**Python程式設計作業**

**範圍： 函式的應用**

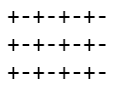
**銘傳大學電腦與通訊工程系**

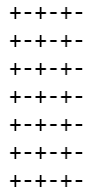
|  |  |
| --- | --- |
| 班 級 | 電通四乙 |
| 姓 名 | 鄒學緯 |
| 學 號 | 05050554 |
| 作業成果 | 應繳作業共 7 題，前3題每題20分，後4題每題10分，滿分為100分  我共完成 7 題，應得 100 分 |
| 授課教師 | 陳慶逸 |

* 請確實填寫自己寫完成題數，並且計算得分。填寫不實者(如上傳與作業明顯無關的答案，或是計算題數有誤者)，本次作業先扣50分。
* 確實填妥封面的內容，完成後請上傳pdf檔。

**【立即練習】**: 試實作一個函式hyberCub()，當使用者所輸入的兩個參數為h和w時，程式會交錯使用「+」和「-」列印一個長寬分別為h和w的長方形。若是函式沒有輸入參數時，則函式會以預設值:h = 6, w = 7來處理。

例如:

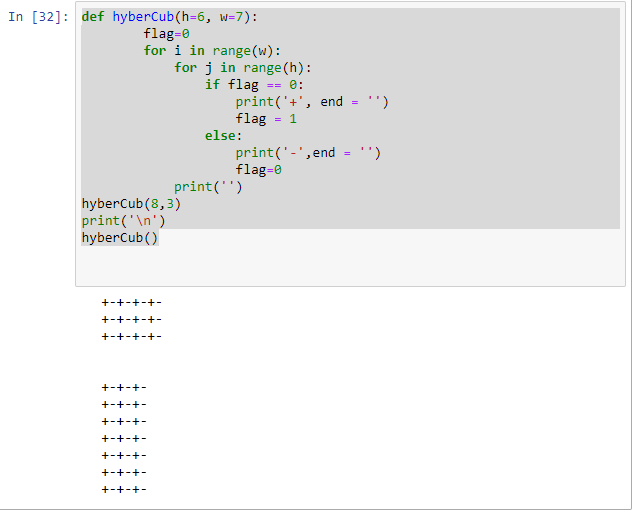
輸入hyberCub(8, 3)，則回傳: 

輸入hyberCub()，則回傳:

**程式碼：**

|  |
| --- |
| **def hyberCub(h=6, w=7):**  **flag=0**  **for i in range(w):**  **for j in range(h):**  **if flag == 0:**  **print('+', end = '')**  **flag = 1**  **else:**  **print('-',end = '')**  **flag=0**  **print('')**  **hyberCub(8,3)**  **print('\n')**  **hyberCub()** |

**執行結果擷圖：**



**【立即練習】**: 遞迴的目的在於分割問題，或將問題縮小以逐步降低問題的難度。一個問題如果可以拆解成規模較小但是性質和原問題完全一樣的問題的時候， 就可以考慮用遞迴函式來處理。舉例來說，m \* n的乘法運算我們便可以將之寫成m + m \* (n – 1)，因為m \* (n – 1)要比m \* n容易一點，在實作一個遞迴函式multiply(m ,n)時，我們重點在於計算出 x = m \* (n – 1)，因為 m + x便是計算結果了。

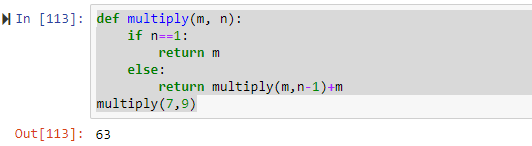
例如:

輸入multiply(6, 5)，則回傳30

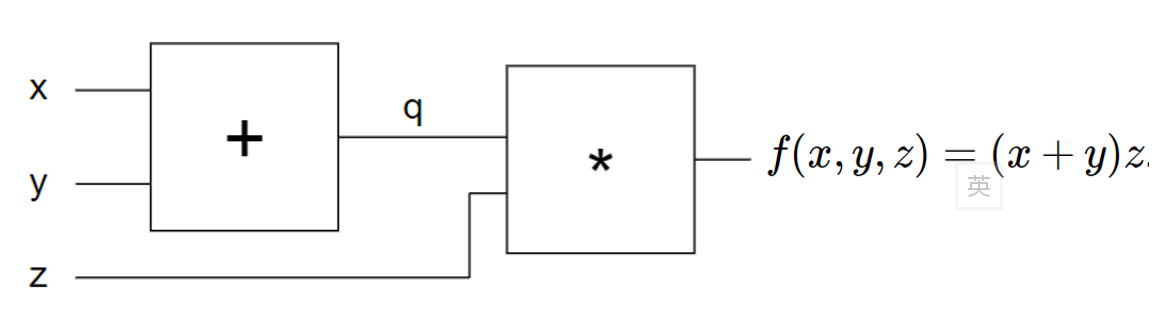
輸入multiply(7, 9)，則回傳63

|  |
| --- |
| **def multiply(m, n):**  **if n==1:**  **return m**  **else:**  **return multiply(m,n-1)+m**  **multiply(7,9)** |

**執行結果擷圖：**



**【立即練習】**:請以Lambda函式來實現實數相加(forwardAddGate(x, y))以及實數相乘(forwardMultGate(x, y))兩個函式，整體電路(forwardCircuit(x,y,z))則以一般函式來撰寫:



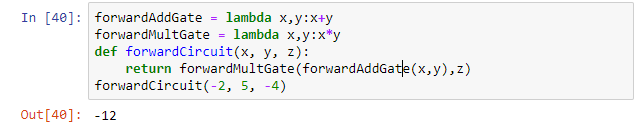
例如:

輸入forwardCircuit(-2, 5, -4)，則回傳-12

輸入forwardCircuit(-4, 3, -5)，則回傳5

|  |
| --- |
| **forwardAddGate = lambda x,y:x+y**  **forwardMultGate = lambda x,y:x\*y**  **def forwardCircuit(x, y, z):**  **return forwardMultGate(forwardAddGate(x,y),z)**  **forwardCircuit(-2, 5, -4)** |

**執行結果擷圖：**



EX 1: 試實作一個函式max\_mul()，當我們給定一個內容皆為整數的串列l時，函式的回傳值是串列中連續兩個相鄰元素乘積的最大值。

例如:

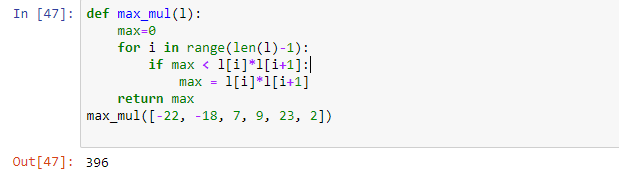
輸入max\_mul([2, -1, 7, 9, 23, 2])， 則輸出207

輸入max\_mul([2, -1, 7, 9, 23, 25])， 則輸出575

輸入max\_mul([-22, -18, 7, 9, 23, 2])， 則輸出396

|  |
| --- |
| **def max\_mul(l):**  **max=0**  **for i in range(len(l)-1):**  **if max < l[i]\*l[i+1]:**  **max = l[i]\*l[i+1]**  **return max**  **max\_mul([-22, -18, 7, 9, 23, 2])** |

**執行結果擷圖：**



EX 2: 試實作一個函式longest\_str()，當我們給定一個內容皆為字串(str)的串列l時，函式的回傳值是串列中長度最長的字串。若長度相同，則回傳index較小的字串。

例如:

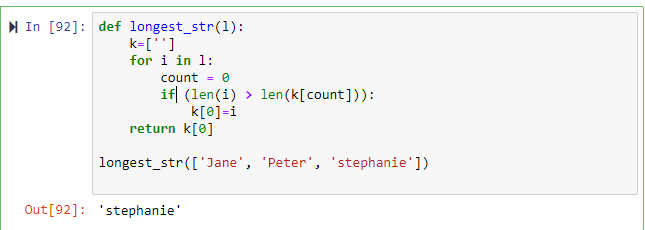
輸入longest\_str(['apple', 'python', 'plasma']),則輸出'python'

輸入longest\_str(['apples', 'python', 'plasma'])，則輸出'apples'

輸入longest\_str(['Jane', 'Peter', 'stephanie'])， 則輸出'stephanie'

|  |
| --- |
| **def longest\_str(l):**  **k=['']**  **for i in l:**  **count = 0**  **if (len(i) > len(k[count])):**  **k[0]=i**  **return k[0]**  **longest\_str(['Jane', 'Peter', 'stephanie'])** |

**執行結果擷圖：**



EX 3: 試實作一個函式score\_threshold()，函式的輸入參數為整數所構成的串列l和門檻值t；當我們給定一個內容皆為整數的串列l,以及一個整數門檻值t時，函式的回傳值是一個判斷原始串列中每個值是否大於等於門檻t的結果(true或是false)。

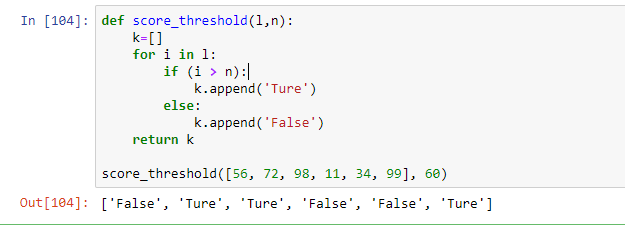
例如:

輸入score\_threshold([56, 72, 98, 11, 34, 99], 60)

輸出[False, True, True, False, False, True]

|  |
| --- |
| **def score\_threshold(l,n):**  **k=[]**  **for i in l:**  **if (i > n):**  **k.append('Ture')**  **else:**  **k.append('False')**  **return k**  **score\_threshold([56, 72, 98, 11, 34, 99], 60)** |

**執行結果擷圖：**



EX 5: 試實作一個函式Insert\_str()，它能在輸入的括號符號中插入指定的字串。

例如:

輸入insert\_str('[[]]', 'Python')，輸出[[Python]]

輸入insert\_str('{{}}', 'C++')，輸出{{C++}}

輸入insert\_str('<<>>', 'Java')，輸出<<Java>>

|  |
| --- |
| **def Insert\_str(str1,str2):**  **ans=len(str1)/2**  **return str1[0:int(ans)]+str2+str1[int(ans):int(len(str1))]**  **Insert\_str('[[]]', 'Python')** |

**執行結果擷圖：**

