

# Python60 系列

# 發票對號小程序 (II)



</PLUS>

推動大學(程式設計)教學  
MOE/TAIEP/AC/NTU/USTC/THU

推動大學程式設計教學計畫。分項六：資料分析領域與學習評量推動團隊（臺師大資工系蔣宗哲老師主編）

# 發票對號 II：課程簡述

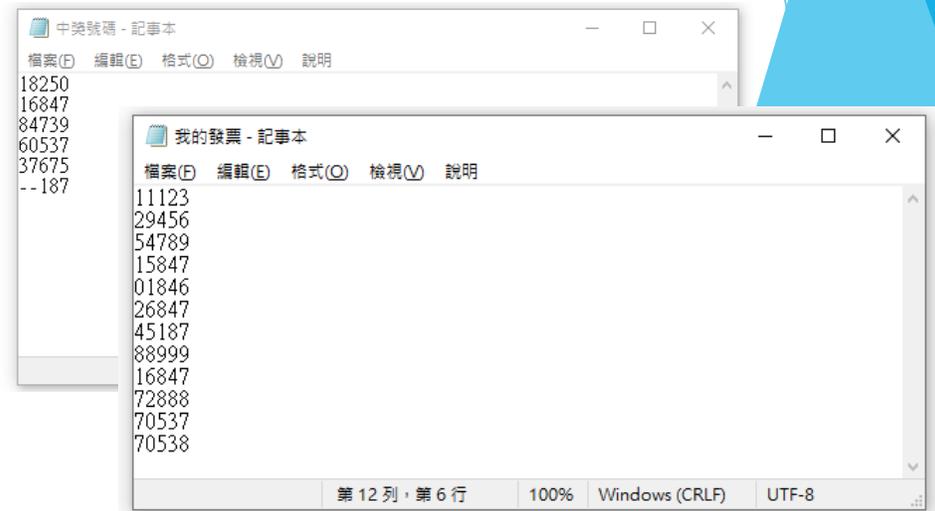
## ▶ 問題情境：

- 給定若干發票號碼 (5 碼) 以及若干個待比對的發票號碼，撰寫一個程式比對有幾張發票中獎。
- 中獎號碼可能為 3, 4, 或 5 碼相符。
- 中獎號碼和待比對號碼均以純文字檔儲存。



## ▶ 本課程將撰寫一個 10 行的程式，實際課程時間 20-30 分鐘。

還沒看過「發票對號」第一集課程的程式新手，建議先看完該課程教材喔～



# 發票對號 II：學習內容

## ▶ 程式設計概念

- 雙層迴圈控制 `for`
- 多分法選擇控制 `if`
- 巢狀控制流程
- 字串處理

## ▶ 資料分析概念

- 搜尋與比對

# 發票對號 II

## 完整程式展示

```
win = open('中獎號碼.txt').read().split()
mine = open('我的發票.txt').read().split()

for i, num in enumerate(mine):
    for win_num in win:
        if num[-5:] == win_num[-5:]:
            print('第', i+1, '張發票對中號碼', num[-5:], '!!')
        elif num[-4:] == win_num[-4:]:
            print('第', i+1, '張發票對中號碼', num[-4:], '!!')
        elif num[-3:] == win_num[-3:]:
            print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```

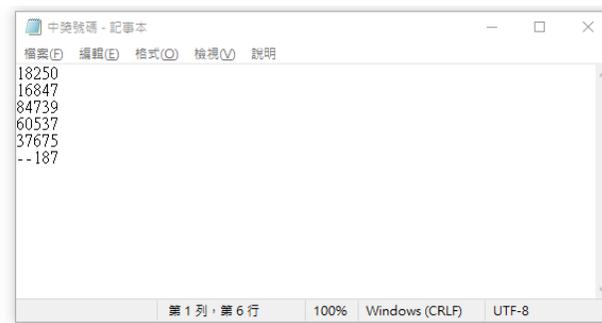
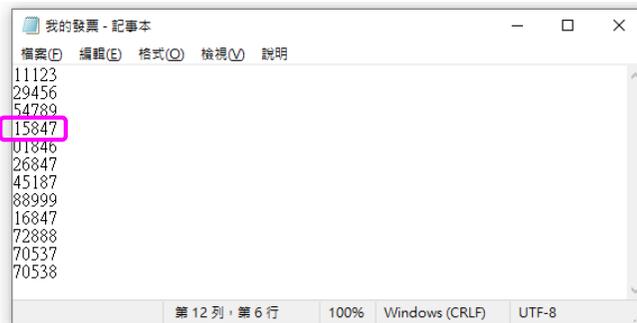
第 4 張發票對中號碼 847 !  
第 6 張發票對中號碼 6847 !  
第 7 張發票對中號碼 187 !  
第 9 張發票對中號碼 16847 !  
第 11 張發票對中號碼 0537 !



中獎號碼



我的發票



# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()
02 mine = open('我的發票.txt').read().split()
03
04 for i, num in enumerate(mine):
05     for win_num in win:
06         if num[-5:] == win_num[-5:]:
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')
08         elif num[-4:] == win_num[-4:]:
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')
10         elif num[-3:] == win_num[-3:]:
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```

- ▶ 本次學習的程式如上所示，共 11 行。（第 3 行空行僅為排版美觀，刪除亦不影響功能。）
- ▶ 第 01-02 行開啟純文字檔並讀入資料，細節已於「發票對號 I」課程講述，此處省略。
- ▶ 第 04、05 行使用雙層重覆控制 for 句型，意在比對 mine 中的號碼和 win 中的號碼。
- ▶ 第 06-11 行使用多分法選擇控制，依序以 5碼、4碼、3碼 來比對 num 和 win\_num。
- ▶ 本次課程的主軸便是放在雙層 for 句型以及多分法 if 句型。

# 程式概念：迴圈控制

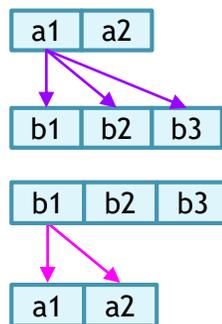
```
01 a = ['123', '456', '789']
02 for i, num in enumerate(a):
03     print(i, num)
04 for num in a:
05     print(num)
```

```
0 123
1 456
2 789
123
456
789
```

- ▶ 上方程式碼中，a 為一個列表，包含三個字串。
- ▶ for ... in 句型可以依序取出列表中的各個資料，並對各資料重覆執行相同的程式動作。
- ▶ 搭配 enumerate() 使用時，for 和 in 之間需有兩個變數，如程式碼第 02 行：
  - 前者 ( 程式碼第 02 行中的 i ) 代表資料序號，由 0 開始，每次遞增 1；
  - 後者 ( 程式碼第 02 行中的 num ) 代表列表中的資料。
  - 第 02-03 行程式碼的輸出共三行，依序為 0 123、1 456 和 2 789。
- ▶ 不需要資料序號時，我們可以省去 enumerate()，如程式碼第 04 行：
  - num 代表列表中的資料。
  - 第 04-05 行程式碼的輸出共三行，依序為 123、456 和 789。

# 程式概念：雙層迴圈控制

```
01 a = ['a1', 'a2']
02 b = ['b1', 'b2', 'b3']
03
04 for x in a:
05     print(x)
06
07 for y in b:
08     print(y)
09
10 for x in a:
11     for y in b:
12         print(x, y)
13
14 for x in b:
15     for y in a:
16         print(x, y)
```

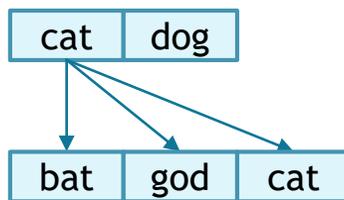


```
a1
a2
b1
b2
b3
a1 b1
a1 b2
a1 b3
a2 b1
a2 b2
a2 b3
b1 a1
b1 a2
b2 a1
b2 a2
b3 a1
b3 a2
```

- ▶ for 句型中內縮的程式碼代表要對資料重覆執行的動作。
- ▶ 左方程式碼第 04-05 行和第 07-08 行分別以 for 句型取出列表 a 和 b 資料並列印出來。
- ▶ 重覆執行的動作本身可以又是一個 for 句型。
- ▶ 第 10 程式碼以 x 代表列表 a 中的資料，並在固定一個 x 的情況下，執行第 11-12 行。
  - 當 x 為 'a1' 時，第 11 行的 for 會令 y 依序為 'b1', 'b2', 'b3'，並執行 print(x, y)，因此會列印出 3 行，分別為 a1 b1、a1 b2 和 a1 b3。
- ▶ 對比第 10-12 行，第 14-16 行是以 x 代表列表 b 中的資料，並在固定一個 x 的情況下，執行第 15-16 行。

# 程式概念：雙層迴圈控制

```
01 a = ['cat', 'dog']
02 b = ['bat', 'god', 'cat']
03
04 for x in a:
05     for y in b:
06         print(x, y)
07
08 for x in a:
09     for y in b:
10         if x==y:
11             print(x, '兩者並存')
```



```
cat bat
cat god
cat cat
dog bat
dog god
dog cat
cat 兩者並存
```

- ▶ 本次課程主題為「發票對號」，其程式流程是拿每一個「待對號碼」比對每一個「中獎號碼」。因為待對號碼有很多個，中獎號碼也有很多個，所以可以想見若各用一個 for 句型去取得號碼，程式結構就會形成兩個層次的迴圈重覆控制。
- ▶ 上方程式碼第 04-06 行可列印列表 a 和列表 b 中所有資料組合的結果，共  $2 \times 3 = 6$  行。
- ▶ 如果我們把重覆執行的動作加上一個條件，只有在 x 和 y 相等時才列印（如程式碼第 10 行），我們就能用程式判斷是否資料是否同時存在兩個列表中。
  - 因此第 08-11 行會檢查六種組合，並對其中一個組合（x 為 'cat' 且 y 為 'cat'）執行 print()。

# 程式概念：選擇控制與關係運算

```
01 a = 'Python'  
02  
03 print('開始判斷')  
04 if a == 'Python':  
05     print('條件成立')  
06 print('結束判斷')  
07  
08 a = '黑'  
09 print('非黑即白')  
10 if a == '黑':  
11     print('是黑的')  
12 else:  
13     print('是白的')  
14 print('結束判斷')
```

別忘了被 if 句型控制的程式碼要內縮唷~

開始判斷  
條件成立  
結束判斷  
非黑即白  
是黑的  
結束判斷

- ▶ 最簡單的 if 句型可以控制一段程式碼是否要執行，例如上方程式碼的第 04-05 行會判斷變數 a 代表的資料是否和字串 'Python' 相等，若相等才會執行第 05 行。第 03 和 06 行程式則是無條件執行。
- ▶ 進階一點的 if 句型會帶有 else 部份，形成二分法，也就是從兩段程式碼擇一執行，如程式碼的第 10-13 行。
- ▶ 大家可以試試在第 01 或 08 行把 a 換成別的值，再觀察程式的執行結果。

# 程式概念：選擇控制與多分法

```
01 ans = input('你喜歡作哪種運動?')
02 print('我也喜歡', ans, '耶!')
03
04 if ans == '籃球':
05     print('你喜歡打三對三鬥牛嗎?')
06 elif ans == '棒球':
07     print('你是投手還是野手?')
08 elif ans == '羽球':
09     print('你喜歡戴資穎嗎?')
10 else:
11     print('改天一起練吧!')
```

你喜歡作哪種運動?籃球  
我也喜歡 籃球 耶!  
你喜歡打三對三鬥牛嗎?

你喜歡作哪種運動?棒球  
我也喜歡 棒球 耶!  
你是投手還是野手?

你喜歡作哪種運動?羽球  
我也喜歡 羽球 耶!  
你喜歡戴資穎嗎?

你喜歡作哪種運動?跑步  
我也喜歡 跑步 耶!  
改天一起練吧!

- ▶ 從二分法再延伸成多分法時，可在 `if` 句型中增加若干個 `elif` 部份，如上方程式碼第 04-11 行構成的 `if ... elif ... else` 句型。
- ▶ 本例中是以 `ans` 變數代表的資料將程式區分成四種可能性–籃球、棒球、羽球還有其它。
  - 注意，只有當上面的條件不成立時，才會檢查下面的條件。例如若第 04 行的條件已成立，就不會再檢查第 06 或 08 行的條件，也不會符合 `else`。
  - 第 05、07、09 和 11 行的程式只會有一個被執行。
  - 每一段內縮的程式碼可以有不只一行，但要確認程式碼對齊，即縮排格數相同。

# 程式概念：子字串 (slice)

```
01 a = 'Python'  
02 print(a) [0] [1] [2] [3] [4] [5]  
03 print(a[0], a[1], a[2])  
04 print(a[0:2], a[1:3])  
05 print(a[:1], a[:2])  
06 print(a[-3:], a[-1:])  
07  
08 if a[-3:] == 'hon':  
09     print(a, '的最後三個字母是 hon')
```

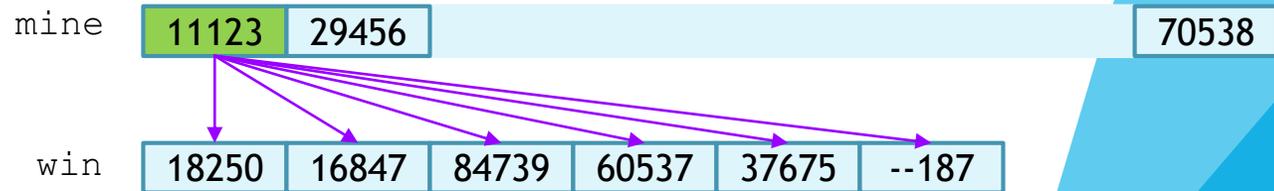
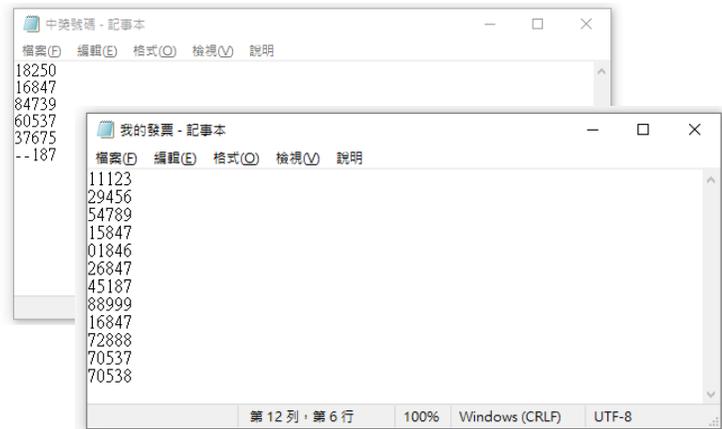


```
Python  
P y t  
Py yt  
P Py  
hon n  
Python 的最後三個字母是 hon
```

- ▶ Python 語言中以一對單引號或雙引號包起來的資料為字串型態，上方程式碼的第 01 行令變數 a 代表字串 'Python'，第 02 行將 a 的內容完整列印出來。
- ▶ 我們可以用 [] 取用字串中的單一個字元，如第 03 行分別列印出 a 的第一、二、三個字元。
  - [] 中的索引值編號從 0 開始。
- ▶ 我們還可以用 [開始:結束] 來取用字串的一段子字串，如程式第 04-06 行和 08 行所示。
  - 此處的「結束」編號，精確來說，是結束位置的下一個位置的編號。因此 a[0:2] 指的是由 a[0] 和 a[1] 構成的子字串，不包含 a[2]。
  - 若省略開始編號，代表從字串的頭開始，若省略結束編號，則代表到字串的尾結束。
  - 編號若為負值，代表從字串尾巴往前數，例如 -1 代表倒數第 1 個位置，-2 代表倒數第 2 個位置。

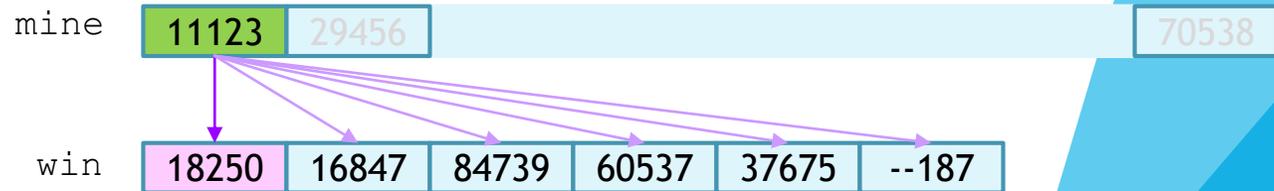
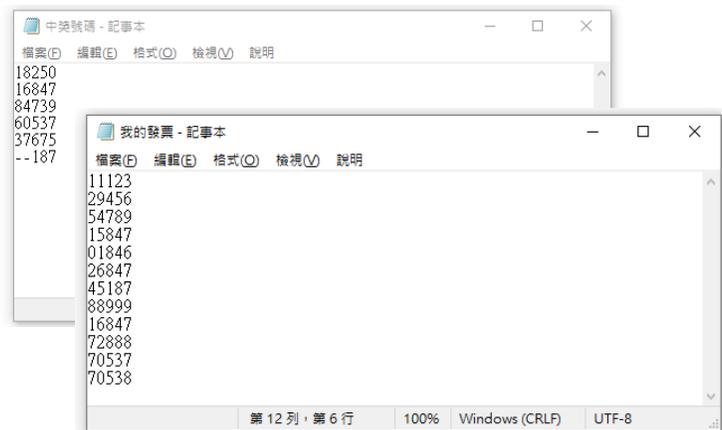
# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()  
02 mine = open('我的發票.txt').read().split()  
03  
04 for i, num in enumerate(mine):  
05     for win_num in win:  
06         if num[-5:] == win_num[-5:]:  
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')  
08         elif num[-4:] == win_num[-4:]:  
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')  
10         elif num[-3:] == win_num[-3:]:  
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```



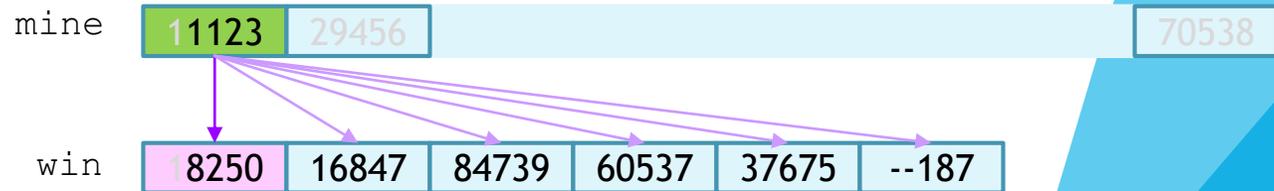
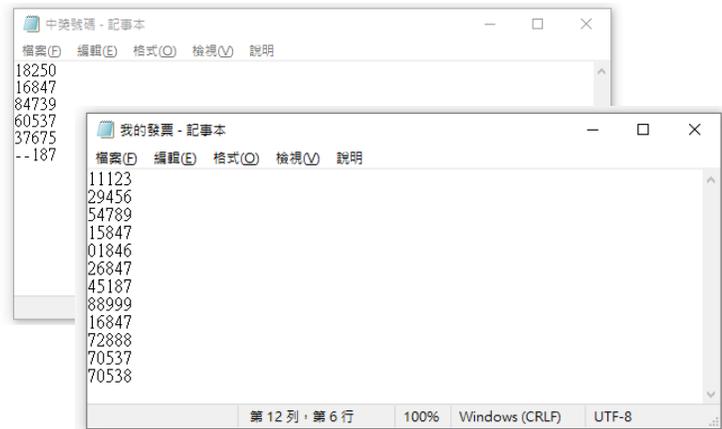
# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()  
02 mine = open('我的發票.txt').read().split()  
03  
04 for i, num in enumerate(mine):  
05     for win_num in win:  
06         if num[-5:] == win_num[-5:]:  
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')  
08         elif num[-4:] == win_num[-4:]:  
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')  
10         elif num[-3:] == win_num[-3:]:  
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```



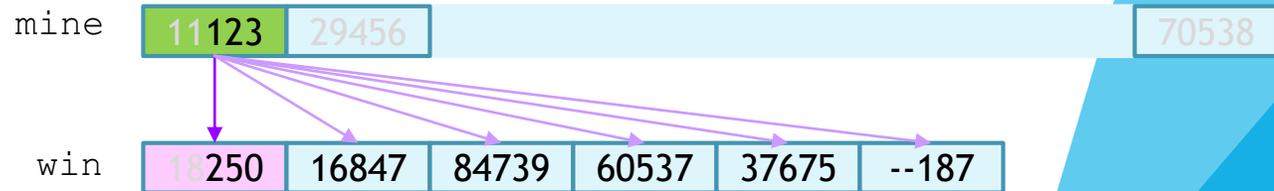
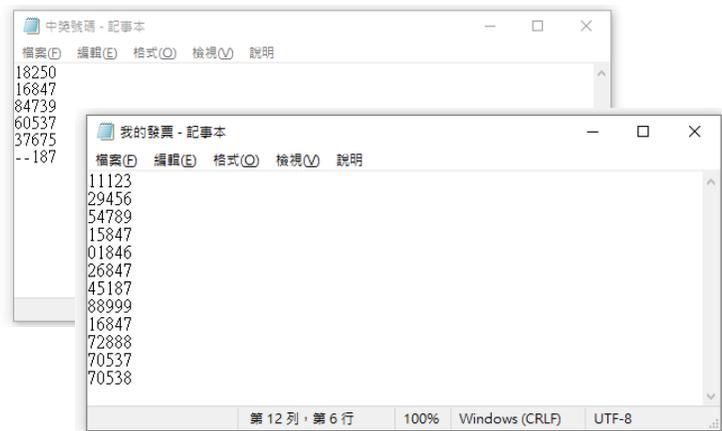
# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()  
02 mine = open('我的發票.txt').read().split()  
03  
04 for i, num in enumerate(mine):  
05     for win_num in win:  
06         if num[-5:] == win_num[-5:]:  
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')  
08         elif num[-4:] == win_num[-4:]:  
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')  
10         elif num[-3:] == win_num[-3:]:  
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```



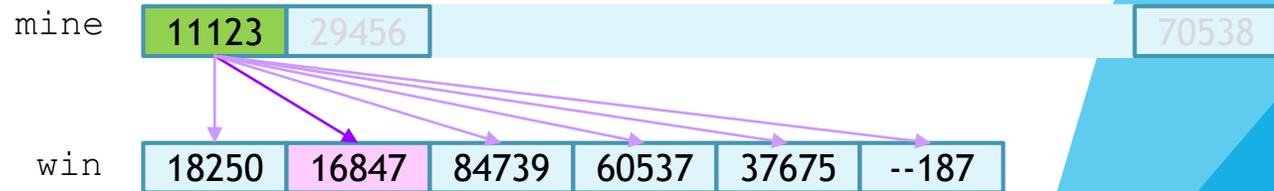
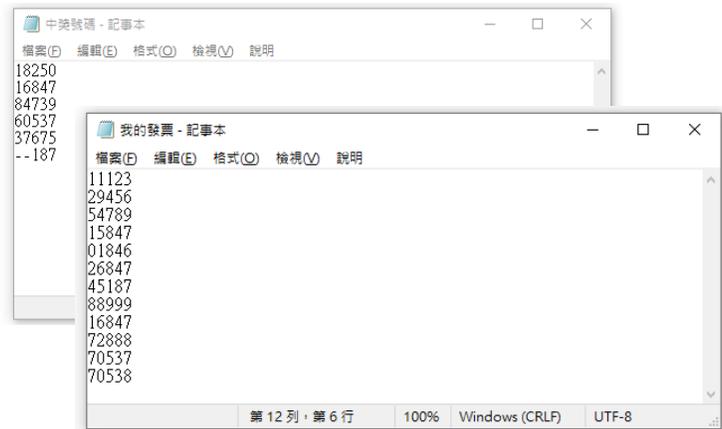
# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()  
02 mine = open('我的發票.txt').read().split()  
03  
04 for i, num in enumerate(mine):  
05     for win_num in win:  
06         if num[-5:] == win_num[-5:]:  
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')  
08         elif num[-4:] == win_num[-4:]:  
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')  
10         elif num[-3:] == win_num[-3:]:  
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```



# 程式碼解說

```
01 win = open('中獎號碼.txt').read().split()  
02 mine = open('我的發票.txt').read().split()  
03  
04 for i, num in enumerate(mine):  
05     for win_num in win:  
06         if num[-5:] == win_num[-5:]:  
07             print('第', i+1, '張發票對中號碼', num[-5:], '!!')  
08         elif num[-4:] == win_num[-4:]:  
09             print('第', i+1, '張發票對中號碼', num[-4:], '!!')  
10         elif num[-3:] == win_num[-3:]:  
11             print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```



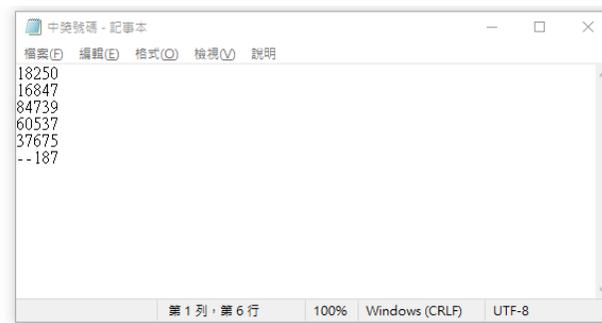
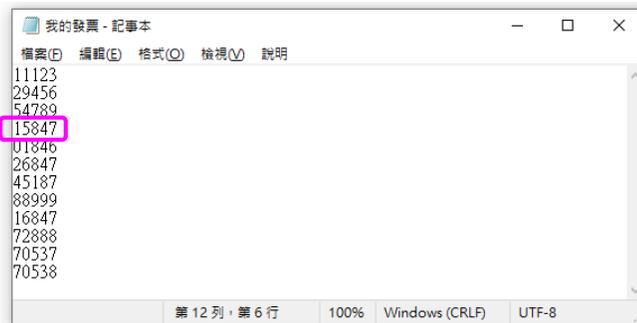
# 發票對號 II

## 完整程式展示

```
win = open('中獎號碼.txt').read().split()
mine = open('我的發票.txt').read().split()

for i, num in enumerate(mine):
    for win_num in win:
        if num[-5:] == win_num[-5:]:
            print('第', i+1, '張發票對中號碼', num[-5:], '!!')
        elif num[-4:] == win_num[-4:]:
            print('第', i+1, '張發票對中號碼', num[-4:], '!!')
        elif num[-3:] == win_num[-3:]:
            print('第', i+1, '張發票對中號碼', num[-3:], '!!')
```

第 4 張發票對中號碼 847 !  
第 6 張發票對中號碼 6847 !  
第 7 張發票對中號碼 187 !  
第 9 張發票對中號碼 16847 !  
第 11 張發票對中號碼 0537 !



# 挑戰時刻

## ▶ 計算總獎金

- 請修改程式，使其能累計總獎金：對中末三位數 200 元，末四位數 1000 元，末五位數 4000 元。

## ▶ 改成全碼都能比對

- 請修改程式，使其能比對更多位數至八位數。

## ▶ 改成精簡程式碼

- 請修改程式，將 `if ... elif ...` 的多分法以 `for` 句型簡化。
- 此小題較難，若無法完成也沒關係。 😊