

自行車竊案分析

字典資料結構

前情提要 (list)

- ▶ 列表是一種資料結構，可以儲存一個以上不同型態的資料。

```
data = [1, '師大', 'NTNU']
```

```
data += [3.14]
```

```
data.append([2019, 1, 1])
```

```
for e in data:  
    print(e, type(e))
```

```
1 <class 'int'>  
師大 <class 'str'>  
NTNU <class 'str'>  
3.14 <class 'float'>  
[2019, 1, 1] <class 'list'>
```

前情提要 (list)

▶ 列表特色

- 將資料以指定順序儲存。
- 以整數編號（索引值）取用資料。

```
data = [9, 7, 4, 5]

data += [6]          #在最後插入 6 ;          [9, 7, 4, 5, 6]
data.insert(1, 22)   #在位置 1 之前插入 22 ; [9, 22, 7, 4, 5, 6]
print('data[1] = ', data[1])

data.insert(3, 88)   #在位置 3 之前插入 88 ; [9, 22, 7, 88, 4, 5, 6]
print('data[3] = ', data[3])

print(data)
```

```
data[1] = 22
data[3] = 88
[9, 22, 7, 88, 4, 5, 6]
```

前情提要 (list)

▶ 列表可以建立一種 0, 1, 2, ... 和資料的關係。

```
scores = [0]*100

n = int(input('全班人數>'))
for i in range(n):
    scores[i+1] = int(input('學生'+str(i+1)+'分數>'))

stu_id = int(input('你想知道幾號學生的分數>'))
if 1<=stu_id<=n:
    print(scores[stu_id], '分')
else:
    print('查無此人')
```

全班人數>5
學生1分數>100
學生2分數>90
學生3分數>80
學生4分數>95
學生5分數>88
你想知道幾號學生的分數>4
95 分

	[0]	[1]	[2]	[3]	[4]	[5]	[6]		[99]
scores	0	100	90	80	95	88	0	...	0

前情提要 (list)

▶ 列表難處理的對應關係

- 以整數作為查詢鍵 (key)，但數值不連續又很大

- 例：ISBN 查詢書名

9789863123798 → 'Python 程式設計超入門'

9789864342549 → 'Python 程式設計實例入門'

```
books = ['']*(10**7) #定一個超級大的列表
books[3123798] = 'Python 程式設計超入門'
books[4342549] = 'Python 程式設計實例入門'

isbn = int(input('輸入 ISBN >')) - 9789860000000

if books[isbn] != '': #或只寫 if books[isbn]:
    print(books[isbn])
else:
    print('查無此書')
```

輸入 ISBN >9789863123798
Python 程式設計超入門

- 此例雖然可行，但會浪費很多空間。

此例我們只考慮 ISBN 是 978986 開頭的書。

前情提要 (list)

▶ 列表難處理的對應關係

■ 非整數作為查詢鍵 (key)

- 例：書名反查 ISBN

'Python 程式設計超入門' → 9789863123798

'Python程式設計實例入門' → 9789864342549

方法 A

```
def GetBook(books, name):  
    for e in books:  
        if e[0] == name:  
            return e  
  
books = [['Python 程式設計超入門', 9789863123798],  
         ['Python程式設計實例入門', 9789864342549]]  
  
name = input('輸入書名 >')  
b = GetBook(books, name)  
if b != None: #或只寫 if b:  
    print(b[1])  
else:  
    print('查無此書')
```

輸入書名 >Python程式設計實例入門
9789864342549

前情提要 (list)

▶ 列表難處理的對應關係

■ 非整數作為查詢鍵 (key)

- 例：書名反查 ISBN

'Python 程式設計超入門' → 9789863123798

'Python程式設計實例入門' → 9789864342549

```
方法 B def GetBook(books, name):  
    if name in books[0]:  
        ind = books[0].index(name)  
        return name, books[1][ind]  
  
books = [['Python 程式設計超入門', 'Python程式設計實例入門'],  
         [9789863123798, 9789864342549]]  
  
name = input('輸入書名 >')  
b = GetBook(books, name)  
if b != None: #或只寫 if b:  
    print(b[1])  
else:  
    print('查無此書')
```

輸入書名 >Python程式設計實例入門
9789864342549

字典 (dict) 資料結構

▶ 字典資料結構可以輕鬆記錄鍵 (key) 與值 (value) 的關係。

- 在列表中，鍵是非負整數。
- 在字典中，鍵可以是數字也可以是文字。

```
rec = {}                                #rec 是一個字典；或可寫為 rec = dict()

rec['milk'] = 100                        #字串 'milk' 對應到整數 100
rec[123] = 4                             #整數 123 對應到整數 4
rec['臺灣'] = 'Taiwan'                  #字串 '臺灣' 對應到字串 'Taiwan'

print(rec['milk'])                       #經由 [] 運算，以 'milk' 查詢
print(1+rec[123])

place = '臺灣'
print(rec[place])                       #[] 裡面可以是變數（如同操作列表一樣）

print(rec[3])
print(rec['台灣'])
```

rec	
'milk'	100
123	4
臺灣	'Taiwan'

```
100
5
Taiwan
KeyError: 3
```


字典 (dict) 資料結構

▶ 來一個「字典」的範例

- 不只可以查，也可以更改 value。
- 每個鍵只能出現一次，對應一個值。
- 和列表相似，我們可以用 `in` 運算來確認「某個鍵」是否存在。

```
mydict = {}

mydict['book'] = '書'
mydict['computer'] = '電腦'
mydict['program'] = '程式'

query = input('英翻中>')

mydict['computer'] = '計算機'

if query in mydict:
    print('解釋:', mydict[query])
else:
    print('未收錄此字')
```

mydict

'book'	'書'
'computer'	'電腦'
'program'	'程式'

mydict

'book'	'書'
'computer'	'計算機'
'program'	'程式'

英翻中>book
解釋: 書

英翻中>csie
未收錄此字

字典 (dict) 資料結構

▶ 初始化字典的語法

- 除了用 `[]` 逐一將 鍵-值對 (key-value pair) 加入字典，我們也可以一次加入多筆。

方法 A

```
eng = 'apple'  
chinese = '蘋果'
```

```
mydict = {'book': '書', 'computer': '電腦', 'program': '程式', eng: chinese}  
print(mydict['book'])
```

也可以是變數

方法 B

```
eng = 'apple'  
chinese = '蘋果'
```

```
mydict = dict([('book', '書'), ('computer', '電腦'), ('program', '程式'), (eng, chinese)])  
print(mydict['program'])
```

方法 C

```
books = [['Python 程式設計超入門', 9789863123798],  
         ['Python 程式設計實例入門', 9789864342549]]
```

```
mydict = dict(books) # 可以直接以 n×2 的二維列表來初始化  
print(mydict['Python 程式設計超入門'])
```

字典 (dict) 資料結構

▶ 撰寫程式來填充字典

- 當要填充的內容較多（例如從檔案中讀取大批資料），可利用迴圈來填充字典。

```
chinese = '一二三四五六七八九'
digit = '123456789'

mydict = {}
for i in range(len(chinese)):
    mydict[ chinese[i] ] = digit[i]

query = input('輸入由一、二、..... 九的中文字>')
for c in query:
    print(mydict[c], end='')
```

mydict

'一'	1
'二'	2
'三'	3

⋮

輸入由一、二、..... 九的中文字>九九八一
9981



有個 `zip()` 函式，有興趣的同學可以試試它有什麼妙用。

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題2 Dictionary of synonyms。

- 給定 n 對同義詞後，查某個詞的同義詞。
 - 範例中，'Hello' 與 'Hi' 為同義詞。

範例輸入	範例輸出
3 Hello Hi Bye Goodbye List Array Goodbye	Bye

提示

1. 用 for 將多筆同義詞讀入。
2. 用 split() 分出兩個詞。
3. 將同義詞關係填入字典（雙向、兩次）。

```
n = int(input())
mydict = {}
for i in range(n):
    word, synonym = input().split()
    mydict[word] = synonym
    mydict[synonym] = word
print(mydict[input()])
```

參考答案

字典 (dict) 資料結構

▶ 字典的值也可以是列表或者是另一個字典。

```
scores = {}  
scores['張三'] = [100, 90, 92]  
scores['李四'] = []  
scores['李四'] += [85]  
scores['李四'] += [75]
```

```
print(scores['張三'][0]) #此時 scores 字典的內容如下  
print(scores['李四'][1]) #{'張三': [100, 90, 92], '李四': [85, 75]}
```

scores

'張三'	[100, 90, 92]
'李四'	[85, 75]

100
75

```
students = {}  
students['張三'] = {}  
students['張三']['出生地'] = '台北'  
students['張三']['體重'] = 70  
students['張三']['考試分數'] = [100, 80, 85]
```

```
stu = input('學生姓名>')  
if stu in students:  
    print(students[stu])
```

students

'張三'	'出生地'	'台北'
	'體重'	70
	'考試分數'	[100, 80, 85]

學生姓名>張三

{'出生地': '台北', '體重': 70, '考試分數': [100, 80, 85]}

字典應用：計數

▶ 字典有個常見應用是計數。

■ 舉例：計算投票數

```
votes = ['張三', '李四', '張三', '張三', '李四']
```

```
counts = {}
for v in votes:
    if v not in counts:
        counts[v] = 1
    else:
        counts[v] = counts[v]+1

query = input('候選人姓名>')
if query in counts:
    print('得票數:', counts[query])
else:
    print('查無此人')
```

votes

'張三'	3
'李四'	2

候選人姓名>張三
得票數： 3

字典應用：計數

▶ 如果不喜歡程式裡有個 `if ... else ...`，字典的 `get()` 函式可以設定鍵不存在時的回傳值。

```
mydict = {}  
print(mydict.get('張三', 0)) #張三不存在，回傳 0  
mydict['張三'] = 4  
print(mydict.get('張三', 0)) #張三存在，回傳4
```

0
4

字典應用：計數

▶ 字典有個常見應用是計數。

■ 舉例：計算投票數

```
votes = ['張三', '李四', '張三', '張三', '李四']

counts = {}
for v in votes:
    counts[v] = counts.get(v, 0) + 1

query = input('候選人姓名>')
if query in counts:
    print('得票數:', counts[query])
else:
    print('查無此人')
```

候選人姓名>張三
得票數: 3

Snakify 線上練習時間



▶ 請完成 *Snakify* 平台主題

11 Dictionaries 的習題1 Number of occurrences。

- 給定一組字串，對每個字串輸出之前出現過的次數。

範例輸入	範例輸出
one two one tho three	0 0 1 0 0

提示

1. 用 `split()` 分出每個字串。
2. 對每個字串，先查詢，再更新字典中的次數。

參考答案

```
mydict = {}  
for word in input().split():  
    print(mydict.get(word, 0), end=' ')  
    mydict[word] = mydict.get(word, 0) + 1
```

字典應用：計數

▶ 自行車竊案分析

字典版

```
results = {} #存放統計結果的字典
for record in data:
    place = GetPlace(record)
    results[place] = results.get(place, 0) + 1
print(results)
```

列表版

```
results = [] #存放統計結果的列表
for record in data:
    place = GetPlace(record)

    #檢查 results 裡面有沒有出現過 place
    #如果有，將竊案數加 1
    found = False
    for e in results:
        if e[0]==place:
            e[1] += 1
            found = True
            break
    #如果沒有，新增 place，竊案數設為 1
    if not found:
        results.append([place, 1])

print(results)
```

走訪 dict 資料

▶ 使用 `for ... in ...` 句型走訪字典。

```
mydict = {'book': '書', 'computer': '電腦', 'program': '程式'}
```

```
for e in mydict:  
    print(e)
```

```
for e in mydict.keys():  
    print(e)
```

```
for e in mydict.items():  
    print(e)
```

```
for e in mydict.values():  
    print(e)
```

mydict

'book'	'書'
'computer'	'電腦'
'program'	'程式'

```
book  
computer  
program  
book  
computer  
program  
( 'book', '書' )  
( 'computer', '電腦' )  
( 'program', '程式' )  
書  
電腦  
程式
```



從 Python 3.7 起，走訪字典內容的順序和插入 (鍵, 值) 的順序相同。

走訪 dict 資料

▶ 使用 `for ... in ...` 句型走訪字典。

■ 應用：在計數後列印出所有結果。

```
votes = ['張三', '李四', '張三', '張三', '李四']
```

```
counts = {}
```

```
for v in votes:
```

```
    counts[v] = counts.get(v, 0) + 1
```

```
for e in counts:
```

```
    print(e, '得了', counts[e], '票')
```

counts

'張三'	3
'李四'	2

張三 得了 3 票
李四 得了 2 票

走訪 dict 資料

▶ 以鍵排序走訪字典

```
books = {9789863123798: 'Python 程式設計超入門',  
         9789864342549: 'Python 程式設計實例入門',  
         9789862487488: '書呆與阿宅'}
```

```
for e in sorted(books.keys()):  
    print(e, books[e])
```

9789862487488 書呆與阿宅
9789863123798 Python 程式設計超入門
9789864342549 Python 程式設計實例入門

走訪 dict 資料

▶ 以值排序走訪字典

```
votes = ['王五', '張三', '李四', '張三', '張三', '李四']

counts = {}
for v in votes:
    counts[v] = counts.get(v, 0) + 1

def cmp(x):
    return x[1]

for e in sorted(counts.items(), key=cmp, reverse=True):
    print(e[0], '得了', e[1], '票')
```

張三	得了	3	票
李四	得了	2	票
王五	得了	1	票

💡 如果是直接印，依序會是王五、張三、李四。

```
for e in counts:
    print(e, '得了', counts[e], '票')
```

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題3 Elections in the USA。

- 給定 n 筆候選人姓名與各州得票數，累計總得票數，並依候選人姓名排序列出統計結果。

範例輸入	範例輸出
5 Obama 9 McCain 10 McCain 5 Obama 8 McCain 1	McCain 16 Obama 17

提示

1. 用 `split()` 分出每個字串。
2. 使用前面講過的計數技巧。
3. 使用剛剛講過的依鍵排序技巧 (排 `keys()`)。

```
n = int(input())
mydict = {}
for _ in range(n):
    name, vote = input().split()
    mydict[name] = mydict.get(name, 0) + int(vote)

for name in sorted(mydict.keys()):
    print(name, mydict[name])
```

參考答案

Counter: 一種內建的計數dict

▶ 傳入資料，自動計數！

```
from collections import Counter
```

```
c = Counter('abcdabcaa')  
print(c)  
print(c['a'])
```

```
Counter({'a': 4, 'b': 2, 'c': 2, 'd': 1})  
4
```

```
from collections import Counter
```

```
c = Counter(['張三', '張三', '李四', '李四', '張三'])  
print(c)  
print(c['張三'])
```

```
Counter({'張三': 3, '李四': 2})  
3
```

▶ 也可以使用 dict 的慣用初始化語法

```
from collections import Counter
```

```
c = Counter({'張三':3, '李四':2})  
print(c)  
print(c['張三'])
```

```
Counter({'張三': 3, '李四': 2})  
3
```


Counter: 一種內建的計數dict

▶ 不存在時預設值為 0

```
from collections import Counter

m = dict()
print(m['a'])    #錯誤：KeyError

c = Counter()
print(c['a'])    #沒問題
```

0

▶ 有個 most_common() 函式可以傳回次數最高的前 n 個鍵-值

對

```
from collections import Counter

c = Counter('acbcdabcaab')
print(c)
print(c.most_common(1))
print(c.most_common(2))
print(c.most_common())
```

```
Counter({'a': 4, 'c': 3, 'b': 3, 'd': 1})
[('a', 4)]
[('a', 4), ('c', 3)]
[('a', 4), ('c', 3), ('b', 3), ('d', 1)]
```

Counter: 一種內建的計數dict

▶ 自行車竊案

- 短短數行程式便可統計並列印出竊案數前五高的行政區。

```
from collections import Counter

f = open('臺北市10401-10709自行車竊盜點位資訊.txt', encoding='utf-8')
f.readline()
data = [e.split() for e in f]
c = Counter([e[4][3:6] for e in data])
for e in c.most_common(5):
    print(e[0], '竊案數', e[1])
```

大安區	竊案數	235
北投區	竊案數	133
中正區	竊案數	125
信義區	竊案數	121
文山區	竊案數	114

```
data = [
    ['1', '自行車竊盜', '1020630', '13~15', '台北市松山區三民路151', '~', '180號'],
    ['2', '自行車竊盜', '1040101', '16~18', '台北市大安區住安里四維路124巷1~30號'],
    ['3', '自行車竊盜', '1040104', '07~09', '台北市大安區敦煌里仁愛路四段266巷1~30號'],
    ...
]
```

Counter: 一種內建的計數dict

▶ Counter 的算術運算

運算	運算式	運算結果
Add	$x+y$	將次數相加。
Subtract	$x-y$	x 次數減 y 的次數，刪除次數非正的鍵值。
Intersect	$x\&y$	取兩者次數的較小值。
Union	$x y$	取兩者次數的較大值。

```
from collections import Counter
```

```
x = Counter('abccdadabcaab')
```

```
y = Counter('abccccd')
```

```
print('  x', x)
```

```
print('  y', y)
```

```
print('x+y', x+y)
```

```
print('x-y', x-y)
```

```
print('x&y', x&y)
```

```
print('x|y', x|y)
```

```
x Counter({'a': 4, 'b': 3, 'c': 3, 'd': 1})
y Counter({'c': 4, 'a': 1, 'b': 1, 'd': 1})
x+y Counter({'c': 7, 'a': 5, 'b': 4, 'd': 2})
x-y Counter({'a': 3, 'b': 2})
x&y Counter({'c': 3, 'a': 1, 'b': 1, 'd': 1})
x|y Counter({'a': 4, 'c': 4, 'b': 3, 'd': 1})
```

Snakify 題目解析

字典資料結構

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題4 The most frequent word。

- 給定 n 行字串，計算每個字串的總出現次數，印出次數最高的字串。若有多個字串的次數都最高，印出字母順序列小者。

範例輸入	範例輸出
3 q w e r t y u i o p a s d f g h j k l z x c v b n m	a

提示

1. 用 `split()` 分出每個字串。
2. 使用前面講過的計數技巧。
3. 以 `sorted()` 將字典的 `items()` 排序。
4. 排序的函式要先看次數（越大越前面），再看字串（越小越前面）。

Snakify 線上練習時間

▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題4 The most frequent word。

- 給定 n 行字串，計算每個字串的總出現次數，印出次數最高的字串。若有多個字串的次數都最高，印出字母順序列小者。

範例輸入	範例輸出
3 q w e r t y u i o p a s d f g h j k l z x c v b n m	a

```
n = int(input())
mydict = {}
for i in range(n):
    words = input().split()
    for w in words:
        mydict[w] = mydict.get(w, 0) + 1

def cmp(x):
    return -x[1], x[0] #次數越大排序時視為越小，所以加負號
res = sorted(mydict.items(), key=cmp)
print(res[0][0])
```

參考答案A

Snakify 線上練習時間

▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題4 The most frequent word。

- 給定 n 行字串，計算每個字串的總出現次數，印出次數最高的字串。若有多個字串的次數都最高，印出字母順序列小者。

範例輸入	範例輸出
3 q w e r t y u i o p a s d f g h j k l z x c v b n m	a

```
n = int(input())
mydict = {}
for i in range(n):
    words = input().split()
    for w in words:
        mydict[w] = mydict.get(w, 0) + 1

max_count = max(mydict.values())
most_frequent = [k for k in mydict if mydict[k] == max_count]
print(min(most_frequent))
```

參考答案B

(不需要排序)
先找出最大次數，再找出符合最大次數
的字串的最小值。

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題5 Access rights 。 (本題稍難一滴滴)

- 給定 n 組檔案與其允許存取權限 (W, R, X) 。
- 多次查詢對檔案操作是否可行。

範例輸入	範例輸出
4 helloworld.exe R X pinglog W R nya R goodluck X W R 3 read nya write helloworld.exe execute nya	OK Access denied Access denied

- 提示
1. 對每個檔案，填入 (程式, [存取權限]) 的關係到字典中。
 2. 可自建第二個字典 {'read': 'R', 'write', 'W', 'execute', 'X'}，以方便查詢。
 3. 查詢權限是否存在於列表內容，請愛用 in 運算。

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題5 Access rights 。 (本題稍難一滴滴)

- 給定 n 組檔案與其允許存取權限 (W, R, X) 。
- 多次查詢對檔案操作是否可行 。

參考答案

```
n = int(input())
permission = {}
for i in range(n):
    line = input().split()
    permission[line[0]] = line[1:]

action = {'write': 'W', 'execute': 'X', 'read': 'R'}
q = int(input())
for i in range(q):
    a, file = input().split()
    if action[a] in permission[file]:
        print('OK')
    else:
        print('Access denied')
```

範例輸入

```
4
helloworld.exe R X
pinglog W R
nya R
goodluck X W R
3
read nya
write helloworld.exe
execute nya
```

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題6 Countries and cities。

- 給定 n 組國家與其城市的資料。
- 多次查詢特定城市的所在國家。

範例輸入	範例輸出
2 USA Boston Pittsburgh Washington Seattle UK London Edinburgh Cardiff Belfast	UK USA UK
3 Cardiff Seattle London	

提示

1. 用 for 將多筆資料讀入。
2. 用 split() 分出國家和城市 (會有不只一個城市)。
3. 對每個城市，填入 (城市, 國家) 的關係到字典中。

參考答案

```
n = int(input())
mydict = {}
for i in range(n):
    line = input().split()
    for city in line[1:]:
        mydict[city] = line[0]

q = int(input())
for i in range(q):
    print(mydict[input()])
```

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題7 Frequency analysis。

- 給定 n 行字串，計算每個字串的總出現次數。將所有字串依出現次數由高到低列印出。若次數相同，依字母順序印出。
- 跟習題4 幾乎一樣。

Snakify 線上練習時間



▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題8 English-Latin dictionary。

- 給定 n 行字串，每行為一個英文單字和一個以上相通的拉丁文單字。
- 請製作一個從拉丁文單字反查英文單字的字典。
- 列印出字典中所有拉丁文單字和其相通的英文單字，列印時依拉丁文單字的字母順序。

範例輸入	範例輸出
3 apple - malum, pomum, popula fruit - baca, bacca, popum punishment - malum, multa	7 baca - fruit bacca - fruit malum - apple, punishment multa - punishment pomum - apple popula - apple popum - fruit

Snakify 線上練習時間



▶ 請完成 *Snakify* 平台主題

11 Dictionaries 的習題8 English-Latin dictionary。

- 給定 n 行字串，每行為一個英文單字和一個以上相通的拉丁文單字。
- 請製作一個從拉丁文單字反查英文單字的字典。
- 列印出字典中所有拉丁文單字和其相通的英文單字，列印時依拉丁文單字的字母順序。

提示

1. 用 `split()` 分出每個字串。記得 `split()` 可以傳入分隔字元(串)，例如 `split(' - ')`。
2. 每個拉丁文單字對應一個英文單字列表，也就是拉丁文單字為鍵，值為一個字串列表。
3. 本題要花點心思處理輸入字串格式與輸出字串格式。

Snakify 線上練習時間

▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題8 English-Latin dictionary。

- 給定 n 行字串，每行為一個英文單字和一個以上相通的拉丁文單字。
- 請製作一個從拉丁文單字反查英文單字的字典。
- 列印出字典中所有拉丁文單字和其相通的英文單字，列印時依拉丁文單字的字母順序。

參考答案A

```
n = int(input())
mydict = {}
for i in range(n):
    eng, latins = input().split(' - ')
    latins = latins.split(',')
    for latin in latins:
        mydict[latin] = mydict.get(latin, []) + [eng]

print(len(mydict))
for latin in sorted(mydict.keys()):
    print(latin, '-', mydict[latin][0], end='')
    for e in mydict[latin][1:]:
        print(', ', e, sep='', end='')
    print()
```

範例輸入

```
3
apple - malum, pomum, popula
fruit - baca, bacca, popum
punishment - malum, multa
```

範例輸出

```
7
baca - fruit
bacca - fruit
malum - apple, punishment
multa - punishment
pomum - apple
popula - apple
popum - fruit
```

Snakify 線上練習時間

▶ 請完成 Snakify 平台主題

11 Dictionaries 的習題8 English-Latin dictionary。

- 給定 n 行字串，每行為一個英文單字和一個以上相通的拉丁文單字。
- 請製作一個從拉丁文單字反查英文單字的字典。
- 列印出字典中所有拉丁文單字和其相通的英文單字，列印時依拉丁文單字的字母順序。

參考答案B

```
n = int(input())
mydict = {}
for i in range(n):
    eng, latins = input().split(' - ')
    latins = latins.split(',')
    for latin in latins:
        mydict[latin] = mydict.get(latin, []) + [eng]

print(len(mydict))
for latin in sorted(mydict.keys()):
    print(latin, '-', ', '.join(mydict[latin]))
```



split() 用來分開字串成字串列表，join() 用來將字串列表合併成字串。

```
print(', '.join(['a', 'b', 'c']))
print('- '.join(['aa', 'bb', 'cc']))
```

範例輸入

```
3
apple - malum, pomum, popula
fruit - baca, bacca, popum
punishment - malum, multa
```

範例輸出

```
7
baca - fruit
bacca - fruit
malum - apple, punishment
multa - punishment
pomum - apple
popula - apple
popum - fruit
```