

python 程式設計

第 八 講

集合、字典

集合：set (一)

- 集合：內存不同元素，且元素無特定次序

- 集合設定：去除相同元素

```
>>> a = { 3, 2, 3, 5 }           # 直接使用大括號定義
>>> a
{2, 3, 5}
```

```
>>> b = set( [ 3, 2, 3, 5 ] )    # 由串列轉型
>>> b
{2, 3, 5}
```

```
>>> c = set( range(5) )          # 由 range 轉型
>>> c
{0, 1, 2, 3, 4}
```

```
>>> d = set( "abbbccde" )        # 分解字串成字元
>>> d
{'e', 'c', 'd', 'a', 'b'}
```

集合：set (二)

➤ 空集合

```
>>> a = set()                # a 是空集合

>>> b = set( [1,2] )
>>> b.clear()                # 使用 clear() 後成為空集合
```

集合：set (三)

■ 集合複製

- 複製元素：兩集合的元素值一樣，各佔不同空間

```
>>> a = set( range(1,4) )      # a = {1, 2, 3}
>>> b = a.copy()              # b 集合元素複製自 a ， 各佔不同空間
```

- 複製名稱：原集合多個名稱

```
>>> a = set( range(1,4) )      # a = {1, 2, 3}
>>> b = a                      # b 與 a 是同個集合，佔用相同空間
```

集合：set (四)

■ 增減元素與元素數量

➤ 集合元素增減

集合操作	作用
<code>foo.add(c)</code>	將 <code>c</code> 元素加入 <code>foo</code> 集合
<code>foo.remove(c)</code>	將 <code>c</code> 由 <code>foo</code> 集合移除。若 <code>c</code> 不存在，有錯誤訊息
<code>foo.discard(c)</code>	將 <code>c</code> 由 <code>foo</code> 集合移除。 若 <code>c</code> 不存在，沒有錯誤訊息，視為無效動作
<code>foo.clear()</code>	清除 <code>foo</code> 集合的所有元素， <code>foo</code> 成為空集合
<code>foo.pop()</code>	隨意取出一個元素，若 <code>foo</code> 無元素，會有錯誤訊息
<code>len(foo)</code>	<code>foo</code> 集合元素個數

❖ 集合內元素位置是「無順序」的，使用 `pop()` 取出元素即變得「隨意」

集合：set (五)

```
>>> a = {2, 9, 8}
```

```
>>> a.add(3)
```

```
>>> a.remove(9)
```

```
>>> a.remove(5)
```

```
>>> a.discard(8)
```

```
>>> a.discard(7)
```

```
>>> a.pop()
```

```
2
```

```
>>> len(a)
```

```
1
```

```
>>> a.clear()
```

```
>>> a
```

```
set()
```

```
# a = {8, 9, 2}
```

```
# a = {8, 9, 2, 3}
```

```
# a = {8, 2, 3}
```

```
# 5 不在 a 中，有錯誤訊息
```

```
# a = {2, 3}
```

```
# 7 不在 a 中，無效動作
```

```
# 隨意取出一個元素
```

```
# a 僅剩一個元素
```

```
# a 變成空集合
```

集合：set (六)

■ 取得集合元素

- 使用 `for` 迴圈可逐一取得元素

```
a = set( [2,9,8,7,1,2,8] )  
for x in a : print( x , end=" ")  
  
print( "\n" + "a =" , a )
```

輸出為：

```
8 9 2 1 7  
a = {8, 9, 2, 1, 7}
```

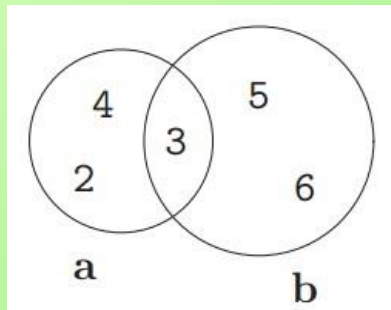
- 利用迴圈與 `in` 配合取得集合元素時不可增減集合元素

```
a = set( [2,8,7] )  
for x in a :  
    a.discard(x)                # 錯誤，迴圈刪除元素  
    a.add(x+10)                 # 錯誤，迴圈新增元素
```

集合：set (七)

■ 聯集、交集、差集、對稱差集

➤ 集合間的運算



`a = { 2, 3, 4 }` `b = { 5, 3, 6 }`

集合運算	函式用法	運算子用法與結果
聯集	<code>a.union(b)</code>	<code>a b = { 2, 3, 4, 5, 6 }</code>
交集	<code>a.intersection(b)</code>	<code>a & b = { 3 }</code>
差集	<code>a.difference(b)</code>	<code>a - b = { 2, 4 }</code>
差集	<code>b.difference(a)</code>	<code>b - a = { 5, 6 }</code>
對稱差集	<code>a.symmetric_difference(b)</code>	<code>a ^ b = { 2, 4, 5, 6 }</code>

集合：set (八)

➤ 集合操作更新

集合操作更新	函式用法	運算子用法 \Leftrightarrow 相等用法
聯集更新	<code>a.update(b)</code>	<code>a = b</code> \Leftrightarrow <code>a = a b</code>
交集更新	<code>a.intersection_update(b)</code>	<code>a &= b</code> \Leftrightarrow <code>a = a & b</code>
差集更新	<code>a.difference_update(b)</code>	<code>a -= b</code> \Leftrightarrow <code>a = a - b</code>
對稱差集更新	<code>a.symmetric_difference_update(b)</code>	<code>a ^= b</code> \Leftrightarrow <code>a = a ^ b</code>

<code>a = { 2, 3, 4 }</code> <code>b = { 5, 3, 6 }</code>		
聯集更新	<code>a = b</code>	<code>a = { 2, 3, 4, 5, 6 }</code>
交集更新	<code>a &= b</code>	<code>a = { 3 }</code>
差集更新	<code>a -= b</code>	<code>a = { 2, 4 }</code>
差集更新	<code>b -= a</code>	<code>b = { 5, 6 }</code>
對稱差集更新	<code>a ^= b</code>	<code>a = { 2, 4, 5, 6 }</code>

集合：set (九)

■ 元素是否在集合內：in 與 not in

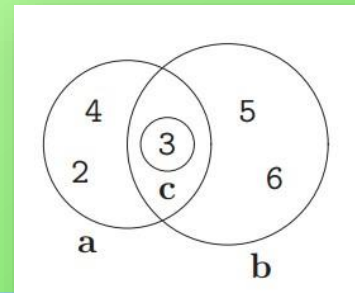
- `a in foo`: 檢查 `a` 元素是否在 `foo` 集合內
- `a not in foo`: 檢查 `a` 元素是否不在 `foo` 集合內

```
>>> a = set( [1,3,5,7] )  
>>> 5 in a  
True  
  
>>> 3 not in a  
False
```

集合：set (十)

■ 兩集合間的關係

➤ 等於、不等於、包含、包含於



`a = { 2, 3, 4 }` `b = { 5, 3, 6 }` `c = { 3 }`

<code>a 等於 b</code>	<code>a == b</code>	<code>==> False</code>
---------------------	---------------------	---------------------------

<code>a 不等於 b</code>	<code>a != b</code>	<code>==> True</code>
----------------------	---------------------	--------------------------

<code>a 包含 c</code>	<code>a >= c</code>	<code>==> True</code>
---------------------	------------------------	--------------------------

<code>a 包含但不等於 c</code>	<code>a > c</code>	<code>==> True</code>
-------------------------	-----------------------	--------------------------

<code>c 包含於 b</code>	<code>c <= b</code>	<code>==> True</code>
----------------------	------------------------	--------------------------

<code>c 包含於 b 但不等於 b</code>	<code>c < b</code>	<code>==> False</code>
-----------------------------	-----------------------	---------------------------

<code>a 與 b 無交集</code>	<code>a.isdisjoint(b)</code>	<code>==> False</code>
------------------------	------------------------------	---------------------------

➤ `a.issubset(b)`：等同 `a <= b`

➤ `a.issuperset(b)`：等同 `a >= b`

集合：set (十一)

■ 凍集合：frozenset

- 凍集合：為建構後即不能變更的集合
- 建構方式：如同集合

```
>>> a = frozenset([1,2])
```

```
>>> b = frozenset(range(10))
```

26 個字母

```
>>> c = frozenset( [ chr(ord('a')+i) for i in range(26) ] )
```

➤ 凍集合與集合互轉

```
>>> a = set( [1,3,5] )
```

```
>>> b = frozenset(a)
```

轉型為凍集合

```
>>> c = frozenset( [2,4,6] )
```

```
>>> d = set(c)
```

轉型為集合

集合：set (十二)

■ 簡單集合範例：

➤ 找出兩位數的數字各不重複

```
for n in range(10,100) :  
    s = str(n)  
    if len(s) == len(set(s)) : print( n , end=" ")
```

輸出：

10 12 13 14 15 16 17 18 19 20 21 23 24 25 26 ... 96 97 98

➤ 找出不同字的字數

```
p = ( "渺渺茫茫墨潑天，飄飄拂拂雨如煙。  
      蒼蒼翠翠山遮寺，白白紅紅花滿川。"  
      "整整齊齊沙上雁，來來往往渡頭傳。  
      行行坐坐看無盡，世世生生作語傳。" )  
  
wc = set( [ c for c in p if c not in "。 , " ] )  
print(len(wc))
```

輸出：

39

集合：set (十三)

■ 簡單集合範例：

➤ 產生六個介於 [1,49] 樂透號碼

```
import random
lottery = set()

while True :
    lottery.add( random.randint(1,49) )
    if len(lottery) == 6 : break

print( " ".join( map(str,sorted(lottery)) ) )
```

輸出：

6 9 12 15 28 32

❖ 以上也可使用 random 套件的打亂數列函式 shuffle

集合：set (十四)

- 產生六個介於 [1,49] 樂透號碼

使用方式如下：

```
import random
```

```
# nums 為 1, 2, 3, ..., 49
```

```
nums = list( range(1,50) )
```

```
# 打亂 nums 串列
```

```
random.shuffle( nums )
```

```
# 取前六個數字，由小到大輸出
```

```
print( " ".join( map(str,sorted(nums[:6])) ) ) )
```

字典：dictionary (一)

- 字典：可使用各種 `immutable` 型別為索引的資料型別

- 字典型別

```
>>> nums = {}                                # 設定 a 為空字典
>>> nums["one"] = 1                          # 索引為字串
>>> nums["two"] = 2
>>> nums["三"] = 3
>>> nums[4] = "four"                         # 索引為整數
>>> nums[0.1] = "point one"                  # 索引為浮點數
>>> nums
{0.1: 'point one', '三': 3, 'two': 2, 4: 'four', 'one': 1}
```

- 字典內的每個索引 (**key**) 與其對映值 (**value**) 為一個資料組 (**pair**)
- 字典的索引稱為 **key**，索引可為任何 `immutable` 型別，
例如：整數、浮點數、字串、常串列等等

字典 (二)

■ 字典設定

➤ dict 設定：使用字串當索引

```
>>> dict( one=1 , two=2, three=3 )  
{ 'two': 2, 'three': 3, 'one': 1 }
```

➤ 使用串列儲存各組資料

```
>>> dict( [ ('one',1), ('two',2), ('three',3) ] )  
{ 'two': 2, 'one': 1, 'three': 3 }
```

➤ 直接使用大括號

```
>>> { 'one':1, 'two':2, 'three':3 }
```

字典 (三)

■ 字典設定

- 使用 `zip` 拉鏈函式將分離的索引與對映值串列合成

```
>>> dict( zip( ["one","two","three"] , [1,2,3] ) )  
{ 'two': 2, 'one': 1, 'three': 3 }
```

- `dict comprehension`

```
>>> { x : x*x for x in range(1,5) }  
{1: 1, 2: 4, 3: 9, 4: 16}
```

- 空字典

```
>>> foo = {}
```

字典 (四)

■ 字典複製

- 複製各組資料：兩字典資料一樣，各佔不同空間

```
>>> a = { x:x*2 for x in range(1,5) }  
>>> b = a.copy()                # b 字典複製自 a ， 各佔不同空間
```

- 複製名稱：原字典多個名稱

```
>>> a = { x:x*2 for x in range(1,5) }  
>>> b = a                        # b 與 a 是同個字典，佔用相同空間
```

字典 (五)

■ 字典資料增減與資料量

➤ 增加一組資料: `foo[key]`

```
>>> foo = { 'one':1, 'two':2 }  
>>> foo['three'] = 3  
>>> foo  
{ 'two': 2, 'one': 1, 'three': 3}
```

➤ 增加一組以上資料: `foo.update()`

```
>>> foo = { 'one':1, 'two':2 }  
>>> foo.update( [ ('three',3) , ('four',4) ] )  
  
>>> foo  
{ 'two': 2, 'four': 4, 'one': 1, 'three': 3}
```

字典 (六)

■ 字典資料增減與資料量

➤ 讀入 b 字典的資料: `foo.update(b)`

```
>>> foo = { 'one':1, 'two':2 }
>>> bar = { 'three':3, 'four':4 }
>>> foo.update( bar )

>>> foo
{'two': 2, 'four': 4, 'one': 1, 'three': 3}
```

➤ 刪除一組資料: `del foo[key]`

```
>>> foo = { 'one':1, 'two':2 }
>>> del foo['two']
>>> foo
{'one': 1}
```

字典 (七)

■ 字典資料增減與資料量

- 清除所有資料，變成空字典：`foo.clear()`

```
>>> foo = { 'one':1, 'two':2 }  
>>> foo.clear()
```

- 取得資料組數：`len`

```
>>> foo = { 'one':1, 'two':2 }  
>>> len(foo)  
2
```

字典 (八)

■ 取出字典資料

- `foo.get(k,b)` : 取出索引為 `k` 的對映值, 若無此索引, 則以 `b` 回傳

```
>>> foo = { 'one':1, 'two':2, 'three':3 }
>>> foo.get('two')
2
>>> foo.get('four')                                # 無此 key , 無回傳
>>> foo.get('four',4)                                # 無此 key , 但回傳 4
4
```

- 取出索引、對映值、與成對資料

操作函式	取出的資料
<code>foo.keys()</code>	取出 <code>foo</code> 字典所有的索引
<code>foo.values()</code>	取出 <code>foo</code> 字典所有的對映值
<code>foo.items()</code>	取出 <code>foo</code> 字典內所有成對資料

字典 (九)

```
foo = { 'one':1, 'two':2, 'three':3 }
```

取出所有的索引

```
for k in foo.keys() :  
    print( k , '-->' , foo[k] )
```

分解索引與對映值成串列

```
for k , v in foo.items() :  
    print( k , '-->' , v )
```

p 為由 (索引,對映值) 組成的串列

```
for p in foo.items() :  
    print( p[0] , '-->' , p[1] )
```

以上三個迴圈都輸出：

```
two --> 2  
one --> 1  
three --> 3
```


字典 (十)

- 可使用 `in` 取出索引

```
foo = { 'one':1, 'two':2, 'three':3 }  
for k in foo :  
    print( k , '-->' , foo[k] )
```

- 以迴圈取得字典資料時，不能在迴圈內增減索引資料：

```
foo = { 'one':1, 'two':2, 'three':3 }  
for k in foo :  
    del foo[k]                # 錯誤，變更字典索引  
    foo[k.upper()] = foo[k]   # 錯誤，變更字典索引  
  
for k in foo.keys() :  
    foo[k.upper()] = foo[k]   # 錯誤，變更字典索引  
  
for k , v in foo.items() :  
    foo[k.upper()] = v        # 錯誤，變更字典索引
```

字典 (十一)

- 使用 `list` 可將 `keys()`, `values()`, `items()` 轉為串列

```
>>> foo = { 'one':1, 'two':2, 'three':3 }
>>> list(foo.keys())
['two', 'one', 'three']

>>> list(foo.values())
[2, 1, 3]

>>> list(foo.items())
[('two', 2), ('one', 1), ('three', 3)]
```

字典 (十二)

■ 檢查索引是否存在: `in` 與 `not in`

➤ `k in foo`: 檢查 `k` 是否為 `foo` 字典的索引

➤ `k not in foo`: 檢查 `k` 是否不為 `foo` 字典的索引

```
>>> foo = dict( one=1 , two=2, three=3 )
>>> 'one' in foo
True

>>> 'four' not in foo
True

>>> 3 in foo
False
```

簡單字典範例

- 讀入一句英文，計算母音的個數

```
vowelc = {}

for c in "To be, or not to be: that is the question." :
    c = c.lower()

    if c in 'aeiou' :
        vowelc[c] = 1 + ( vowelc[c] if c in vowelc else 0 )

# 依字母順序列印
for k in sorted( vowelc.keys() ) :
    print( k , ':' , vowelc[k] , sep="" , end="  " )

print()

# 依字母出現多寡列印
for k , v in sorted( vowelc.items() , key = lambda x : -x[1] ) :
    print( k , ':' , v , sep="" , end="  " )

print()
```

輸出：

```
a:1 e:4 i:2 o:5 u:1
o:5 e:4 i:2 u:1 a:1
```

尋找詩句交集字

讀入以下四首詩，不考慮標點符號，找出四首詩都有的字。

去年今日此門中，人面桃花相映紅。人面不知何處去，桃花依舊笑春風。
春城無處不飛花，寒食東風御柳斜。日暮漢宮傳蠟燭，輕煙散入五侯家。
雲想衣裳花想容，春風拂檻露華濃。若非群玉山頭見，會向瑤臺月下逢。
折戟沉沙鐵未銷，自將磨洗認前朝。東風不與周郎便，銅雀春深鎖二喬。

```
with open("poems.txt") as infile :  
    for n , line in enumerate(infile) :  
        # 去除標點符號  
        poem = set( filter( lambda c : c not in " 。 , " , line.strip() ) )  
        if n > 0 :  
            words.intersection_update(poem)  
        else :  
            words = poem.copy()  
print( " ".join( words ) )
```

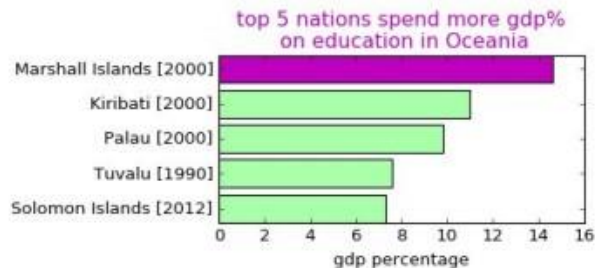
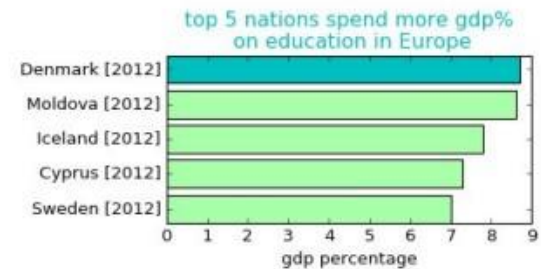
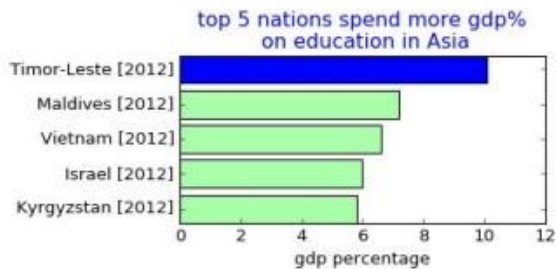
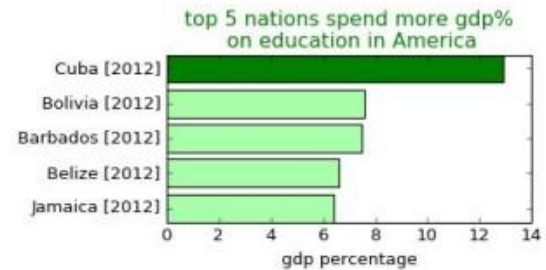
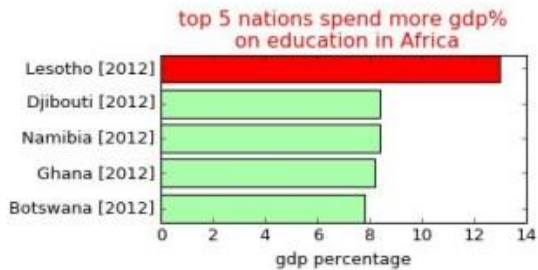
輸出：

風 春

各洲國家在教育支出佔 gdp 百分率的前五名 (一)

- 此題與上一章世界各國在教育支出佔該國 gdp 百分率範例類似，但資料檔的最後一個欄位改為國家所在的洲，如美洲、亞洲、歐洲等等。程式要在各洲內找出國家在教育支出佔 gdp 百分率最高的前五名，用橫條圖畫出。程式的作法是將洲名與其內的國家資料存為字典組，索引為洲名，國名與其 gdp 資料存成常串列。之後再利用排序程式，先比 gdp 再比國名方式重新排列，最後取出前五名印成橫條圖。在列印橫條圖時，程式刻意的將第一名國家與其餘四名分開，以不同顏色顯示，藉以突顯第一名的國家。

各洲國家在教育支出佔 gdp 百分率的前五名 (二)



各洲國家在教育支出佔 gdp 百分率的前五名 (三)

```
import matplotlib.pyplot as plt

ctn = {}
with open("edu2.dat") as infile :

    for line in infile :

        if line.isspace() : continue

        # 分離資料再重新組合 ( 國名+年 , gdp ) 常串列
        *nation , gdp , year , continent = line.split()
        if gdp == 'n.a.' : gdp = "-1"
        data = ( " ".join(nation) + " [" + year + "]" , float(gdp) )

        # 跨兩洲的國家以斜線分割：如俄羅斯
        for c in continent.split('/') :
            if c in ctn :
                ctn[c].append( data )
            else :
                ctn[c] = [ data ]
```


各洲國家在教育支出佔 gdp 百分率的前五名 (四)

```
no = 5
ys = [ y for y in range(no,0,-1) ]

colors = 'rgbcmyk'
plt.figure(facecolor='white')

for i , k in enumerate(sorted(ctn.keys())) :

    # 將圖形分割為 3x2 子圖分佈
    fig = plt.subplot(3,2,i+1)

    # 各洲國家排列：先比 gdp 再比國名
    ctn[k].sort( key=lambda p :
                 ( -p[1] , p[0] ) )

    vals , nations = [] , []
    for j , p in enumerate(ctn[k]) :
        vals.append(p[1])
        nations.append(p[0])
        if j+1 == no : break

    # 設定第一名國家顏色與其餘四國顏色不同
    fig.barh(ys[0],vals[0],align='center' ,
             color=colors[i])
    fig.barh(ys[1:],vals[1:],align='center' ,
             color='#aaffaa')
```

```
# 設定子圖文字說明
fig.set_title("top " + str(no) +
              " nations spend
              more gdp%\n on
              education in "
              + k ,
              color=colors[i])

# x 軸文字
fig.set_xlabel("gdp percentage")

# y 軸刻度位置
fig.set_yticks(ys)

# y 軸刻度文字
fig.set_yticklabels(nations,
                    color='black')

fig.plot()

plt.tight_layout()
plt.show()
```

列印課表 (一)

- 撰寫程式讀入左側的課程上課時間檔案，印出對應的右側課程表：

微積分	四:78	五:56
物理	三:12	二:7
化學	三:8	一:34
經濟	二:56	
英文	五:34	
體育	四:34	
國文	三:56	

---->

	一	二	三	四	五
1			物		
2			物		
3		化		體	英
4		化		體	英
5		經	國		微
6		經	國		微
7		物		微	
8			化	微	

列印課表 (二)

```
Day , Section = 5 , 8
cnum = '一二三四五'
c2n = dict( [ ( b , a ) for a , b in enumerate(cnum) ] )
wkclass = [ [None]*Day for i in range(Section) ]

with open("schedule.dat") as infile :
    for line in infile :
        course , *csect = line.split()
        for p in csect :
            a , b = p.split(':')
            w = c2n[a]                # 星期幾
            for c in b :
                s = int(c)-1
                wkclass[s][w] = course[0]

print( "    | " + " ".join( cnum ) )
print( "-"*(5+3*5) )

for s in range(Section) :
    if s == 4 : print( "-"*(5+3*5) )
    print( " "+str(s+1)+" | " , end="" )
    for w in range(Day) :
        print( wkclass[s][w] if wkclass[s][w] else " " , end=" " )
    print()
```

中文筆畫數 (一)

- 在萬國碼 (unicode) 中，中文的範圍是介於 [U+4E00, U+9FA5]，這是以十六進位表示的編號順序，若以十進位表示則是介於 [19968, 40869]。今有一個中文筆畫檔，其中的每一行包含中文字在萬國碼的編號順序與其筆畫數，例如：

....			U+752A	6
U+7521	10		U+752B	7
U+7522	11		U+752C	7
U+7523	11		U+752D	9
U+7524	12		U+752E	9
U+7525	12		U+752F	12
U+7526	12		U+7530	5
U+7527	14		U+7531	5
U+7528	5		U+7532	5
U+7529	5		

中文筆畫數 (二)

- 以上資料的末三個十六進位數所對應的中文字分別為「田」「由」「甲」，都是 5 畫。現要利用這個筆畫檔，列印出所輸入的中文字串的各字筆畫數與全部的筆畫。例如：

> 中央大學

中:4 央:5 大:3 學:16 總筆畫數:28

> 山明水秀

山:3 明:8 水:4 秀:7 總筆畫數:22

中文筆畫數 (三)

```
sdict = {}

# 讀入筆畫檔存入 sdict 字典
with open( "strokes.txt" ) as infile :

    for line in infile :
        ucode , strokes = line.split()
        num = int(unicode[2:],16)
        sdict[num] = int(strokes)

# 輸入中文句子
while True :

    name = input("> ")

    totals = 0
    for c in name :
        strokes = sdict[ord(c)]
        print( c , ":" , strokes , sep="" , end="  " )
        totals += strokes

    print( " 總筆畫數:" , totals , sep="" , end="\n\n" )
```

萬國碼中文字依筆畫排序 (一)

- 利用上例的筆畫檔，撰寫程式將萬國碼中的所有中文字依筆畫排序如下：

— | \ / 1 1 2 1 1

丁 𠂔 七 上 下 𠂔 𠂔 𠂔 𠂔 𠂔 乃 𠂔 𠂔 九 了 二 十 一 人 亻 儿 入
八 冫 彳 冫 几 口 刀 刁 乚 力 勹 匕 匸 亡 卩 卜 尸 厶 又

也加子升
乞刃子广
之刃女么马
可九矢干乍
义九大中飞
么凡夕巳门
毛凶文已己
久兀文已己
久亿士工乡
丸凶土川乡
斗今口凶乡
丫△口山才
个亡及中才
与亏叉尸个
刀亏卫尤个
下于月小乡
上于什寸乡
三个千△乡
丈乡刊子弓
万习勺子弋

予介内匆公
书今毋勾丛
乏仇目勿玄
乏伙节勾历
纣仇兮勾厅
毗仆六勾厄
乏仅公勾户
乌仄内办印
为行元劝卞
丹仿先刈卞
丰仁允切卞
丰什以分卞
丰亦仓办卞
中亢令凶升
专三仓凤杂
刃井今亢卞
丑五仙亢区
丐元从冈匹
丐互仍有化
不云欠冈勾

• • • • •

萬國碼中文字依筆畫排序 (二)

```
sdict = {}

# 讀筆畫資料檔，將同筆畫的字存在一起
with open( "strokes.txt" ) as infile :

    for line in infile :
        ucode , strokes = line.split()
        char = chr(int(unicode[2:],16))
        stroke = int(strokes)

        # 若筆畫數第一次出現，以字串儲存
        if stroke not in sdict :
            sdict[stroke] = char
        else :
            sdict[stroke] += char

# 依筆畫由小到大排序
for stroke in sorted( sdict.keys() ) :
    chars = sdict[stroke]
    print( stroke , "畫 :" , len(chars) , "個" )
    a = 0
    while a < len(chars) :
        print( " ".join(chars[a:a+20]) )
        a += 20

    print()
```