

# 商管程式設計 (107-2)

## 作業九

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/judge/>) 為第一題上傳一個 PDF 檔、為第二題做同儕互評，再為第一題 (a)、第三題與第四題各上傳一份 Python 3 原始碼（以複製貼上原始碼的方式上傳）。第四題是 bonus 加分題。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。請以英文或中文作答。

這份作業的截止時間是 **5 月 27 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方心成。

### 第一題

(40 分) 請回答以下問題。

- (a) (10 分) 請寫一個 Python 3 程式，去建立一個簡單的視窗，裡面有一個文字方塊、一個下拉選單 (dropdown list) 和一個按鈕，文字方塊可以讓使用者輸入英文字母，下拉選單可以讓使用者選擇「全大寫」或「全小寫」，而當使用者按下按鈕，文字方塊內的文字就會根據下拉選單的選擇被改成全部大寫或全部小寫，然後取代這個文字方塊內的內容。你可以假設使用者只會輸入半形英文字母，字數不超過 10。請在 PDOGS 上傳你本題的程式碼。由於本題並沒有要使用自動批改，因此也不會有測試資料，你上傳後會直接看到「Accepted」以及 0 分，但只要你看得到自己上傳的程式碼，那就是有完成繳交了，助教會去檢閱你的程式碼的合理性和正確性並給分。
- (b) (10 分) 承上題，請在一份文件檔中貼上你創造的視窗的數個截圖，用這些截圖來展示你的程式，例如程式剛打開時的畫面、輸入文字的畫面、選擇全大寫後按按鈕的畫面、選擇全小寫後按按鈕的畫面，依此類推，然後和 (c) 一起存成一個 PDF 檔上傳到 PDOGS。
- (c) (20 分) 本學期至今我們也教了蠻多東西的，應該也讓一些同學從無到有地學會寫程式了。請考慮你本人所屬的科系，告訴我們一門系上必修課，是你認為適合在其中要求修課同學用我們有教的程式設計做點事的。那門課並不是要教程式設計，而是要「用」程式設計。請告訴我們是哪門課、那門課裡面的什麼任務適合融入程式設計，以及為什麼。請盡量不要寫超過三百字。完成後，和 (b) 一起存成一個 PDF 檔上傳到 PDOGS。

### 第二題

(0 分) 請在 PDOGS 上批改你被隨機分配到的作業八第三題的程式碼，根據它在正確性以外的部份給它 1 至 5 分的評分，並且說明你給分的依據。建議在評分時參考以下六個面向。在前五個面向上，一個面向上做得好就得一分，還不錯則半分，不好則零分；在第六個面向上則在有必要時扣分。六個面向的分數合計後無條件進入即為你最後給的總分。

- 可讀性：變數與函數名稱是否具有合適的資訊量？程式碼排版是否良好且具有前後一致性？是否有合適的註解？關於註解，當然不需要每一行都有註解，但若你發現在某一大段落裡都沒有註解，

或某個你覺得很不易看懂的部份沒有註解，你可以指出來；不要直接說「註解太少」但沒有說是哪邊缺乏註解。

- 模組化程度：是否有宣告合適的函數與 class？是否有避免將非常類似的程式片段寫複數次而非寫成函數？是否有避免一個函數做非常多事情？函數間是否有合適的 decoupling？直接閱讀程式是否能很快地理解程式在大方向上的運算邏輯？
- 效率：程式運算是否有合理的運算效率？當然我們不要求每個同學都寫出超級有效率的精妙演算法，但至少一個程式不應該進行過多不必要的運算，也不應該耗用過多不必要的記憶體空間。如果你看不出這個程式的效率有明顯的問題，我們建議你直接給一分。
- 擴充性：當要解的問題變得更複雜的時候，我們能不能簡單地修改這個程式以解決新的問題，而不是寧可砍掉重練？這個議題當然也很主觀，所以如果你不能明確地指出在怎樣的新問題上，這個程式會有擴充性問題，我們建議你直接給一分；如果你不能指出很嚴重的問題，我們建議你至少給半分。但對批改者來說，這個關於擴充性的思考其實是很好的訓練。試試看吧！
- 其他：如果有任何其他令你想扣分的理由，請明確地寫出來並且在這個面向上扣分；沒有的話就給一分。
- 題目規範：你應該檢查那份程式碼有沒有違反題目的規範，如果有（例如題目說不可以用上課沒教過的東西，但他用了，或者題目說一定要寫個函數，但他沒寫），就給他零分。當然，請明確地指出他哪邊違反了題目的規範。

本題其中 10 分取決於檢視你的程式碼的同學給你的分數總和（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性（原則上除非被申訴，且助教檢視後發現你確實評得很不公平，否則只要有評就會得到 10 分）

### 第三題

（60 分）給定  $n$  棵樹的高度  $x_1, x_2$  直到  $x_n$ ，我們可以選擇砍掉任意一棵，接著檢查剩下的  $n - 1$  棵樹的高度是否是「單調」的，亦即是否遞增（ $x_1 \leq x_2 \leq \dots \leq x_n$ ）或遞減（ $x_1 \geq x_2 \geq \dots \geq x_n$ ）。我們想知道有多少種砍掉一棵樹的砍法，會使剩下的  $n - 1$  棵樹的高度是單調的。顯然這個答案一定是 0 到  $n$  之間的一個整數，請求出它來。

這一題看起來是很簡單，就各種選項都試一次然後檢查一下就好。但身為整學期的最後一題作業，當然是沒有這麼單純。在這一題中，我們想請大家體驗一下好的跟不好的演算法（algorithm），可以在效能上有重大差別，亦即在運算時間上差異甚大。因此，這一題在 PDOGS 我們只期待你找到正確答案，但下一題（bonus）中，我們不僅會期望你能找到正確答案，還期望你能在夠短的時間內找到正確答案。為此，在下一題的測試資料中，助教會設計規模相當多的數字，並且給予相對嚴苛的執行秒數上限，讓你如果單純地實作了最天真的演算法，會超過秒數上限因此得不到分數。

題外話，這一題其實是趨勢科技徵聘今年暑假的實習軟體工程師時的線上測驗題目，趨勢科技就看看來應徵的人，能否分析題目並且找到高效率的演算法。大家也試試看吧，如果能順利完成，你也就通過趨勢科技的第一關面試了呢！

## 輸入輸出格式

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有兩行，第一行包含一個整數  $n$ ，第二行包含  $n$  個正整數  $x_1$ 、 $x_2$  直到  $x_n$ 。已知  $2 \leq n \leq 100$ 、 $1 \leq x_i \leq 999$ 。兩個整數之間用一個逗點隔開。

讀入資料後，請按照題目的指示，印出有幾種砍掉一棵樹的方法能讓剩餘的樹維持高度單調。舉例來說，如果輸入是

```
5
1 4 7 10 13
```

則輸出應該是

```
5
```

如果輸入是

```
5
1 4 7 3 10
```

則輸出應該是

```
1
```

如果輸入是

```
5
1 4 3 2 10
```

則輸出應該是

```
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 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，但上課介紹過的函式庫中所有的功能都可以用。

## 評分原則

- 這一題的其中 40 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。一筆測試資料佔 2 分。
- 這一題的其中 20 分會在作業十中被評定。屆時我們會讓同學們互相檢視彼此的本題程式碼，並且就可讀性、易維護性、模組化程度、排版等面向寫評語和給評分（當然一切都是匿名的）。該任務在本題中會佔 20 分，其中 10 分取決於檢視你的程式碼的同學給你的分數（必要時助教會出來主

持公道，請不用緊張)，另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性。若你在本次作業中完全沒有寫這一題，那屆時自然沒有人能檢視你的程式碼，你也就得要損失這 10 分了。

## 第四題 (bonus)

(20 分) 承上題，現在  $n$  和  $x_i$  的範圍擴大，變成  $2 \leq n \leq 1000000$ 、 $1 \leq x_i \leq 9999999$ 。

針對這個題目，你**可以**使用任何方法。這一題的 20 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。