

# 商管程式設計 (110-1)

## 作業五

### 作業設計：盧信銘

### 國立臺灣大學資訊管理學系

繳交作業時，請至PDOGS (<http://pdogs.ntu.im/> (<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11月15日晚上九點。為這份作業設計測試資料並且提供解答的助教是彭晨。

## 第一題

阿花生活在新世界，而新世界正流行一種叫元病毒的疾病。阿花收集了每日確診的人數，並且想要知道哪些區段的確診人數是連續 $K(K > 2)$ 天或更久的持續上升的。比如說 `[0, 0, 1, 5, 200, 400, 423, 1282, 523, 7, 2]` 這個數列裡，`0, 1, 5, 200, 400, 423, 1282` 這個區段是連續上升。這個區段包含7個元素，我們稱為連續6天上升。對應到List的Index為由1開始，至7結束。

請幫阿花寫一個名為`mono_inc`的函數，這個函數有兩個Parameters，依順序分別是 `inlist` 與 `k`。其中 `inlist` 是一個內含確診人數的 `List`，`k` 則是要找尋連續 $k$ 天(或更久)上升的區段。`k` 的預設值是 3。這個函數將會找出第一個連續 $k$ 天(或更久)上升的區段，並回傳 `[start, end]`。其中 `start` 是這個區段的起始Index，`end` 是結束Index。如果沒有符合條件的區段，`mono_inc`回傳`None`。

## 輸入輸出格式

系統將會提供測試資料。輸入第一行為確診人數數列，格式為逗點分隔的非負整數，單一數值在十億以下，數列總長度不會超過五千個元素。你的程式應該在取得這些資料後轉換成整數元素構成的List。第二行為 $k$ 值。如果  $k \geq 3$ ，則呼叫`mono_inc`時將確診人數數列與 $k$ 值傳入；如果  $k < 3$  時則使用default值  $k = 3$ ，並傳入確診人數數列。

如果`mono_inc`回傳`None`，則印出 `None`。如果`mono_inc`回傳區段Index，則傳回值分兩行印出，第一行印出區段的起始Index，第二行印出區段結束Index

舉例而言，如果輸入是：

```
0,1,3,3,7,6,7,8,10,9,9,9,9,7,8,7,9,7,8,8
3
```

那輸出是:

5

8

如果輸入是:

0,1,2,3,4,5

3

那輸出是:

0

5

如果輸入是:

0,1,2,3,4

5

那輸出是:

None

## 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。上傳至PDOGS的程式碼中**不能import任何套件**。

## 評分原則

- 這一題的其中24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12 組測試資料，一筆測試資料佔2分。
- 這一題的其中10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！

## 第二題

接續上一題的設定，寫一個mono\_inc\_plus函數，這個函數的Parameters跟mono\_inc一樣。不同的地方是mono\_inc\_plus回傳所有的連續 $K(K>2)$ 天或更久的持續上升的區段。

舉例而言，以下的List

```
[0, 1, 2, 3, 0, 1, 2, 3, 4]
```

傳進mono\_inc之後會回傳：

```
[0, 3]
```

但其實這個List中有兩個連續上升的區段。第二個區段在  $[4, 8]$ 。因此，mono\_inc\_plus要傳回這兩個區段。傳回的方式使用兩層的List, 也就是  $[[0, 3], [4, 8]]$ 。同理，如果更多遞增的區段，則將所有區段放在一個List中回傳。例如如果有五個區段，分別為  $[0, 4], [12, 18], [21, 33], [102, 125], [212, 342]$ ，則回傳：

```
[[0, 4],  
 [12, 18],  
 [21, 33],  
 [102, 125],  
 [212, 342]]
```

## 輸入輸出格式

系統將會提供測試資料。輸入第一行為確診人數數列，格式為逗點分隔的非負整數，單一數值在十億以下，數列總長度不會超過五千個元素。你的程式應該在取得這些資料後轉換成整數元素構成的List。第二行為 $k$ 值。如果  $k \geq 3$ ，則呼叫mono\_inc\_plus時將確診人數數列與 $k$ 值傳入；如果  $k < 3$  時則使用default值  $k = 3$ ，並傳入確診人數數列。

如果mono\_inc\_plus回傳None, 則印出 None。否則則印出分別印出所有的起始與結束區段，每一行是一組開始與結束Index, 中間以空格分開。

舉例而言，如果輸入是：

0,1,3,4,8,8,11,13,17,18,20,22,23,23,25,25,28,28,30,32,36,37,37,41,41,45,48,49,53,56,58,61,62,65,67,70,71,73,76,78

3

輸出是:

0 4

5 12

17 21

24 39

如果輸入是:

6,5,4,3,2,1

3

那輸出是:

None

如果輸入是:

1,2,3,4,5,6,7,8

3

那輸出是 :

## 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。上傳至PDOGS的程式碼中**不能import其他套件**。

## 評分原則

- 這一題的16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8 組測試資料，一筆測試資料佔2分。

## 第三題

Herfindahl–Hirschman Index (HHI)是一個計算產業競爭程度的指標。假設某個產業有N家廠商，其營收為 $Y_1, Y_2, \dots, Y_N$ ，廠商i的市場占有率為 $s_i = Y_i / \sum_{i=1}^N |Y_i|$ 。基於這些設定，我們有兩種計算HHI的方式，第一種是：

$$HHI = \sum_{i=1}^N s_i^2$$

第二種計算方式為：

$$HHI^* = \frac{HHI - 1/N}{1 - 1/N}$$

我們稱第一種方式為"Original"，第二種方式為"Normalized"。

寫一個名為my\_hhi的函數來計算HHI。這個函數有兩個Parameters，第一個為ylist，沒有預設值，預計接收包含N個元素的List，每個元素為一家廠商的營收。第二個為htype，預設值為"Original"。如果htype為"Original"，則計算第一種HHI。如果htype不是"Original"，則計算第二種HHI。計算完成後將結果回傳(i.e., return)。

## 輸入輸出格式

系統將會提供測試資料。輸入第一行為N家廠商營收，格式為逗點分隔的數值，單一數值絕對值在十億以內，數列總長度不會超過五千個元素。你的程式應該在取得這些資料後轉換成浮點數元素構成的List，第二行為計算HHI的方式，可能的數值為"Original"或"Normalized"。

在取得這些輸入資料，並進行適當前處理之後，你的程式接下來應呼叫my\_hhi，並傳入營收List與HHI計算方式。為確保PDOGS可以正確批改，my\_hhi傳回值應以 printresult 函數列印之。這個函數會將結果印出來，四捨五入至小數點第四位。printresult 函數的定義如下：

```
def printresult(value):  
    print(f"{value:.4f}")
```

printresult 函數用到格式化輸出的語法，將在後續週次才會介紹。目前你只要將函數定義拷貝至你的程式碼，並呼叫使用即可。

舉例而言，如果輸入是：

1235.2  
Original

那輸出是:

1.0000

如果輸入是:

123,456  
Original

那輸出是:

0.6654

如果輸入是:

123,456  
Normalized

那輸出是:

0.3308

## 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。上傳至PDOGS的程式碼中**不能import其他套件**。

## 評分原則

- 這一題的26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13組測試資料，一筆測試資料佔2分。

## 第四題

大雄參與了一個分析賣場交易資料的研究計畫，他發現交易資料中除了一般的商品之外，也會有折扣資訊。比如說：

```
[["孔雀吉祥捲心禮盒378G",1,99],  
 ['肉鋪子-厚感蜜汁豬肉乾 205g',1,150],  
 ['華元 波的多洋芋片-蚵仔煎口味315g',2,230],  
 ['活動折扣',1,-23]]
```

上面是一個List包List的結構。第二層的List中每一筆資料中依序為：品項、數量與小計(單項總金額)。其中最後一筆資料的小計是-23，也就是消費者會享受23元的折扣。在與研究團隊討論之後，大雄認為合理的作法是將這個折扣依照各品項的總金額攤列。也就是說，在這次交易中，所有正項的總金額是  $99+150+230=479$ ，所以 孔雀吉祥捲心禮盒378G 應該享受的折扣為  $-23*(99/479)=-4.7536534$ 。照這個邏輯，這個品項的小計應該調整為  $99-4.7536534=94.2463466 \approx 94.25$ 。同樣的計算可以應用在其他金額為正的品項上。在完成調整之後，折扣項即可丟棄。

寫一個名為apply\_discount的函數。這個函數有一個Parameter，名為records，預計接收像上面這樣的雙層List。這個函數將所有小計為負的項目視為折扣，並應用前面的規則調整其他品項的小計，最後輸出(return)一個新的雙層List。輸出中資料的順序需與原始資料一致。要特別注意的是，在處理的過程中不應該改動原始輸入List的內容。因此在建構輸出時需善用List內建的 .copy() 方法，呼叫方法為 list.copy()。（附註：.copy() 是複製的方法。請同學在第一層的List中使用，也就是使用迴圈讀取每筆交易資料時，再用 單筆交易資料list.copy() 進行每筆交易資料的複製。請不要直接複製二維陣列，以免發生「淺複製」會發生的問題。）

## 輸入輸出格式

系統將會提供測試資料。第一行中有第一筆資料，第二行中有第二筆資料，依此類推。如果一行的輸入為「RECORDSTOP」，則代表所有的紀錄已經輸入完畢。每筆資料中包含品項、數量與小計。這三個欄位以 \_ 分隔。每次輸入的資料筆數不會超過1000筆。你可以假設輸入的品項不會重複。你的程式應該在取得這些資料後轉換成一個雙層的List，轉換的過程中需將數量與小計轉換成浮點數格式。前處理完成即可呼叫apply\_discount，並將這個雙層List傳給apply\_discount函數。

apply\_discount執行完畢之後，將其輸出的List以printrec函數印出。printrec的定義如下：

```
def printrec(records):  
    for arec in records:  
        # 避免浮點數誤差  
        if abs(arec[2]) < 1e-3:  
            arec[2] = 0  
        print(f"{arec[0]}_{arec[1]:.2f}_{arec[2]:.2f}")
```

printrec 函數用到格式化輸出的語法，將在後續週次才會介紹。目前你只要將函數定義拷貝至你的程式碼，並呼叫使用即可。

在列印完輸出之後，下一行應列印"---Original---"，換行後呼叫printrec繼續列印原始資料。

舉例而言，如果輸入是：

```
孔雀吉祥捲心禮盒378g_1.00_99.00  
肉鋪子-厚感蜜汁豬肉乾 205g_1.00_150.00  
華元 波的多洋芋片-蚵仔煎口味315g_2.00_230.00  
活動折扣_1.00_-23.00  
RECORDSTOP
```

那輸出應該是：

```
孔雀吉祥捲心禮盒378g_1.00_94.25  
肉鋪子-厚感蜜汁豬肉乾 205g_1.00_142.80  
華元 波的多洋芋片-蚵仔煎口味315g_2.00_218.96  
---Original---  
孔雀吉祥捲心禮盒378g_1.00_99.00  
肉鋪子-厚感蜜汁豬肉乾 205g_1.00_150.00  
華元 波的多洋芋片-蚵仔煎口味315g_2.00_230.00  
活動折扣_1.00_-23.00
```

如果輸入是：



```
立大膠柄草鐮刀_1.00_65.00
棉紗手套-6雙_1.00_79.00
活動折扣_1.00_-12.00
RECORDSTOP
```

那輸出應該是:

```
立大膠柄草鐮刀_1.00_59.58
棉紗手套-6雙_1.00_72.42
---Original---
立大膠柄草鐮刀_1.00_65.00
棉紗手套-6雙_1.00_79.00
活動折扣_1.00_-12.00
```

## 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，像是題目敘述提到的`list.copy()`。上傳至PDOGS的程式碼中不能import其他套件。

## 評分原則

- 這一題的24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12組測試資料，一筆測試資料佔2分。