

Python60 系列

發票對號小程序 (I)



</PLUS>

推動大學(程式設計)教學
MOE-AC-105-2-1-010-A1-001

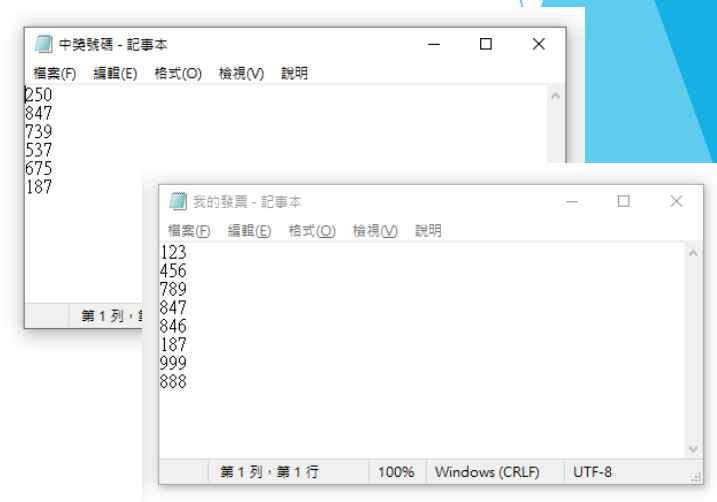
推動大學程式設計教學計畫。分項六：資料分析領域與學習評量推動團隊（臺師大資工系蔣宗哲老師主編）

發票對號：課程簡述

▶ 問題情境：

- 給定若干發票六獎號碼 (固定 3 個數字) 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。
- 中獎號碼和待比對號碼均以純文字檔儲存。

▶ 本課程將撰寫一個 5 行的程式，實際課程時間 20-30 分鐘。



發票對號：學習內容

▶ 程式設計概念

- 變數
- 純文字格式檔案讀取
- 迴圈控制 `for ... in ...`
- 選擇控制 `if`

▶ 資料分析概念

- 搜尋與比對

發票對號

▶ 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):
    if num in win:
        print('第', i+1, '張發票對中號碼', num, '!!')
```

程式輸出

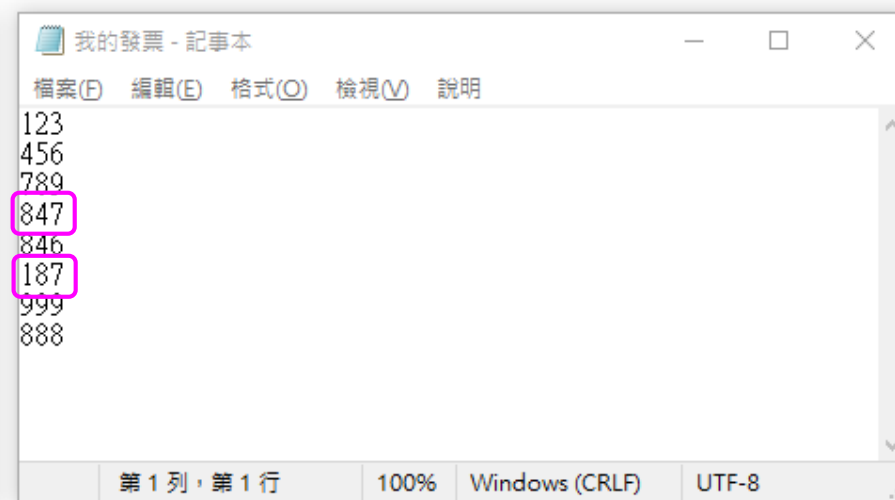
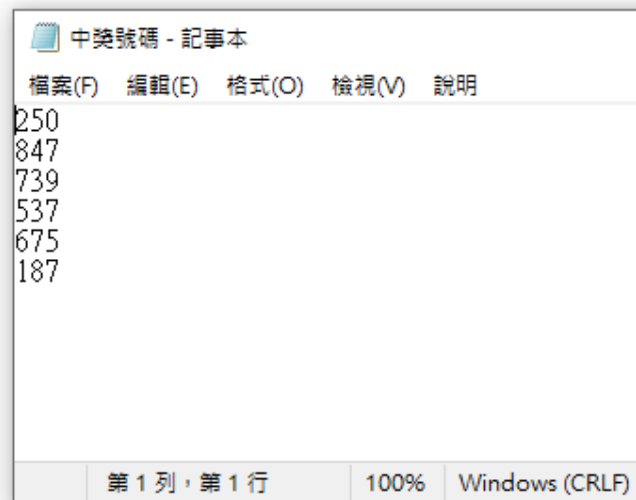
第 4 張發票對中號碼 847!
第 6 張發票對中號碼 187!



中獎號碼



我的發票



程式碼解說

▶ 重要觀念

- **函式**：函式像是一種動作，例如「列印」、「開啟檔案」、「計算平均數」等等。
- **資料**：資料有多種型態，像是整數、小數、文字、檔案等等。

▶ 資料與函式常用的程式碼型式

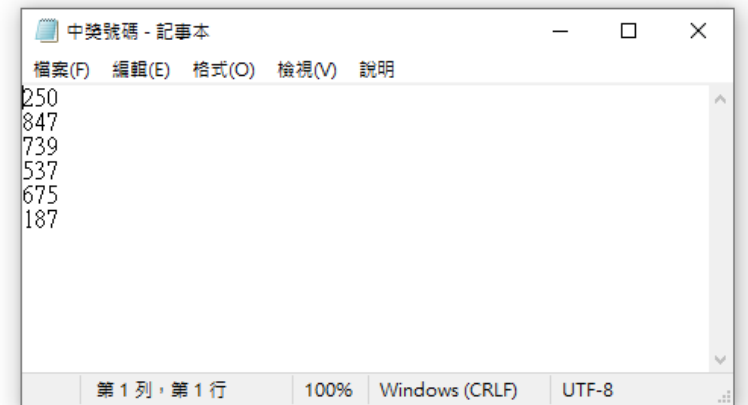
程式碼型式	程式碼舉例	程式碼意義
函式(資料)	<code>print('Hello')</code>	列印出 'Hello' 文字。
資料.函式()	<code>'Hello'.upper()</code>	產生 'Hello' 文字全大寫的結果。
資料.函式(資料)	<code>'a,b,c'.split(',')</code>	將 'a,b,c' 以 ',' 號分隔成三段。

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     if num in win:  
06         print('第', i+1, '張發票對中號碼', num, '!')
```

- ▶ 本程式完整程式碼如上，共六行。（第 3 行空行僅為排版美觀，刪除亦不影響功能。）
- ▶ 首先，我們解釋第 01-02 行的程式碼，這兩行程式碼的結構與意義相近。
- ▶ 第 01 行包含四個步驟：

- 使用 `open()` 函式開啟純文字檔，其檔名為 `中獎號碼.txt`。
- 開啟檔案後，使用 `read()` 函式讀入檔案中的文字。
- 接著使用 `split()` 函式將文字以換行切分。
- 最後將依序執行完 `open()`、`read()`、`split()` 三個動作的結果存入變數 `win` 中。
- `win` 儲存的結果為各行文字組成的列表，以右圖檔案內容為例，`win` 儲存的資料為 `['250', '847', '739', '537', '675', '187']`。



程式概念：變數

```
01 a = 10
02 b = 20
03 c = a*10 + b
04 d = '123'
```

- ▶ 變數用來儲存程式中會使用到的資料。
- ▶ 我們以 = 來建立與設定變數代表的內容。
- ▶ 以本頁上方的程式碼為例，我們一共定義了四個變數，其中 a 和 b 分別代表 10 和 20，c 代表 a 的值乘以 10 再加 b 的值，即為 120，而 d 代表文字 '123'。
- ▶ 注意事項：
 - 此處的語法一定要把變數放在 = 的左邊，因此不能寫成 $10 = a$ 。
 - 在程式中有兩種基本型態、數值和文字。在撰寫程式時應謹慎區分。

程式概念：檔案開啟與讀取

```
01 win = open('中獎號碼.txt').read().split()  
02  
03 f = open('中獎號碼.txt')  
04 text = f.read()  
05 win = text.split()
```

- ▶ Python 語言提供很多好用的函式。就像很多人都曾使用過的 Excel 的 SUM() 和 AVERAGE() 函式一樣，這些函式能快速地幫我們完成想要的功能。
- ▶ 本頁上方程式碼的第 01 行其實是第 03-05 行程式碼合併後的結果，我們依序說明第 03-05 行程式碼。
- ▶ open() 函式用來開啟檔案，我們需要將欲開啟的檔案名稱放在 () 裡面，並且用一對單引號（雙引號亦可）將名稱包起來，代表這是一段文字資料。第 03 行我們開啟檔案 *中獎號碼.txt*，並以變數 *f* 代表開啟的結果。（變數名稱不一定要是 *f*，也可命名為 *file*, *x*, ... 等等。）
- ▶ 當我們有一個 open() 函式開啟的檔案變數（第 03 行的 *f*），便可對它呼叫 read() 函式，如第 04 行所示。此時 read() 的結果會是整個文字檔案的內容，我們以變數 *text* 代表。
- ▶ 第 05 行是以 split() 將文字切斷，我們繼續在下一頁說明。

程式概念：字串與列表

```
01 a = '123'  
02 b = '456 789 888'  
03 print(a)  
04 print(a, b)  
05 c = b.split()  
06 print(c)
```

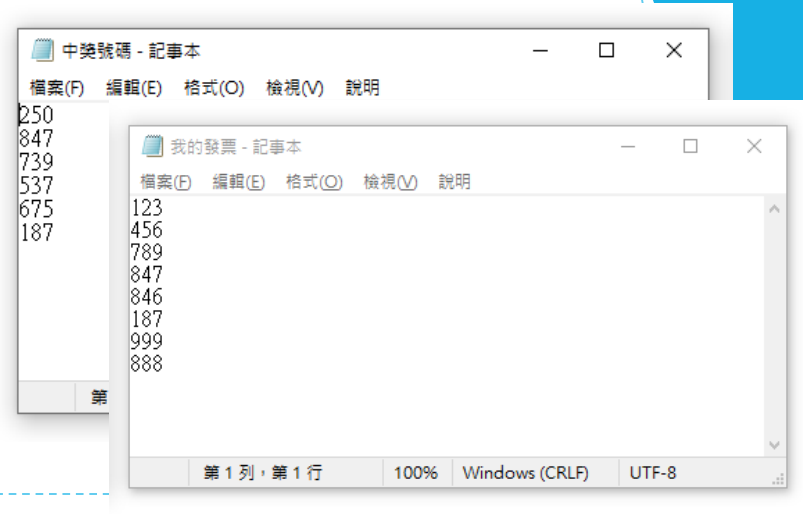
程式輸出

```
123  
123 456 789 888  
['456', '789', '888']
```

- ▶ 在 Python 程式中，我們將以一對單/雙引號包起來的資料稱為文字或字串 (str) 型資料。
- ▶ 本面上方第 01、02 行分別定義了兩個變數 a 和 b，並令它們代表字串 '123' 和 '123 456 789'。
- ▶ 第 03 行我們使用 print() 函式來列印變數 a 的內容，第 04 行則展示若需要列印多個變數時，僅需使用逗號，將兩個變數分開。
 - 預設以一個空格將列印的多個資料內容分開。預設會在最後加印一個換行。所以第 03 和 04 行程式碼列印的結果會分在兩行。
- ▶ Python 的字串可以呼叫 split() 函式，預設以空白、跳位和換行字元切割字串內容。由於變數 b 代表的資料是字串，所以我們可以對它作 split()，如第 05 行所示。我們將結果以變數 c 代表。
- ▶ 第 06 行我們使用 print() 列印出 c 的結果，可以看到原本字串 '456 789 888' 已經被切分為三個字串 '456'、'789' 和 '888' 所組成的列表。

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     if num in win:
06         print('第', i+1, '張發票對中號碼', num, '!!')
```



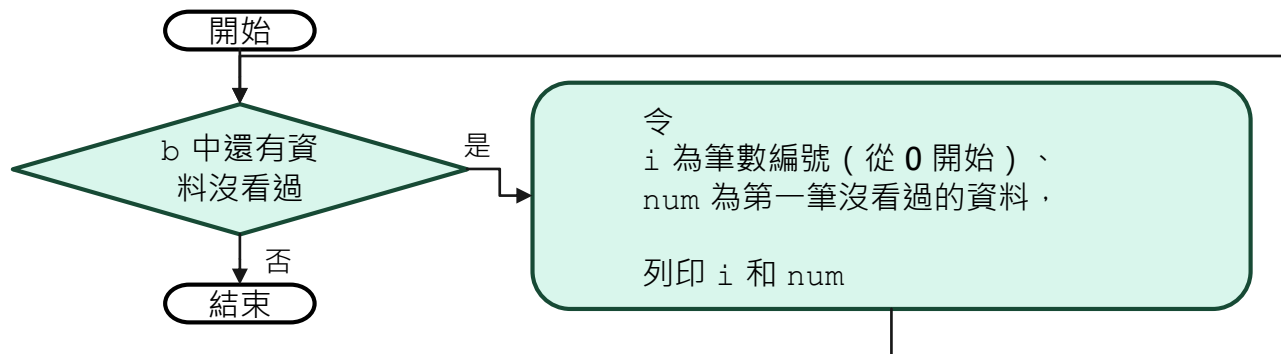
- ▶ 現在我們知道第 **01-02** 行程式的意思了，這兩行會分別讀取指定的文字檔案，將其內容讀取逐行切分，然後以變數 `win` 和 `mine` 代表。
 - 我們可以試著 `print(win, mine)` 來看看這兩個變數代表的資料內容。
- ▶ 接著，我們將以第 **04-06** 行程式比對 `win` 和 `mine` 中的資料是否相同。
- ▶ 第 **04** 行我們使用一個 `for` 句型，其主要目的是依序取出一堆資料中的各個資料。（`mine` 當中現在是從 '*我的發票.txt*' 檔案中讀取到的一堆發票號碼。）
- ▶ 第 **05** 行我們使用 `if` 句型以及 `in` 運算，判斷某個字串 `num` 是否存在 `win` 之中。如果存在，我們在第 **06** 行會列印中獎的訊息。

程式概念：重覆控制 (迴圈)

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

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

- ▶ 程式當中每種資料能作的運算和操作不同，例如之前我們說過字串可以進行 `split()`。
- ▶ 列表 (`list`) 是一個將多筆資料包裝起來的資料型態，在程式中我們以一對中括號來表示。上方程式碼第 01 行的變數 `a` 代表單一個字串 '123'，而第 02 行的 `b` 則代表一個列表，內含三個字串。
- ▶ 第 03 行的句型較複雜，涉及到 `for ... in ...` 句型以及 `enumerate()` 函式。此外我們把焦點放在 `i`、`num` 和 `b` 就好。這一整句的作用在於從代表一堆資料的 `b` 逐一取出單個資料，並令 `num` 代表之。同時，也以變數 `i` 來計數，從 0 開始，依序每次加 1。



程式概念：重覆控制 (迴圈)

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

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

- ▶ 程式當中每種資料能作的運算和操作不同，例如之前我們說過字串可以進行 `split()`。
- ▶ 列表 (`list`) 是一個將多筆資料包裝起來的資料型態，在程式中我們以一對中括號來表示。上方程式碼第 **01** 行的變數 `a` 代表單一個字串 `'123'`，而第 **02** 行的 `b` 則代表一個列表，內含三個字串。
- ▶ 第 **03** 行的句型較複雜，涉及到 `for ... in ...` 句型以及 `enumerate()` 函式。此外我們把焦點放在 `i`、`num` 和 `b` 就好。這一整句的作用在於從代表一堆資料的 `b` 逐一取出單個資料，並令 `num` 代表之。同時，也以變數 `i` 來計數，從 `0` 開始，依序每次加 `1`。
- ▶ 我們在第 **04** 行以 `print()` 列印出變數 `i` 和 `num` 的內容。從結果比對第 **02-03** 行的程式碼，應該可以掌握到第 **03** 行的用途和效果。由於 `b` 中有三個字串，因此 `num` 會在第 **03** 行依序代表其中一個字串。而 `i` 則從 `0` 開始，依序代表 `0`、`1`、`2`。
 - 注意第 **04** 行的程式碼必須先打至少一個空白（通常是四個空白，稱之為程式碼內縮），這是因為這一行程式碼是依附著 `for ... in ...` 句型的。

程式概念：選擇控制 (分支)

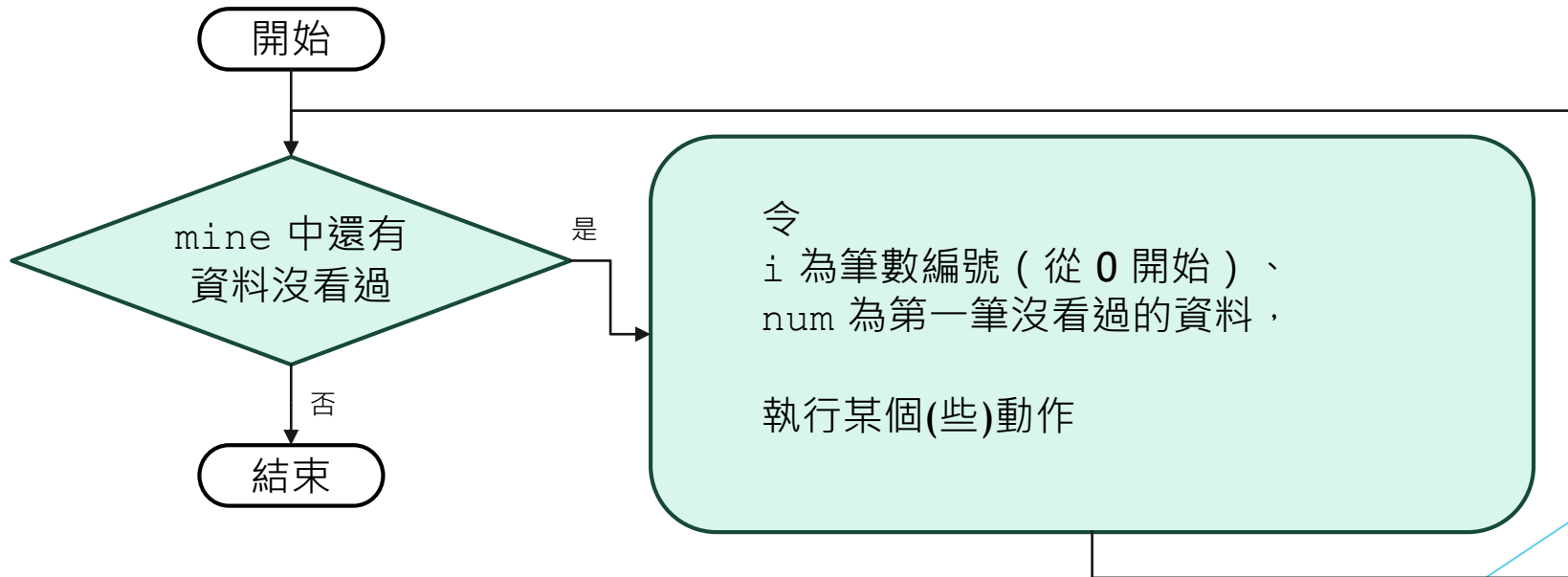
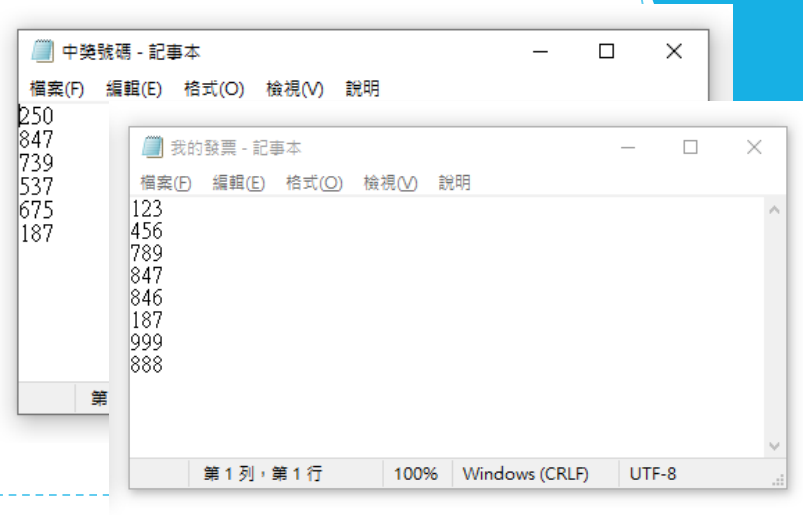
```
01 a = 3
02 b = [7, 4, 3, 6, 25]
03 if a > 2:
04     print('a 比 2 大')
05 if a in b:
06     print('b 中有 a')
```

```
a 比 2 大
b 中有 a
```

- ▶ 上方程式碼第 **01-02** 行定義 `a` 代表 **3**，`b` 代表包含了五個整數值的列表。
- ▶ 程式碼的基本執行流程是由上往下，一行一行執行。當我們需要控制某段程式碼只有在特定條件下才執行時，我們便可使用 `if` 句型。
- ▶ 第 **03-04** 行的意思是如果變數 `a` 的值大於 **2**，便執行 `print('a 比 2 大')`。
 - 注意：此處 `print('a 比 2 大')` 依附著 `if`，所以要往內縮。
- ▶ 第 **05-06** 行的意思是如果變數 `a` 的值有在 (`in`) 列表 `b` 當中的話，便執行 `print('b 中有 a')`。
- ▶ 本例中兩個 `if` 句子的條件都成立 (都是真的)，因此會執行第 **04** 和 **06** 行。

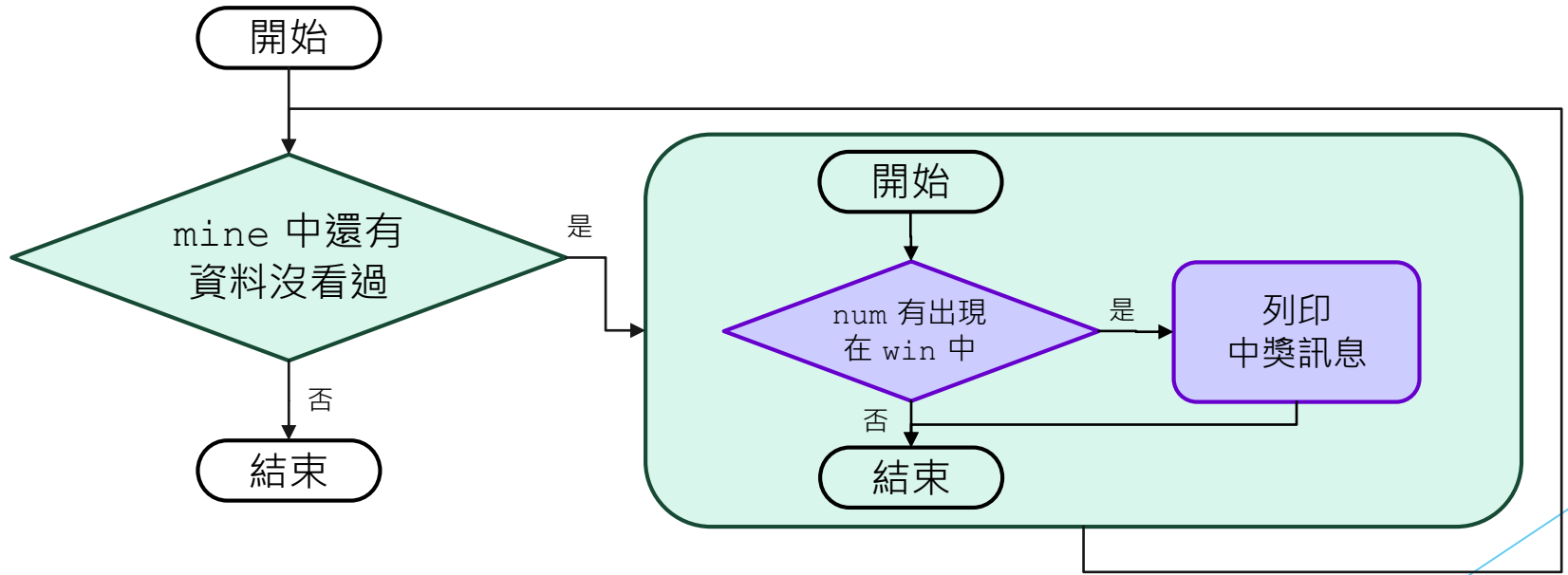
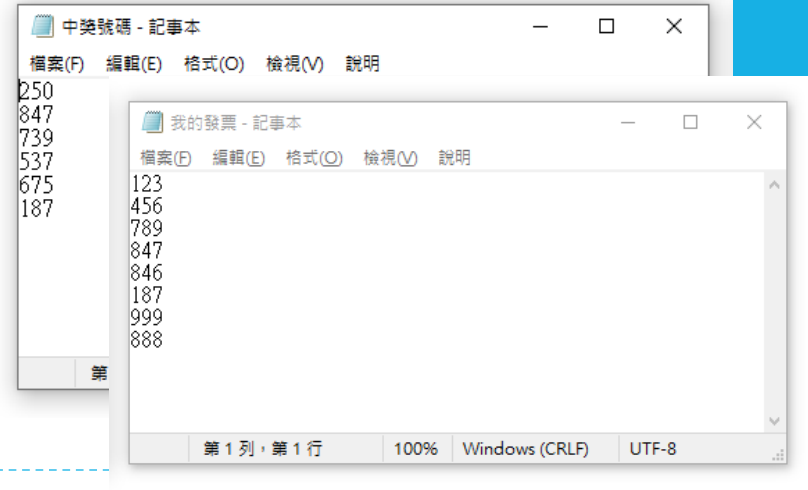
程式碼解說

```
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     if num in win:  
06         print('第', i+1, '張發票對中號碼', num, '!!')
```



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     if num in win:  
06         print('第', i+1, '張發票對中號碼', num, '!!')
```



發票對號

▶ 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):  
    if num in win:  
        print('第', i+1, '張發票對中號碼', num, '!!')
```

程式輸出

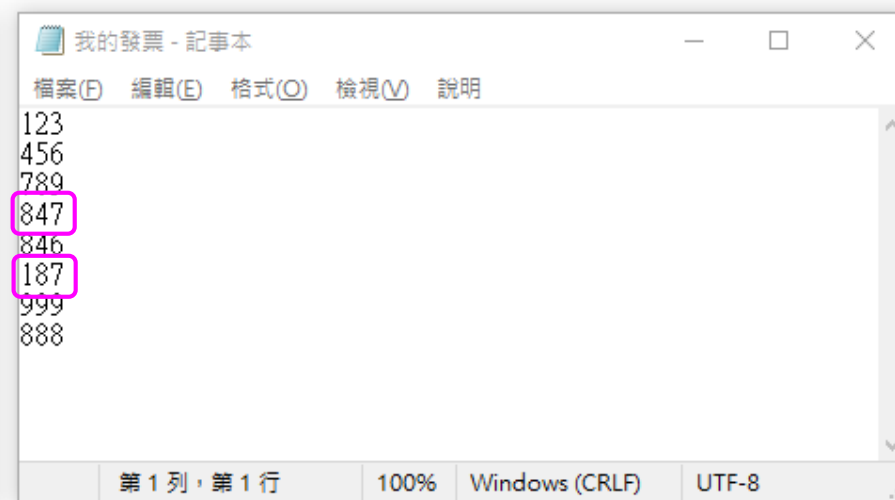
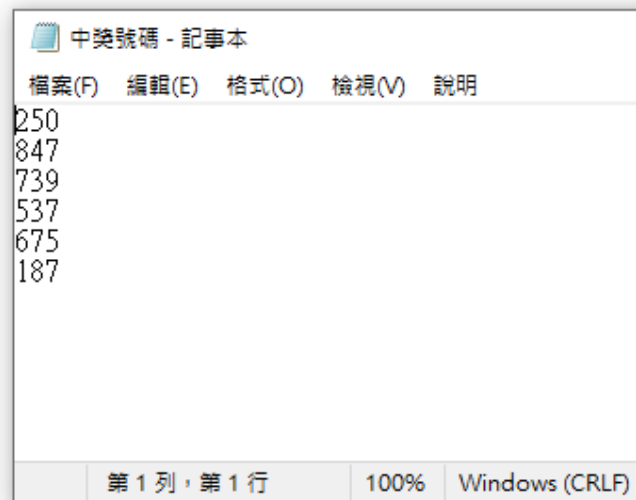
第 4 張發票對中號碼 847!
第 6 張發票對中號碼 187!



中獎號碼



我的發票



挑戰時刻

```
01 sum = 0
02 data = [7, 4, 3, 6, 25]
03 for i, num in enumerate(data):
04     sum = sum + num
05 print(sum)
```

45

```
01 cnt = 0
02 data = [7, 4, 3, 6, 25]
03 for i, num in enumerate(data):
04     if num > 5:
05         cnt = cnt + 1
06 print(cnt)
```

3

- ▶ 上面兩段程式分別展示如何計算一組資料的總和以及其中有幾個數大於 5，請閱讀上述程式並確認了解其意義。
- ▶ 請修改今天的發票對號程式，使得它能在最後輸出總共有幾張發票中獎以及中獎的總金額。