255,30,5700
```



## 你上傳的原始碼裡應該包含什麼

你的.py 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 Python 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 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。