**Python程式設計**

**範圍： Set、Dictionary的應用**

**銘傳大學電腦與通訊工程系**

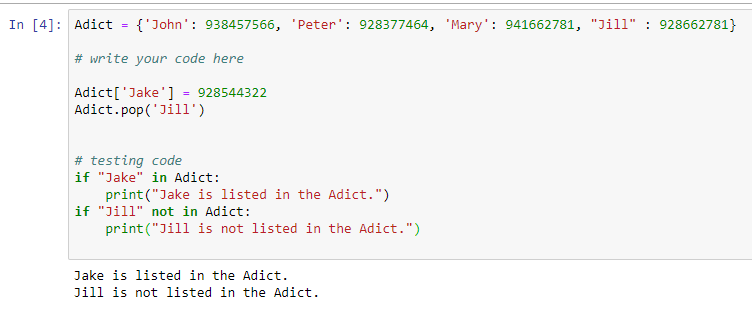
|  |  |
| --- | --- |
| 班 級 | 電通四乙 |
| 姓 名 | 謝曜陽 |
| 學 號 | 05051071 |
| 成 績 | 應繳作業共 8 題，前六題每題10分，後兩題每題20分，滿分為100分  共完成 題，應得 分 |
| 授課教師 | 陳慶逸 |

※直接將你的程式碼貼在指定的欄位裡，並且執行題目要求的輸入參數

※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。

EX 1: 針對下面的字典Adict，試寫幾行python程式碼，增加'Jake'這個名字，且其電話為928544322；並且刪除Jake這筆資料。

|  |
| --- |
| **Adict = {'John': 938457566, 'Peter': 928377464, 'Mary': 941662781, "Jill" : 928662781}**  **# write your code here**  **Adict['Jake'] = 928544322**  **Adict.pop('Jill')**  **# testing code**  **if "Jake" in Adict:**  **print("Jake is listed in the Adict.")**  **if "Jill" not in Adict:**  **print("Jill is not listed in the Adict.")** |



EX 2: 若有一個字典dict\_double的內容如下，試利用程式產生之；並在該字典下面接著寫一個isKeyPresent(x)函式，該函式的功能可檢查所輸入的鍵值x是否存在於Adict之中。

dict\_double的內容如下:

{0: 0, 1: 2, 2: 4, 3: 6, 4: 8, 5: 10, 6: 12, 7: 14, 8: 16, 9: 18, 10: 20, 11: 22, 12: 24, 13: 26, 14: 28, 15: 30, 16: 32, 17: 34, 18: 36, 19: 38}

例如：

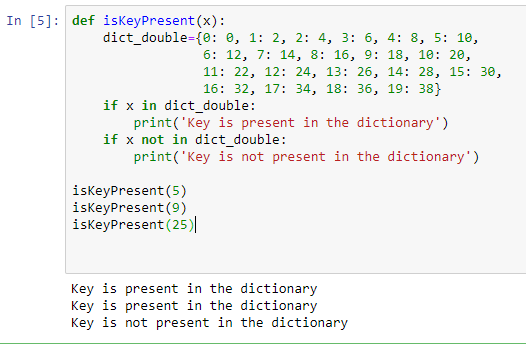
|  |  |
| --- | --- |
| **isKeyPresent(5)** | **Key is present in the dictionary** |
| **isKeyPresent(9)** | **Key is present in the dictionary** |
| **isKeyPresent(25)** | **Key is not present in the dictionary** |

我的作答：

**請在下面欄位貼上程式碼：**

|  |
| --- |
| **def isKeyPresent(x):**  **dict\_double={0: 0, 1: 2, 2: 4, 3: 6, 4: 8, 5: 10,**  **6: 12, 7: 14, 8: 16, 9: 18, 10: 20,**  **11: 22, 12: 24, 13: 26, 14: 28, 15: 30,**  **16: 32, 17: 34, 18: 36, 19: 38}**  **if x in dict\_double:**  **print('Key is present in the dictionary')**  **if x not in dict\_double:**  **print('Key is not present in the dictionary')**    **isKeyPresent(5)**  **isKeyPresent(9)**  **isKeyPresent(25)** |

**執行結果擷圖：**



EX 3: 針對下面的字典Adict，試以下面型式輸出Adict的內容。

Member: John -> Tel: 938457566

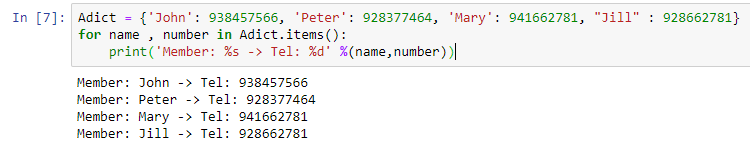
Member: Peter -> Tel: 928377464

Member: Mary -> Tel: 941662781

Member: Jill -> Tel: 928662781

|  |
| --- |
| **Adict = {'John': 938457566, 'Peter': 928377464, 'Mary': 941662781, "Jill" : 928662781}**  **for name , number in Adict.items():**  **print('Member: %s -> Tel: %d' %(name,number))** |

**執行結果擷圖：**

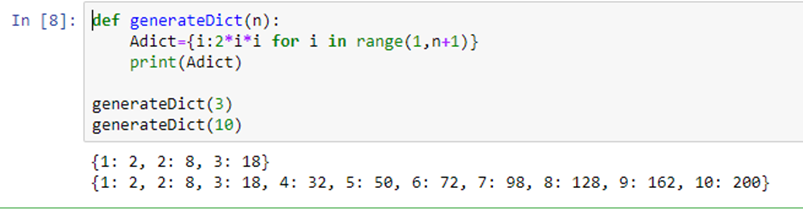


EX 4: 試寫一python函式generateDict(n)，它會根據我們所給予的n值，回傳一個內容為{1:2, 2:8, 3:18, 4: 32, …, n: 2\*n\*n}的字典。例如：

|  |  |
| --- | --- |
| generateDict(3) | {1: 2, 2: 8, 3: 18} |
| generateDict(10) | {1: 2, 2: 8, 3: 18, 4: 32, 5: 50, 6: 72, 7: 98, 8: 128, 9: 162, 10: 200} |

|  |
| --- |
| **def generateDict(n):**  **Adict={i:2\*i\*i for i in range(1,n+1)}**  **print(Adict)**    **generateDict(3)**  **generateDict(10)** |

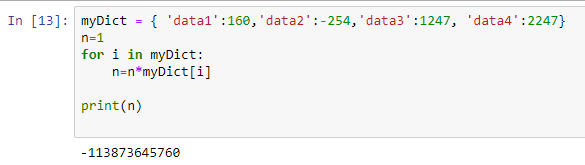
**執行結果擷圖：**



EX 5: 若myDict = { 'data1':160,'data2':-254,'data3':1247, 'data4':2247}，試寫一python程式來得到字典裡所有值(value)連乘的結果。以本例而言，其輸出值應為 -1333800。

|  |
| --- |
| **myDict = { 'data1':160,'data2':-254,'data3':1247, 'data4':2247}**  **n=1**  **for i in myDict:**  **n=n\*myDict[i]**  **print(n)** |

**執行結果擷圖：**



EX 6: zip() 是 Python 的一個內建函數，它接受一系列可迭代的對象作為參數，將對象中對應的元素打包成一個個 tuple。例如：

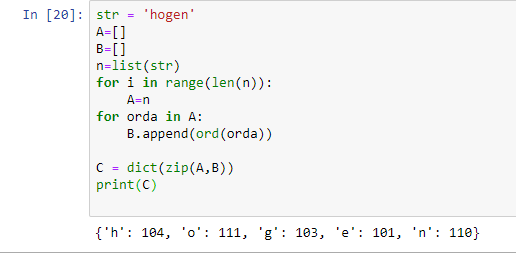
|  |
| --- |
| **a = [1, 2, 3]**  **b = ['bird', 'dog', 'cat']**  **c = dict(zip(a,b)) # c = {1: 'apple', 2: 'dog', 3: 'cat'}** |

現有字串str = 'hogen', 試將之轉成一個由該字串裡的字母所構成的串列A；再建立一個由這些字母對應的ASCII 碼所構成的串列B (使用ord()函式可將英文字母轉換為ASCII碼)，最後列印出由A和B建立的字典C。C的內容如下:

C = {'h': 104, 'o': 111, 'g': 103, 'e': 101, 'n': 110}

|  |
| --- |
| **str = 'hogen'**  **A=[]**  **B=[]**  **n=list(str)**  **for i in range(len(n)):**  **A=n**  **for orda in A:**  **B.append(ord(orda))**  **C = dict(zip(A,B))**  **print(C)** |

**執行結果擷圖：**



EX 7: 現在一字典num = {'n1': [2, 3, 7, 9, 1], 'n2': [5, 1, 2, 8, 13], 'n3': [3, 2, 4, 6, 9]}，試寫一python程式來將字典裡所有由串列所構成的值(value)都進行排序。例如本例最後應輸出：

{'n1': [1, 2, 3, 7, 9], 'n2': [1, 2, 5, 8, 13], 'n3': [2, 3, 4, 6, 9]}

以for-loop寫法：

|  |
| --- |
| **num = {'n1': [2, 3, 7, 9, 1], 'n2': [5, 1, 2, 8, 13], 'n3': [3, 2, 4, 6, 9]}** |

以解析式for-loop寫法：

|  |
| --- |
| **num = {'n1': [2, 3, 7, 9, 1], 'n2': [5, 1, 2, 8, 13], 'n3': [3, 2, 4, 6, 9]}** |

**執行結果擷圖：**

EX 8: 找出同時出現在兩個不同的字典中的鍵-值對，

例如：

x = {'key1': 1, 'key2': 3, 'key3': 2}

y = {'key1': 1, 'key2': 2}

Sample output: ('key1', 1)

x = {'key1': 1, 'key2': 3, 'key3': 2}

y = {'key1': 1, 'key2': 3}

Sample output:

('key2', 3)

('key1', 1)

|  |
| --- |
| **x = {'key1': 1, 'key2': 3, 'key3': 2}**  **y = {'key1': 1, 'key2': 3}** |

**執行結果擷圖：**