商管程式設計（106-1）
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/judge/`）為第一、二題上傳一個 PDF 檔，再為第三題與第四題各上傳一份 Python 3.6 原始碼（以複製貼上原始碼的方式上傳）。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 ***2017 年 10 月 2 日凌晨一點***。在你開始前，請閱讀課本的第五、七章[1]。為這份作業設計測試資料並且提供解答的助教是薛光佑。

第一題

（20 分）請依照下列要求改寫給定的程式，以使你的程式會跑出和原本程式一模一樣的結果。你改寫後的程式如果含有冗贅的程式碼（例如內含 `a = a` 的 `if` 區塊、不可能被執行到的 `break` 之類的），或是效率較原本的程式碼有數量級上的差別，會被扣一些分。

(a)（5 分）請使用 ternary if 去改寫

```
a = int(input())
if a > 0:
  print(a)
else:
  print(-a)
```

讓你改寫後的程式一共只有兩行程式碼。

(b)（5 分）請加入 `break` 到下方程式的迴圈中

```
a = int(input())

while a != 1:
  if a % 2 == 1:
    a = 3 * a + 1
  else:
    a //= 2
  print(a, end = " ")
```

(c)（5 分）承上題，請刪掉 `break` 改成加入 `continue`。

(d)（5 分）請比較 (b) 小題的原始程式碼、用 `break` 改寫後的，以及用 `continue` 改寫後的，說說你覺得哪一個程式最好，並解釋原因。

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

第二題

（20 分）針對上課介紹的定價最佳化程式

```
a = int(input("base demand = "))
b = int(input("price sensitivity = "))
c = int(input("unit cost = "))

maxProfit = 0
optimalPrice = 0
for p in range(c + 1, a // b):
  profit = (a - b * p) * (p - c)

  if profit > maxProfit:
    maxProfit = profit
    optimalPrice = p

print("optimal price = " + str(optimalPrice))
print("maximized profit = " + str(maxProfit))
```

請回答下列各自獨立的問題。

(a)（5 分）請將此程式改成只使用 `while`，不使用 `for`。

(b)（5 分）假設現在我們不再允許商家將價格定在任意的整數值，而是只允許商家將價格設定為 5 的倍數。請改寫程式以滿足這個限制。當你寫出正確的程式後，你的程式若能執行愈少圈的迴圈運算，你就能得到愈高的分數。

(c)（10 分）假設商家可以投資在生產設備以降低單位生產成本。對於所有的 $x \in \{0, 1, ..., c-1\}$，若商家投資 dx^2 元，就可以將生產成本從 c 元降低到 $c-x$ 元。這裡 $d > 0$ 是投資效率係數，d 愈大表示需要愈多投資才能降低 1 元成本。舉例來說，若 $d = 10$，則投資 10 元可以將成本降低 1 元、投資 40 元可以降低 2 元、投資 90 元可以降低 3 元，依此類推。須注意若投資額介於 $d(x-1)^2$ 元和 dx^2 元之間，則單位成本只會下降 $x-1$ 元，例如在上例中，若投資 39 元，只能將成本降低 1 元。

請修改原本的程式，幫助商家找出能最大化總利潤的單位售價與生產設備投資金額，其中總利潤為銷貨利潤（銷貨營收減銷貨成本）減投資金額。

第三題

（60 分）你的商店賣兩個商品，而你必須幫這兩個商品定價。商品 1 的價格為 p_1，商品 2 則為 p_2。一旦價格被決定，商品 1 和商品 2 的需求量就各會是

$$q_1 = a_1 - b_1 p_1 + h p_2 \quad \text{和} \quad q_2 = a_2 - b_2 p_2 + h p_1，$$

其中 $a_i > 0$ 和 $b_i > 0$ 各是商品 i 的基本需求量和價格敏感度，而 $h \geq 0$ 則表現兩個商品間的相似程度（互相可取代性）：h 愈大則相似度愈高，反之亦然。商品 1 和商品 2 的單位生產成本各是 $c_1 > 0$ 和 $c_2 > 0$。只要兩個商品的需求量都不是負的，你就能隨意設定 p_1 和 p_2。

在本題中，你將被給定 a_1、a_2、b_1、b_2、c_1、c_2 和 h，而你的任務是找出能最大化兩個商品的利潤總和的最佳整數價格 p_1^* 和 p_2^*。

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。每個檔案含有七行，每行含有一個整數，依序是 a_1、a_2、b_1、b_2、c_1、c_2 和 h。已知任一個數字都介於 0 和 1000 之間，並且滿足 $a_1 - b_1c_1 + hc_2 > 0$、$a_2 - b_2c_2 + hc_1 > 0$、$b_1 > h$ 和 $b_2 > h$。讀入這些數字之後，請依照題目指定的規則，印出商品 1 和商品 2 的最佳價格 p_1^* 和 p_2^*，接著印出銷售兩個商品的總利潤，共三個數字。任意兩個整數之間用一個空白字元隔開。如果有複數個價格組合都能得到最高的利潤，請選擇 p_1 比較大的；如果有複數個 p_1 相同的價格組合都能得到最高的利潤，請選擇 p_2 比較大的。

舉例來說，如果輸入是

```
100
80
2
3
20
30
1
```

則輸出應該是

```
48 41 1315
```

如果輸入是

```
10
10
2
2
1
1
1
```

則輸出應該是

```
6 6 40
```

請注意在第二個例子中，如果我們允許價格為小數，則最佳價格應該是 5.5 和 5.5，但因為我們不允許小數，所以最佳價格是 6 和 6。5 和 5 會得到一樣多的利潤，但根據題目規則，我們會選擇 6 和 6。

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 3.6 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。

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運算邏輯、可讀性，以及可擴充性。請寫一個「好」的程式吧！

第四題（bonus）

（20 分）在一個二維平面上，我們用 $(0,0)$、$(a,0)$、$(0,a)$ 和 (a,a) 這四個點圍出一個正方形。在這個範圍內，我們一共有 $(a+1)^2$ 個格子點（x 和 y 座標都是整數的點）。在其中一些格子點上面躲了 3 個狙擊手：(x_1,y_1)、(x_2,y_2) 和 (x_3,y_3)。一個狙擊手可以狙擊跟他位在同一個橫線、直線或斜線的所有人。如果一個位置可以被任意一位狙擊手狙擊，我們就說它是「危險」的。更精確地說，如果位置 (x,y) 滿足下面任何一個條件：

- 對至少一個 $i \in \{1,2,3\}$ 滿足 $x = x_i$，
- 對至少一個 $i \in \{1,2,3\}$ 滿足 $y = y_i$，
- 對至少一個 $i \in \{1,2,3\}$ 滿足 $|x - x_i| = |y - y_i|$，

那我們就說這個位置是危險的。我們想要找出「安全」的位置的總數，而安全的定義就是不危險。

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。每個檔案含有七行，每行含有一個整數，依序是 a、x_1、y_1、x_2、y_2、x_3 和 y_3。已知任一個數字都介於 0 和 10 之間，並且滿足 $0 \le x_i \le a$、$0 \le y_i \le a$，$i \in \{1,2,3\}$。讀入這些數字之後，請依照題目指定的規則，依序印出安全位置的總數以及危險位置的總數。兩個整數之間用一個空白字元隔開。

舉例來說，如果輸入是

```
5
0
0
0
1
0
2
```

則輸出應該是

```
6 30
```

如果輸入是

```
9
3
3
4
5
5
7
```

則輸出應該是

```
27 73
```

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。

提示 1：Python 3 有個函數叫 `abs`，大家不妨看看課本（在課本的 PDF 檔上搜尋「abs」）或上網查查看（比如說去 Google 搜尋「python abs」），可能會幫上你的忙。

提示 2：使用所謂的 Python list 可能會有幫助，不過這一題並不需要這麼做。