



Python程式語言

part 1

劉和師

2018/12/08

1. Python簡介

- ▶ Python 是一種泛用性的動態物件導向程式語言。
- ▶ Python 是一種直譯式語言。
- ▶ 它可以在命令列模式下操作，也可以寫成Script檔。
- ▶ 附檔名為 .py。
- ▶ Python 可以執行在 Windows、Mac OS X、Linux 等常見的作業系統平台和其它較少使用的作業系統上，也可以在 Java 和 .Net 環境中執行。
- ▶ 透過 Jython，Python 能與 Java 合作愉快；
透過 IronPython，Python 能與 .Net 合作愉快。

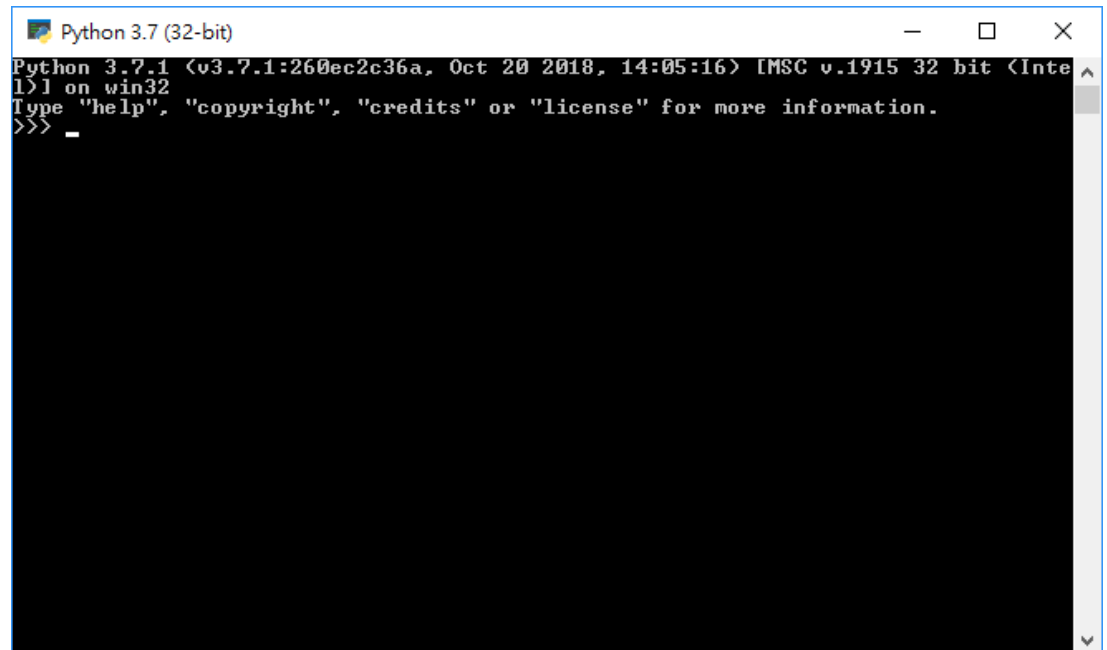


1. Python簡介

- ▶ Python官網 <https://www.python.org/>
- ▶ 目前是3.7.1版(2018年10月)，下載安裝後可於命令列模式下執行，提示字元為 >>>
- ▶ 若須測試簡單的指令或執行已寫好的程式，可用此模式



圖示

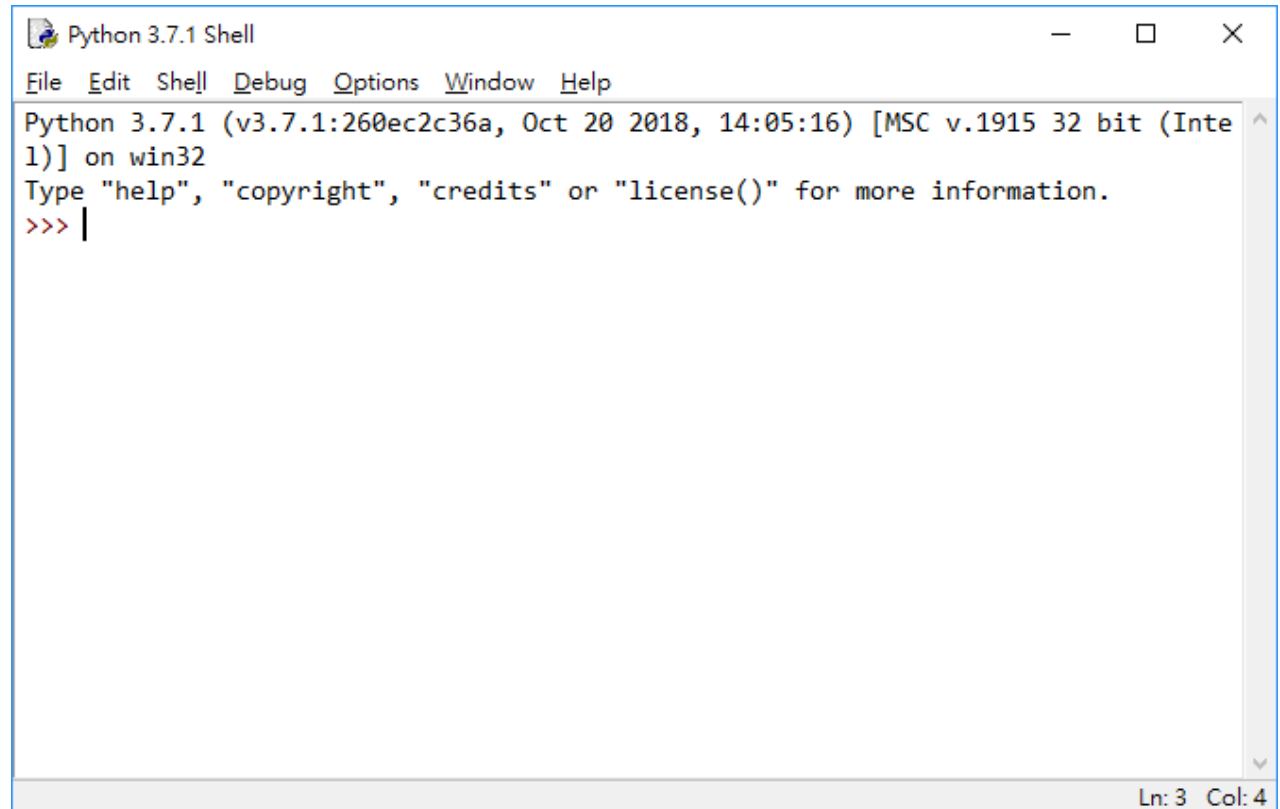
A screenshot of a Windows command prompt window titled "Python 3.7 (32-bit)". The window has a black background and white text. The text inside the window reads: "Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32", "Type 'help', 'copyright', 'credits' or 'license' for more information.", and ">>> _". The cursor is positioned after the underscore.

1. Python簡介

- ▶ 安裝後也有一個執行環境叫IDLE，不用再找IDE了，之後我們都用這個環境來練習。



圖示



1. Python簡介

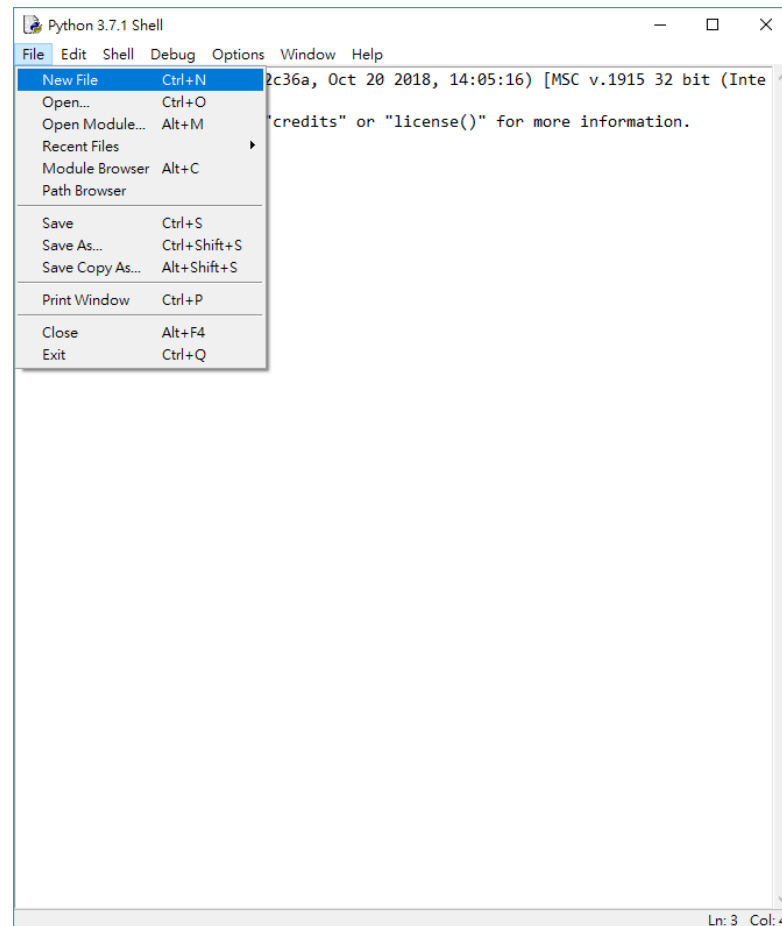
- ▶ 可以開啟Script編寫視窗來發展程式。



A screenshot of a Python script editor window. The title bar shows the file path: 9-1.py - D:\MyDocs\MyProgram\專門為中學生寫的程式語言設計\Python\ch9\9-1.py (3.7....). The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code is as follows:

```
def sum(N):  
    if N == 1:  
        value = 1  
    else:  
        value = N + sum(N-1)  
    return value  
  
#main  
x = int(input("計算1到N的總和，請輸入N:"))  
y = sum(x)  
print("1到"+str(x)+"的總和為"+str(y))
```

The status bar at the bottom right indicates "Ln: 1 Col: 0".



A screenshot of the Python 3.7.1 Shell window. The title bar shows "Python 3.7.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The File menu is open, showing the following options:

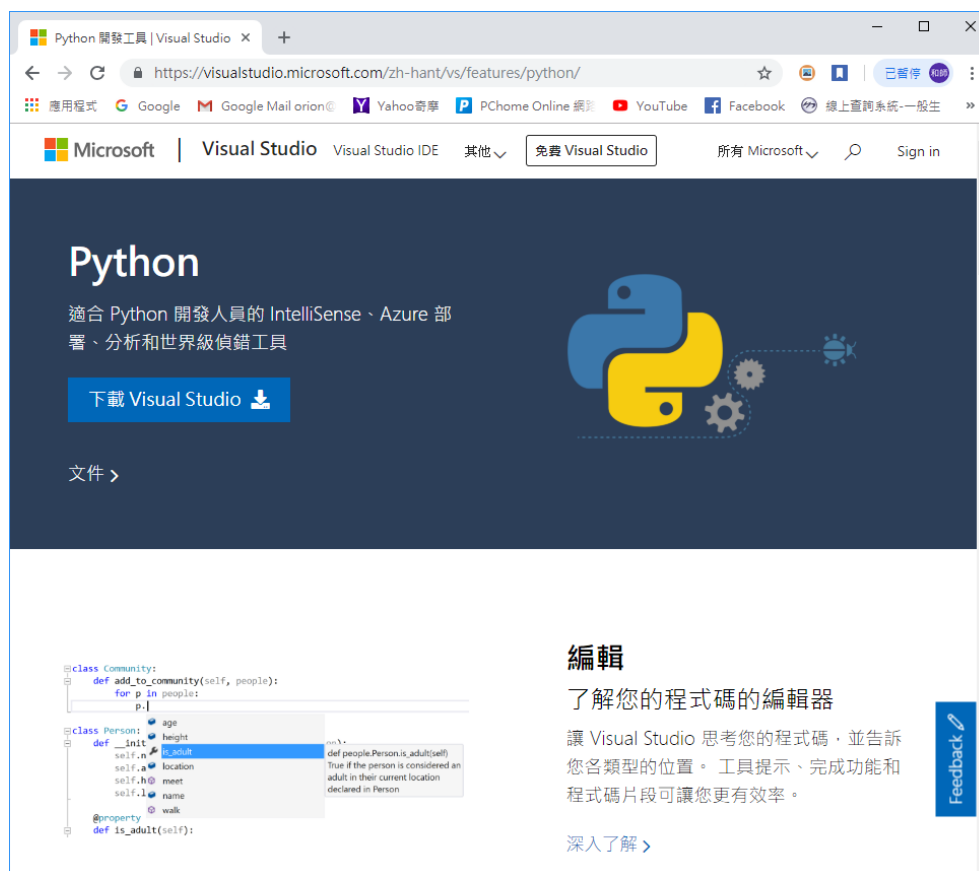
- New File (Ctrl+N)
- Open... (Ctrl+O)
- Open Module... (Alt+M)
- Recent Files
- Module Browser (Alt+C)
- Path Browser
- Save (Ctrl+S)
- Save As... (Ctrl+Shift+S)
- Save Copy As... (Alt+Shift+S)
- Print Window (Ctrl+P)
- Close (Alt+F4)
- Exit (Ctrl+Q)

The status bar at the bottom right indicates "Ln: 3 Col: 4".

1. Python簡介

► MS Visual Studio環境也支援Python

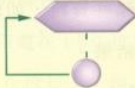
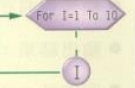
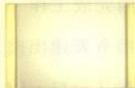


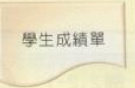

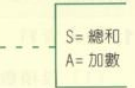
<https://visualstudio.microsoft.com/zh-hant/vs/features/python/>



2. 流程圖(Flow Chart)

- ▶ 解決問題的流程很重要，要先想好：
 - ▶ 1.現在是要解決甚麼問題？
 - ▶ 2.需要輸入什麼？
 - ▶ 3.要獲得什麼輸出？
 - ▶ 4.把解決過程畫(寫)出來，依步驟寫出程式

- ▶ 流程圖是用簡單的圖形來代表執行的步驟，協助我們思考並建立解決步驟

名稱	符號	意義	範例
7.迴圈符號		表示程式迴圈控制變數初值及終值的假設	
8.副程式符號		表示一群程式步驟或流程，用以說明副程式或其他流程的組合	
9.報表符號		表示以列表機印出報表文件	
10.註解符號		表示對某一流程加以註解	

2. 流程圖(Flow Chart)

- ▶ 流程圖有許多符號，代表各種意義。
- ▶ 在寫程式前應該都要先畫出流程圖。

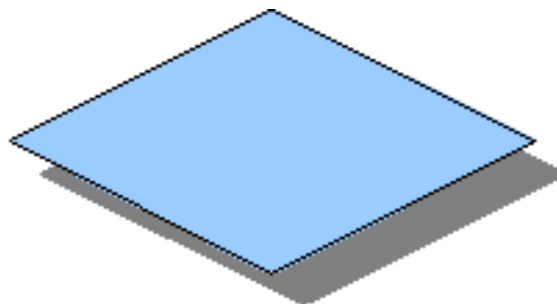
名稱	符號	意義	範例
1. 起止符號		表示程式的開始或結束	 
2. 流程符號		表示流程進行的方向	
3. 輸入/輸出符號		表示資料之輸入或結果的輸出	 
4. 處理符號		表示執行或處理某些工作	 
5. 決策判斷符號		表示對某一個條件做判斷	
6. 連接符號		用於： 1. 轉接到另一頁 2. 避免流線交叉 3. 避免流線太長	

2. 流程圖(Flow Chart)

- ▶ 對於簡單的問題，其實記住這四個也就夠了



端點



決策



程序步驟



輸出輸入

2. 流程圖(Flow Chart)

- ▶ 流程圖是你解決問題的想法及過程，沒有標準答案，只要能解決問題就可以
- ▶ 不過**效率**就有關係了，練習讓你的思考更清晰，步驟更簡潔



參考教材

- ▶ 我們以李家同教授編寫的「專門為中學生寫的程式語言設計」一書為主，這是一本很簡單的書。
- ▶ 程式依書中範例為主，習題解答為參考，請要自己親自做出來

聯經出版，2018年8月



加油~~



Ch1. 簡單的程式

▶ 整數加法：

假設我們有兩個整數 x 和 y ，要求輸出 $x+y$ 的結果。

▶ 流程圖：

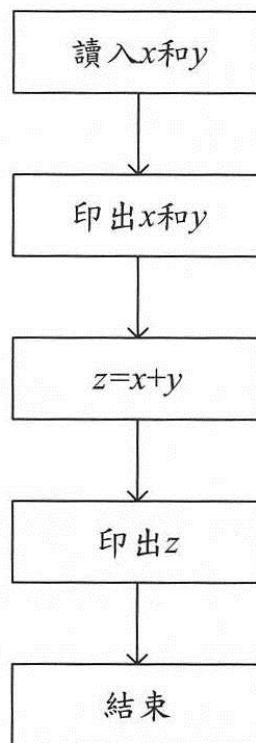


圖 1.1 ▶ 例題 1.1 的流程圖

Ch1. 簡單的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x value:",x)
print("y value:",y)

z = x + y

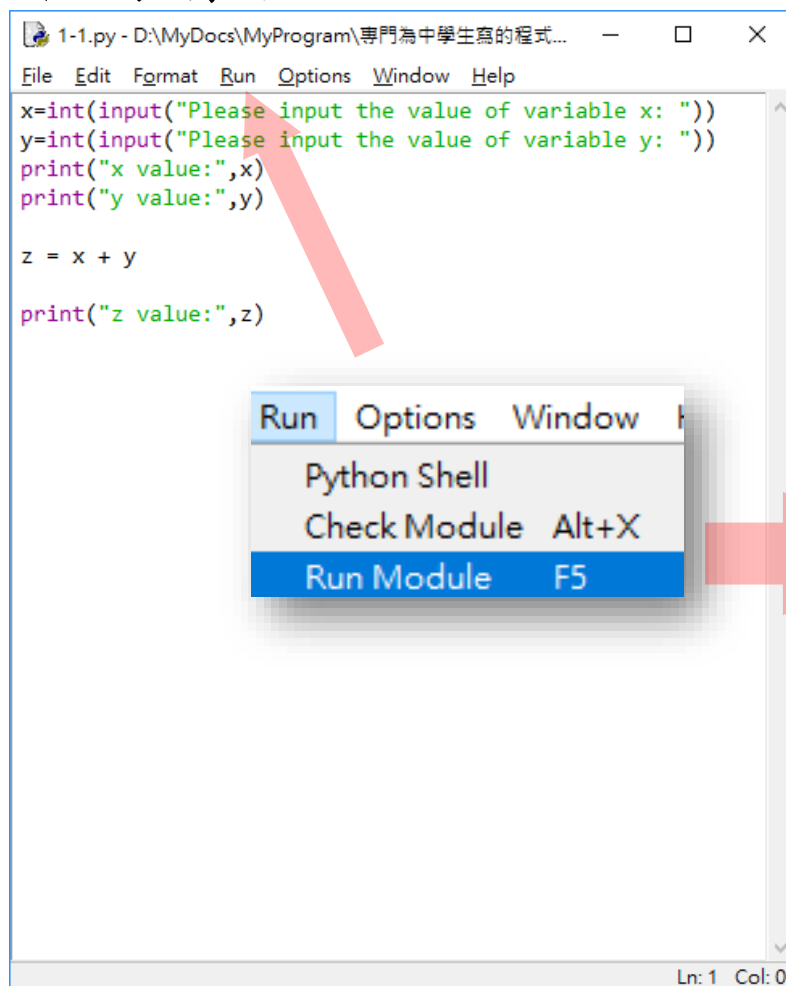
print("z value:",z)
```

▶ 說明：

- ▶ 1.深紅色的部分稱“字串”，會忠實地顯示在結果畫面上
- ▶ 2. x、y、z稱“變數”，會以它們的值呈現
- ▶ 3.input()是請求鍵盤輸入，print()是印出資訊
- ▶ 4.int()是將輸入的字串轉成整數(鍵盤輸入的都視為“字串“)

Ch1. 簡單的程式

▶ 執行情形：



The screenshot shows a Python IDE window titled "1-1.py - D:\MyDocs\MyProgram\專門為中學生寫的程式...". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x value:",x)
print("y value:",y)

z = x + y

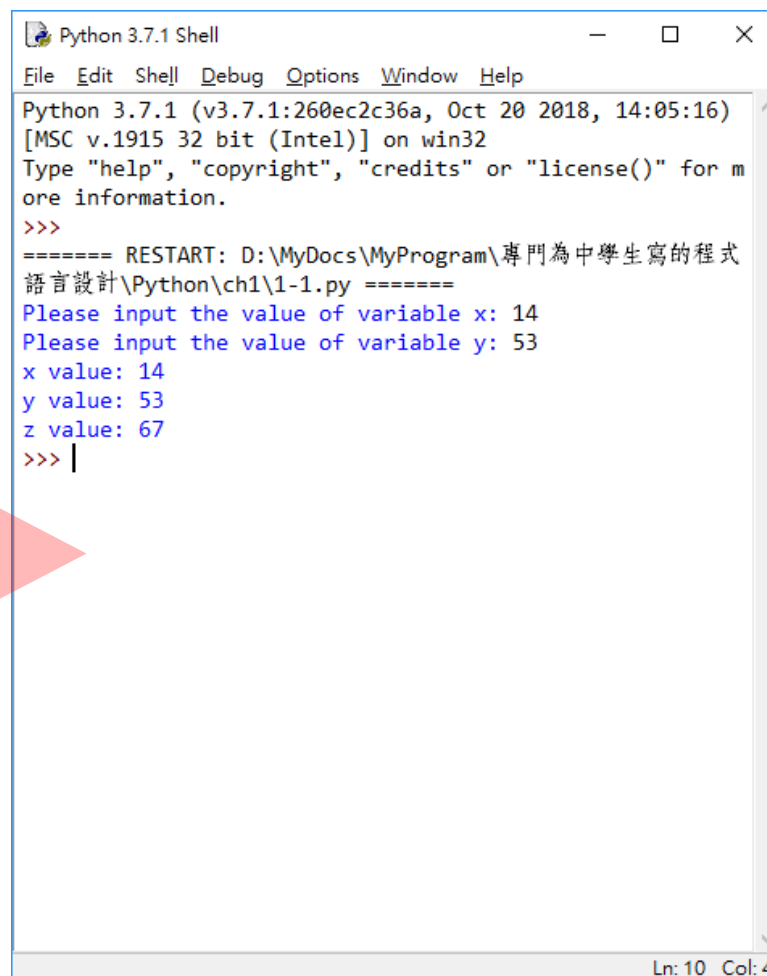
print("z value:",z)
```

The "Run" menu is open, showing the following options:

- Run
- Options
- Window
- Python Shell
- Check Module Alt+X
- Run Module F5

A red arrow points from the "Run" menu to the Python Shell window on the right.

Ln: 1 Col: 0



The screenshot shows the "Python 3.7.1 Shell" window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The output of the script is as follows:

```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16)
[MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: D:\MyDocs\MyProgram\專門為中學生寫的程式
語言設計\Python\ch1\1-1.py =====
Please input the value of variable x: 14
Please input the value of variable y: 53
x value: 14
y value: 53
z value: 67
>>> |
```

Ln: 10 Col: 4

Ch1. 簡單的程式

▶ 計算 $(x+y)/2$ ：

假設我們有兩個整數 x 和 y ，要求輸出 $(x+y)/2$ 的結果。

▶ 流程圖：

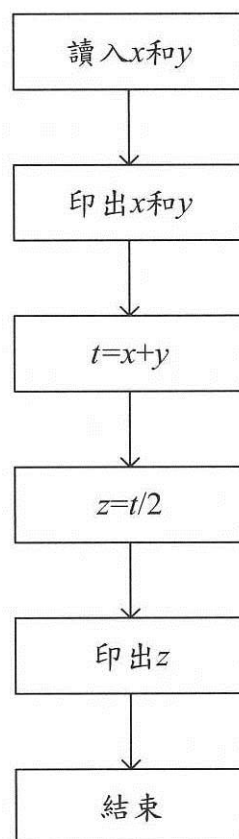


圖 1.2 ▶ 例題 1.2 的流程圖

Ch1. 簡單的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x value:",x)
print("y value:",y)

t = x + y
z = t / 2

print("z value:",z)
```

▶ 說明：

- ▶ 執行時空白行會予以忽略，空行是為了程式美觀或易於閱讀。

Ch1. 簡單的程式

▶ 計算 $(x+y)*(u+v)$ ：

假設我們有四個整數 x, y, u 和 v ，
要求 $(x+y)*(u+v)$

▶ 流程圖：

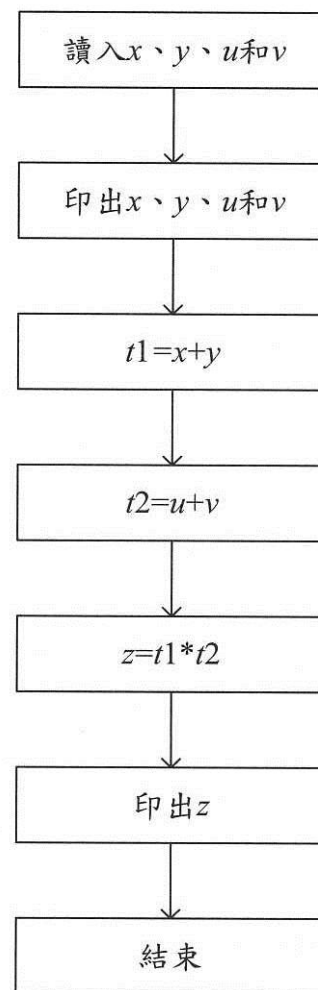


圖 1.3 ▶ 例題 1.3 的流程圖

Ch1. 簡單的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
u=int(input("Please input the value of variable u: "))
v=int(input("Please input the value of variable v: "))
print("x value:", x)
print("y value:", y)
print("u value:", u)
print("v value:", v)

t1 = x + y
t2 = u + v
z = t1 * t2
print("z value:", z)
```

Ch1. 簡單的程式

- ▶ 浮點數字加法：
- ▶ 用float()取代int()即可

```
x=float(input("Please input the value of variable x: "))
y=float(input("Please input the value of variable y: "))
print("x value:",x)
print("y value:",y)

z = x + y

print("z value:",z)
```

- ▶ 另外input()裡面也可以是空的，只是沒有提示訊息容易造成使用者困擾，程式應盡量將訊息表達清楚

Ch1. 簡單的程式

- ▶ 另外input()裡面也可以是空的，只是沒有提示訊息容易造成使用者困擾，程式應盡量將訊息表達清楚
- ▶ 例如：

```
1-5.py - D:\MyDocs\MyProgra...  
File Edit Format Run Options Window Help  
x=int(input())  
y=int(input())  
  
z = x + y  
  
print(z)  
|
```

```
*Python 3.7.1 Shell*  
File Edit Shell Debug Options Window Help  
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:\MyDocs\MyProgram\專門為中學生寫的程式語言設計\Python\ch1\1-5.py =====
```

程式已經在執行了，卻沒有任何提示要求輸入

Ln: 5 Col: 0

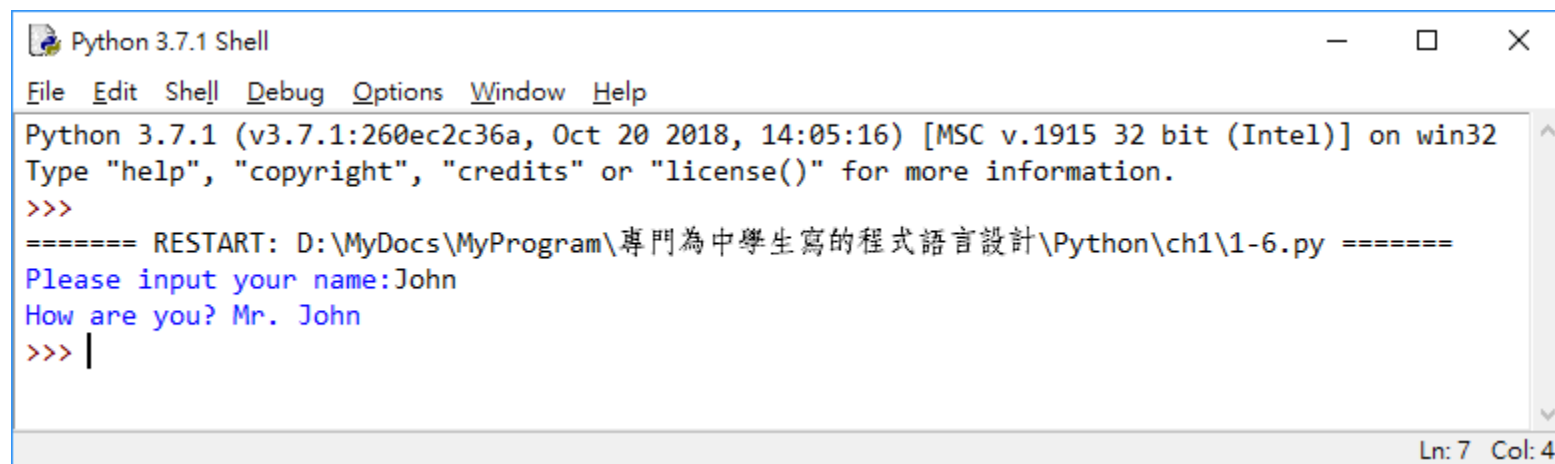
Ch1.簡單的程式

▶ 輸入文字：

如果要輸入文字，簡單的用input()即可

```
A = input("Please input your name:")  
print("How are you? Mr.", A)
```

▶ 執行結果：



```
Python 3.7.1 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:\MyDocs\MyProgram\專門為中學生寫的程式語言設計\Python\ch1\1-6.py =====  
Please input your name:John  
How are you? Mr. John  
>>> |
```

Ln: 7 Col: 4

Ch1. 簡單的程式

▶ 計算二次方 $a^2 + b^2$:

```
a = int(input("請輸入 a = "))  
b = int(input("請輸入 b = "))  
c = pow(a,2) + pow(b,2)  
print("c = ",c)
```

或

```
a = int(input("請輸入 a = "))  
b = int(input("請輸入 b = "))  
c = a**2 + b**2  
print("c = ",c)
```

▶ 以上結果是相同的

- ▶ 計算次方可以用函數pow()或運算符號 **
- ▶ 例如計算 x^y ，可寫成pow(x,y)或 $x**y$

Ch1.簡單的程式

▶ 小於1的次方：

計算 x^y ，但 y 為介於0到1之間的浮點數

```
*1-9.py - D:\MyDocs...
File Edit Format Run Options Window
Help
x = int(input("請輸入 x = "))
y = float(input("請輸入 y = "))
z = x**y
print("z = ",z)
Ln: 4 Col: 0
```

```
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\MyDocs\MyProgram\專門為中學生寫的程式語言設計\Python\ch1\1-9.py =====
請輸入 x = 16
請輸入 y = 0.5
z = 4.0
>>> |
Ln: 8 Col: 4
```

輸入浮點數

Ch1. 簡單的程式

- ▶ 計算 $\log(x)$ ：

```
import math
x = int(input("請輸入 x = "))
y = math.log(x)
print("y = ", y)
```

- ▶ 在這裡 $\log(x)$ 是以 $e = 2.718$ 為底
- ▶ 某些特別的函數需要另外引入，我們用import指令，這裡是引入數學函數，所以用：**import math**，然後用**math.log()**函數

math 的 log()

Ch1. 簡單的程式

▶ 計算平方根 $y = \sqrt{x}$:

- ▶ 1. 以import方式，使用sqrt()函數開根號

```
import math
x = int(input("請輸入 x = "))
y = math.sqrt(x)
print("y = ", y)
```

- ▶ 2. 不使用引入，直接用次方運算符號

```
x = int(input("請輸入 x = "))
y = x**0.5
print("y = ", y)
```

- ▶ 說明：開根號是 $\frac{1}{2}$ 次方 = 0.5次方



Ch1. 簡單的程式

- ▶ 計算 $y = \sin(x)$:

```
import math
x = float(input("請輸入 x = "))
y = math.sin(x * math.pi)
print("y = ", y)
```

- ▶ `math.sin()`、`math.pi` 都是 `math` 提供的函數
- ▶ Python 的 π 精確到小數點後 16 位



Ch1. 習題

- ▶ 1-1. 寫一程式，輸入 a, b, c, d ，計算 $\frac{(a+b)}{(c-d)} \times d$ 。先畫出流程圖。

- ▶ 1-2. 假設有一組二元一次方程式如下：

$$a_1x + b_1y = c_1, \quad a_2x + b_2y = c_2$$

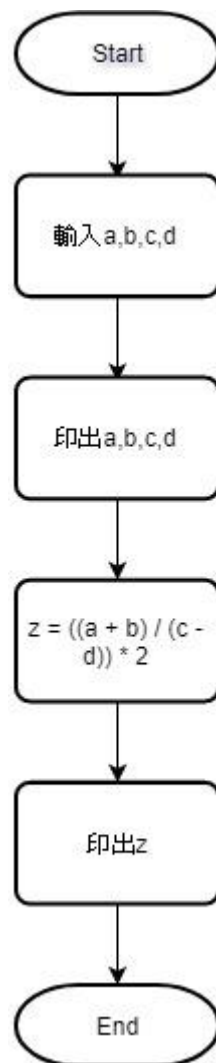
此組方程式的解如下：

$$x = \frac{(c_1b_2 - c_2b_1)}{(a_1b_2 - a_2b_1)}, \quad y = \frac{(c_1a_2 - c_2a_1)}{(b_1a_2 - a_1b_2)}$$

寫一程式，輸入此方程式變數之係數。計算此組方程式的解先畫出流程圖。

Ch1. 習題

► 1-1 流程圖：



Ch1. 習題

▶ 1-1 參考程式：

```
a=float(input("Please input the value of variable a: "))
b=float(input("Please input the value of variable b: "))
c=float(input("Please input the value of variable c: "))
d=float(input("Please input the value of variable d: "))

print("a value:",a)
print("b value:",b)
print("c value:",c)
print("d value:",d)

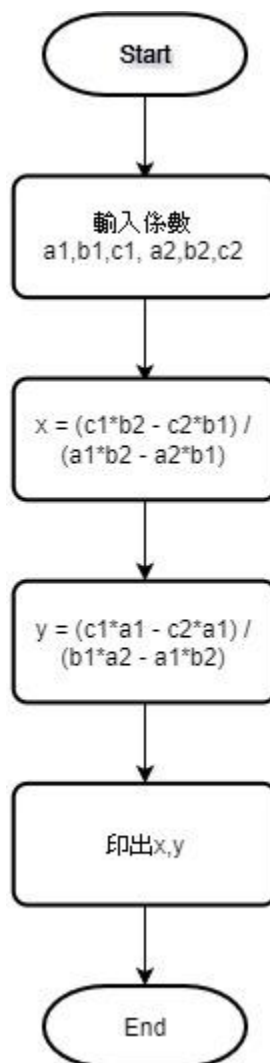
z = ((a + b) / (c - d)) * 2

print("((a + b) / (c - d)) * 2 = ",z)
```



Ch1. 習題

► 1-2 流程圖：



Ch1. 習題

▶ 1-2 參考程式：

```
a1 = float(input("請輸入係數 a1:"))
b1 = float(input("請輸入係數 b1:"))
c1 = float(input("請輸入係數 c1:"))
a2 = float(input("請輸入係數 a2:"))
b2 = float(input("請輸入係數 b2:"))
c2 = float(input("請輸入係數 c2:"))

x = (c1*b2 - c2*b1) / (a1*b2 - a2*b1)
y = (c1*a1 - c2*a1) / (b1*a2 - a1*b2)

print("x = ",x)
print("y = ",y)
```


Ch1. 習題

- ▶ 1-3. 輸入a和b，求：

$$c = \frac{a^1 + b^2}{a^2 - b^2}$$

- ▶ 1-4. 輸入a和b，求：

$$c = \sqrt{a^2 + b^2}$$

- ▶ 1-5. 輸入a、b和c，求：

$$c = a - (b + c)(3a - c)$$



Ch1. 習題

▶ 1-3 參考程式：

```
a = float(input("請輸入 a:"))  
b = float(input("請輸入 b:"))  
  
c = (a**2 + b**2) / (a**2 - b**2)  
  
print("c = ",c)
```



Ch1. 習題

▶ 1-4 參考程式：

```
import math
a = float(input("請輸入 a:"))
b = float(input("請輸入 b:"))

c = math.sqrt(a**2 + b**2)

print("c = ",c)
```

或

```
a = float(input("請輸入 a:"))
b = float(input("請輸入 b:"))

c = (a**2 + b**2)**0.5

print("c = ",c)
```

Ch1. 習題

▶ 1-5 參考程式：

```
a = float(input("請輸入 a:"))  
b = float(input("請輸入 b:"))  
c = float(input("請輸入 c:"))  
  
c = a - (b + c) * (3 * a - c)  
  
print("c = ", c)
```

補充說明：變數、運算式和敘述

▶ Python的主要變數型態有：

- ▶ number，數值
- ▶ string，字串
- ▶ list，清單(陣列)
- ▶ tuple (set of (so many) elements -- usually used of sets with ordered elements)
- ▶ dictionary，字典



補充說明：變數、運算式和敘述

▶ number 包含：

▶ 浮點數：

▶ 3.1415, 6.02E23 (64位元，倍精度)

▶ 整數：

▶ 100 (32 位元), 0324(八進位數), 0xA140 (十六進位數)

▶ 長整數：

▶ 50000000000 (任意精度)

▶ 複數：

▶ 1-5j (兩個浮點數)



補充說明：變數、運算式和敘述

▶ string :

▶ 單引號

- ▶ 'this is a "single-quoted" string,\n don\'t you like it?'

▶ 雙引號

- ▶ "this is a 'double-quoted' string"

▶ 三引號(三個單引號)

- ▶ '''this is a triply-quoted string

字串中可以換行,

也可以用'單引號'和"雙引號"'''

補充說明：變數、運算式和敘述

▶ list :

- ▶ 和陣列類似，由中括號加逗號組成

Ex: $m = [1, 5, 7, -3, -1, 0]$

- ▶ 清單中有清單：

Ex: $k = [3, m, \text{'hello'}]$

補充說明：變數、運算式和敘述

▶ tuple :

- ▶ 和清單相似, 是一些量的有序集合。和清單最大的不同是不能改變, 可以作為dictionary的鍵、函式回傳、多變數給值 ... 等
- ▶ 以小括號加逗號表示, 小括弧視情況可省略

Ex : `t = (-1, 3.5, 'hello')`

`x, y, z = 1, 2, 3`



補充說明：變數、運算式和敘述

▶ dictionary

- ▶ 鍵 (key) 和值 (value) 的對照表 --> 「關連式記憶」
- ▶ 鍵必需是不可變的型態 (如: 常數、常字串、Tuple)
- ▶ Ex: prices = { 'apple':7.5 , 'orange':4.5 , 'banana':2 }

price 包含

'apple'	7.5
'orange'	4.5
'banana'	2

補充說明：變數、運算式和敘述

- ▶ 變數直接使用，不需宣告：
 - ▶ counter = 100 整數
 - ▶ miles = 1000.0 浮點數
 - ▶ name = "Devid" 字串
- ▶ 多個變數一起指定：
 - ▶ a = b = c = 1
 - ▶ a,b,c = 1,2,"Devid"
- ▶ 變數指定後，變數型態就固定，重新指定新型態的值，等同重新宣告：
 - ▶ a = 1 a是整數
 - ▶ a = "Devid" a變成字串了
- ▶ 變數的刪除：
 - ▶ a = b = c = 1
 - ▶ del a 刪除了變數a

補充說明：變數、運算式和敘述

▶ 正負號：

符號	意義	範例
+	正號	+ a
-	負號	- a

▶ 算術運算子

運算子	功能	範例
+	加	a + b
-	減	a - b
*	乘	a * b
/	除	a / b
%	取餘數	a % b
**	指數	a ** b
//	整除(向下取整數)	a // b

補充說明：變數、運算式和敘述

▶ 邏輯運算子：

運算子	功能	範例
and	且	a and b
or	或	a or b
not	非	not a

▶ 關係運算子：

運算子	功能	範例
==	相等	a == b
!= 或 <>	不相等	a != b , a <> b
>	大於	a > b
>=	大於等於	a >= b
<	小於	a < b
<=	小於等於	a <= b

補充說明：變數、運算式和敘述

▶ 位元運算子：

運算子	功能	範例
&	And	a & b
	Or	a b
^	Xor	a ^ b
~	Not	~ a
<<	位元左移	a << 2
>>	位元右移	a >> 2

▶ 成員運算子：

運算子	功能	範例
In	如果在指定的序列中找到值返回 True ，否則返回 False 。	a in list
not in	如果在指定的序列中沒有找到值返回 True ，否則返回 False 。	a not in list

補充說明：變數、運算式和敘述

► 指派運算子：

運算子	功能	範例
=	指派	a = b
+=	相加同時指派	a += b
-=	相減同時指派	a -= b
*=	相乘同時指派	a *= b
/=	相除同時指派	a /= b
%=	取餘數同時指派	a %= b
**=	取指數同時指派	a **= b

補充說明：變數、運算式和敘述

運算子優先順序	說明
**	指數 (最高優先順序)
~ + -	按位翻轉, 一元加號和減號 (最後兩個的方法名為 +@ 和 -@)
* / % //	乘, 除, 取模和取整除
+ -	加法減法
>> <<	右移, 左移運算子
&	位 'AND'
^	位運算子
<= < > >=	比較運算子
<> == !=	等於運算子
= %= /= // = -= += *= **=	設定運算子
is is not	身份運算子
in not in	成員運算子
not and or	邏輯運算子

補充說明：變數、運算式和敘述

- ▶ + 與 * 也可用於字串 (string)，例如

b = "a" + "b" # 字串連接，b 會等於 "ab"

c = "a" * 2 # 字串重複，c 會等於 "aa"

- ▶ = 也可以用平行指派 (parallel assignment)，例如

a, b, c = 11, 22, 33

a 會等於 11

b 會等於 22

c 會等於 33

a = b = c = 1

a、b、c 都等於 1

- ▶ 適度的使用括符 () 區分優先順序

補充說明：變數、運算式和敘述

▶ 型別轉換：

- ▶ `str(x)` # 轉換成字串
- ▶ `int(x)` # 轉換成整數
- ▶ `float(x)` # 轉換成浮點數
- ▶ `complex(x)` # 轉換為複數



補充說明：變數、運算式和敘述

▶ 索引切片運算：

只要具有索引特性，基本上都能進行切片運算。如字串、list、tuple等。

▶ 例如：

name = 'Justin' 則

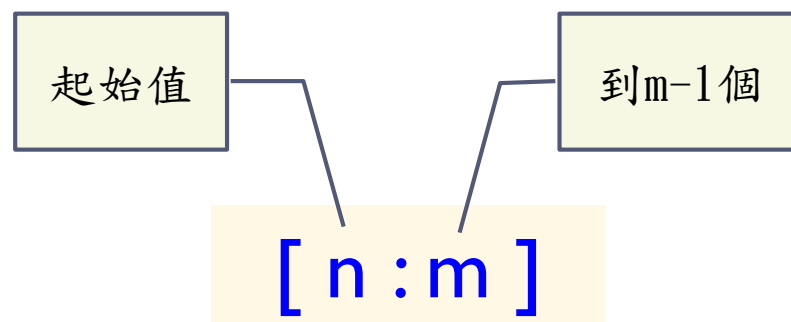
name[0:3] 得到 'Jus'

name[3:] 得到 'tin'

name[:4] 得到 'Just'

name[:-1] 得到 'Justi' (-1是倒數第一個)

name[-5:-1] 得到 'usti'



補充說明：變數、運算式和敘述

▶ 註解：

▶ 單行註解：用 # 開頭的部分至行尾

```
# This is a comment.....  
print('Hello') # This is a comment....
```

▶ 多行註解：連續三個單引號 ''' 框住的範圍

```
'''
```

```
這是
```

```
多行註解
```

```
'''
```

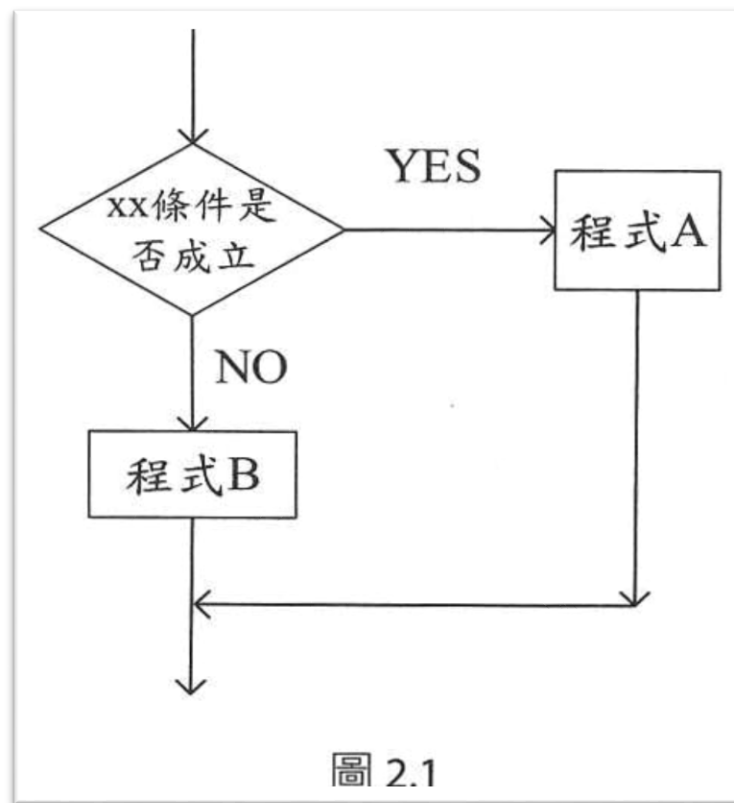
休息一下~



程式設計師的終極形態

Ch2. 有if指令的程式

- ▶ if是：「如果怎樣怎樣，你就該如何如何」
- ▶ 圖2.1的意義是：如果某某條件成立，執行程式A，否則執行程式B。



Ch2. 有if指令的程式

- ▶ 讀入兩個數 x 和 y ，選出大的那一個。
- ▶ 流程圖：

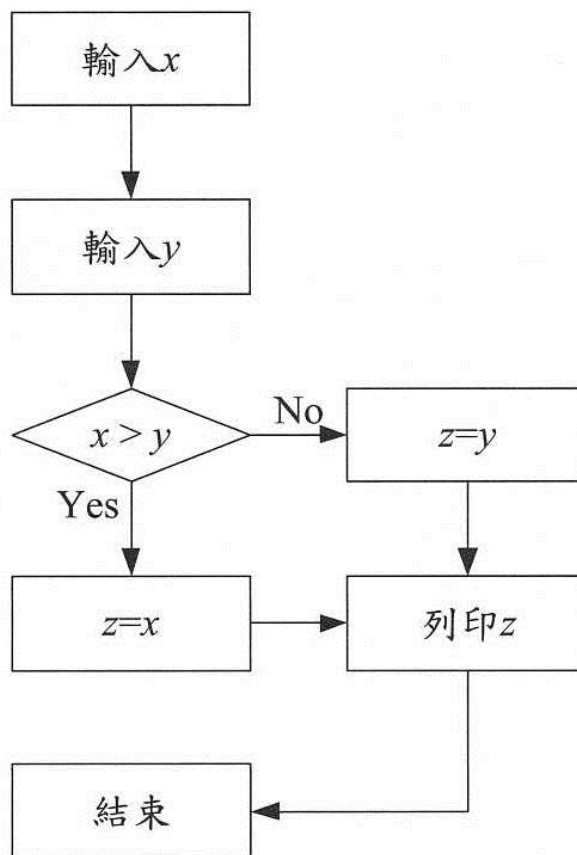


圖 2.2 ▶ 例題 2.1 的流程圖

Ch2. 有if指令的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x = ",x)
print("y = ",y)
```

```
if x > y:
    z = x
else:
    z = y
```

```
print("z = ",z)
```

注意冒號及縮排！
if 和 else要對齊

- ▶ 程式中縮排的空格數目必須相同，一般用4格，不可以和Tab混用，一般是用空白

Ch2. 有if指令的程式

- ▶ 讀入 x 和 y ，如果 $x > y$ ，則回傳 $x - y$ ，否則回傳 $y - x$ 。
- ▶ 流程圖：

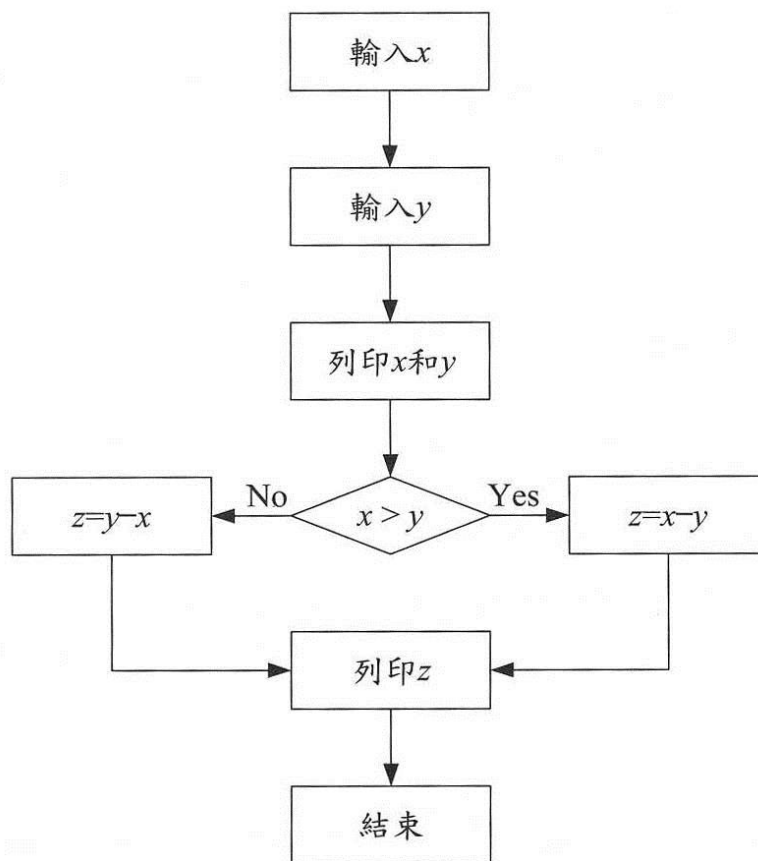


圖 2.3 ▶ 例題 2.2 的流程圖

Ch2. 有if指令的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x = ",x)
print("y = ",y)

if x > y:
    z = x - y
else:
    z = y - x

print("z = ",z)
```



Ch2. 有if指令的程式

- ▶ 讀入一數字，如果是負數，就將它轉成正數。
- ▶ 流程圖：

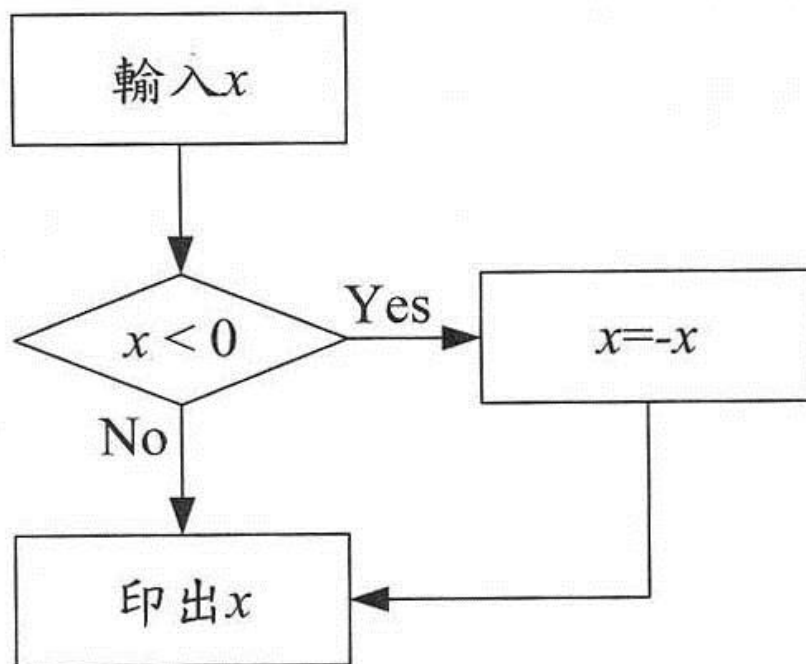


圖 2.4 ▶ 例題 2.3 的流程圖

Ch2. 有if指令的程式

▶ 參考程式：

```
x=int(input("Please input the value of variable x: "))  
  
if x < 0:  
    x = -x  
  
print("x = ",x)
```



Ch2. 有if指令的程式

- ▶ 輸入一個學生的分數，輸出該成績級別，分數級別如下：

90分以上 → A

80~89分 → B

70~79分 → C

60~69分 → D

60分以下 → F

- ▶ 流程圖：

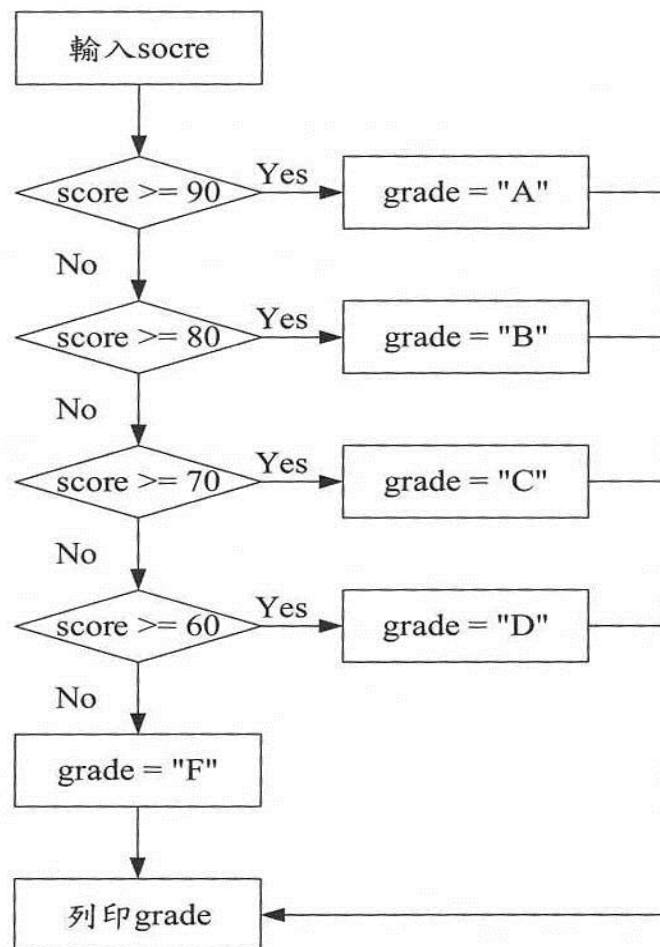


圖 2.5 ▶ 例題 2.4 的流程圖

Ch2.有if指令的程式

▶ 參考程式：

```
print("輸入學生成績，將會輸出對應的成績級別")
score = int(input("輸入成績(0~100): "))
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
else:
    grade = "F"
print("此成績級別為 ", grade)
```

Ch2. 有if指令的程式

- ▶ 某商店在大特價，根據購買的商品總金額決定打折的折扣，總金額越高折扣越多，折扣表如下：

未滿1000 → 無折扣

1000~4999 → 9折

5000~9999 → 8折

高於10000 → 7折

小數點四捨五入，印出折扣後的金額。



Ch2. 有if指令的程式

► 流程圖：

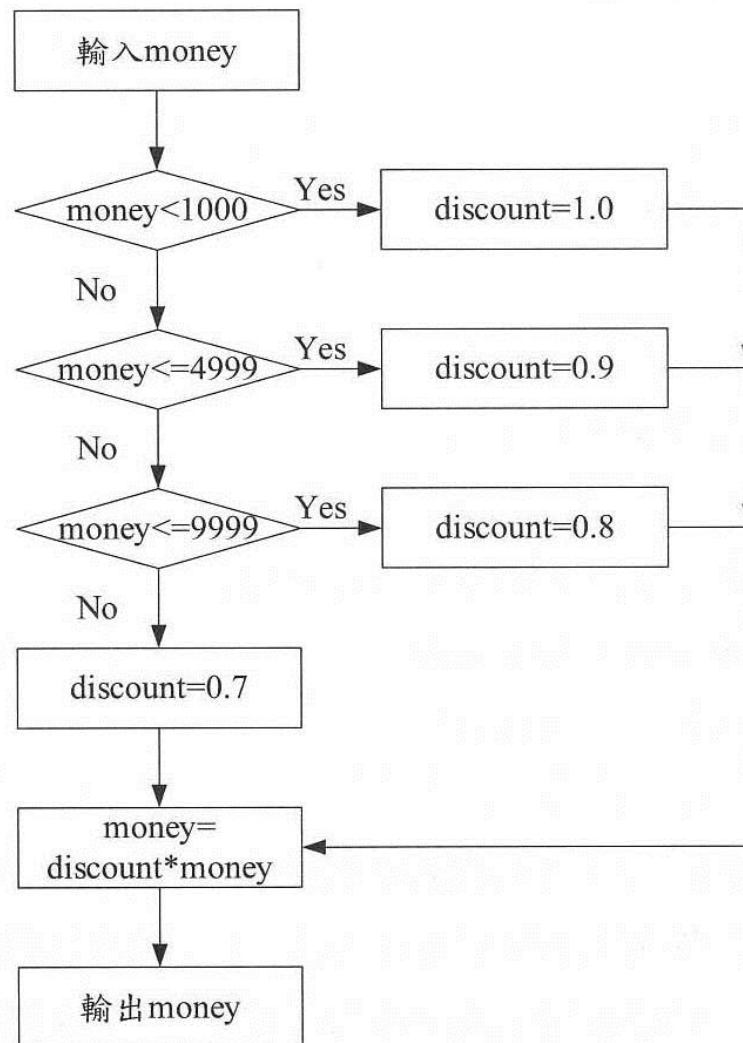


圖 2.6 ► 例題 2.5 的流程圖

Ch2.有if指令的程式

▶ 參考程式：

```
print("輸入總金額，將會依據金額打折，並輸出打折後的金額")
money = int(input("請輸入購買總金額： "))
if money < 1000:
    print("抱歉，沒有折扣唷!")
    discount = 1.0
elif money < 4999:
    print("享有9折優惠唷!")
    discount = 0.9
elif money < 9999:
    print("享有8折優惠唷!")
    discount = 0.8
else:
    print("享有7折優惠唷！")
    discount = 0.7
money = int(round(money * discount,0))
print("打折後總金額為： ",money)
```

函數**round(n,m)**
將**n**值在小數點第**m**位
四捨五入

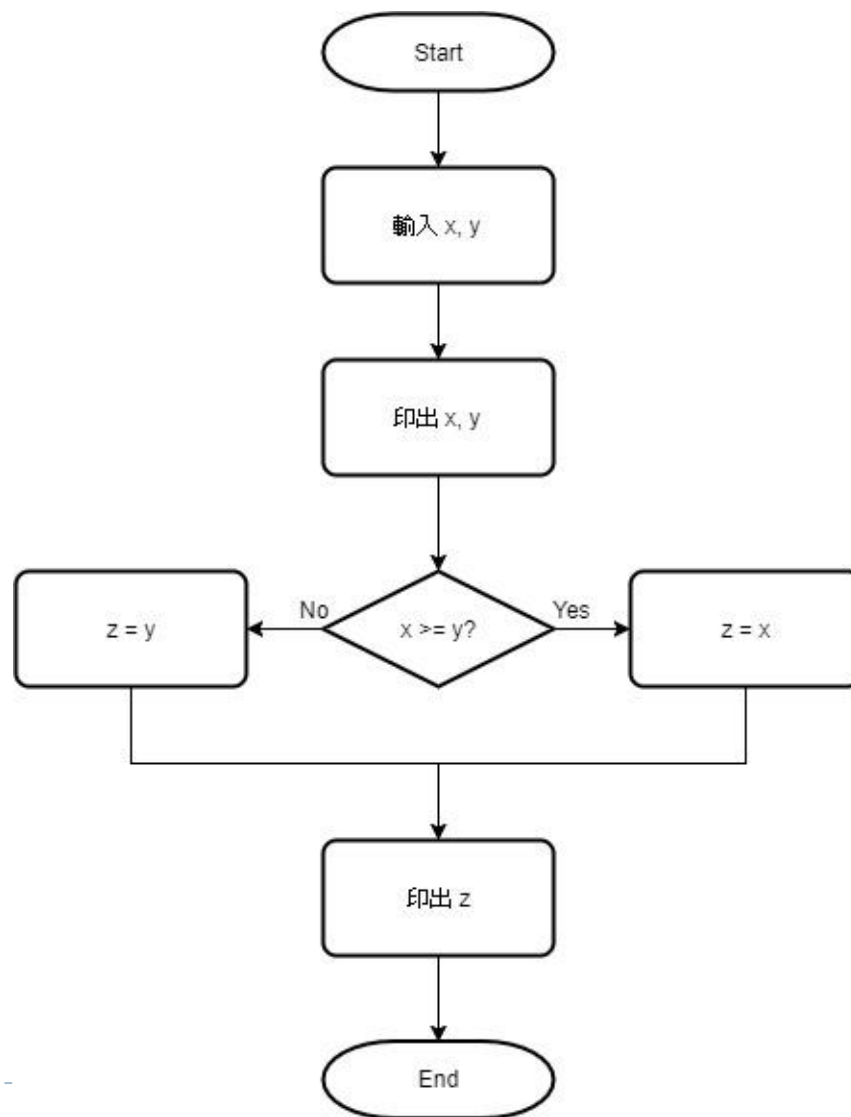
Ch2. 習題

- ▶ 以下各題均需先畫出流程圖。
- ▶ 2-1. 寫一程式，輸入 x 和 y ，如果 $x \geq y$ ，則列印 x ，否則列印 y
- ▶ 2-2. 寫一程式，輸入 x 和 y ，如果 x 和 y 都是正數，令 $z=1$ ，如兩者均為負數，令 $z=-1$ ，否則令 $z=0$
- ▶ 2-3. 寫一程式，輸入 x 、 y 、 u 、 v ，如果 $(x+y) > (u+v)$ ，則令 $z=x-y$ ，否則令 $z=u-v$



Ch2. 習題

► 2-1 流程圖：



Ch2. 習題

▶ 2-1 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x = ",x)
print("y = ",y)

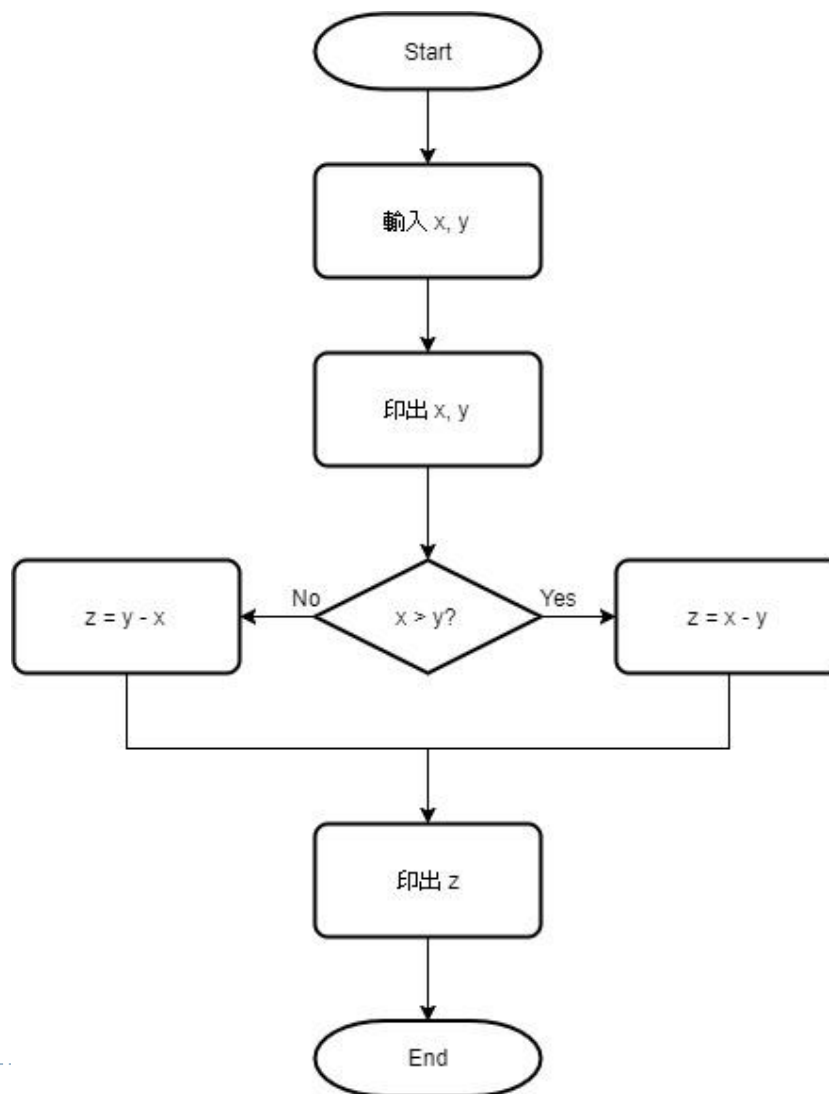
if x >= y:
    z = x
else:
    z = y

print("z = ",z)
```



Ch2. 習題

► 2-2 流程圖：



Ch2. 習題

▶ 2-2 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x = ",x)
print("y = ",y)

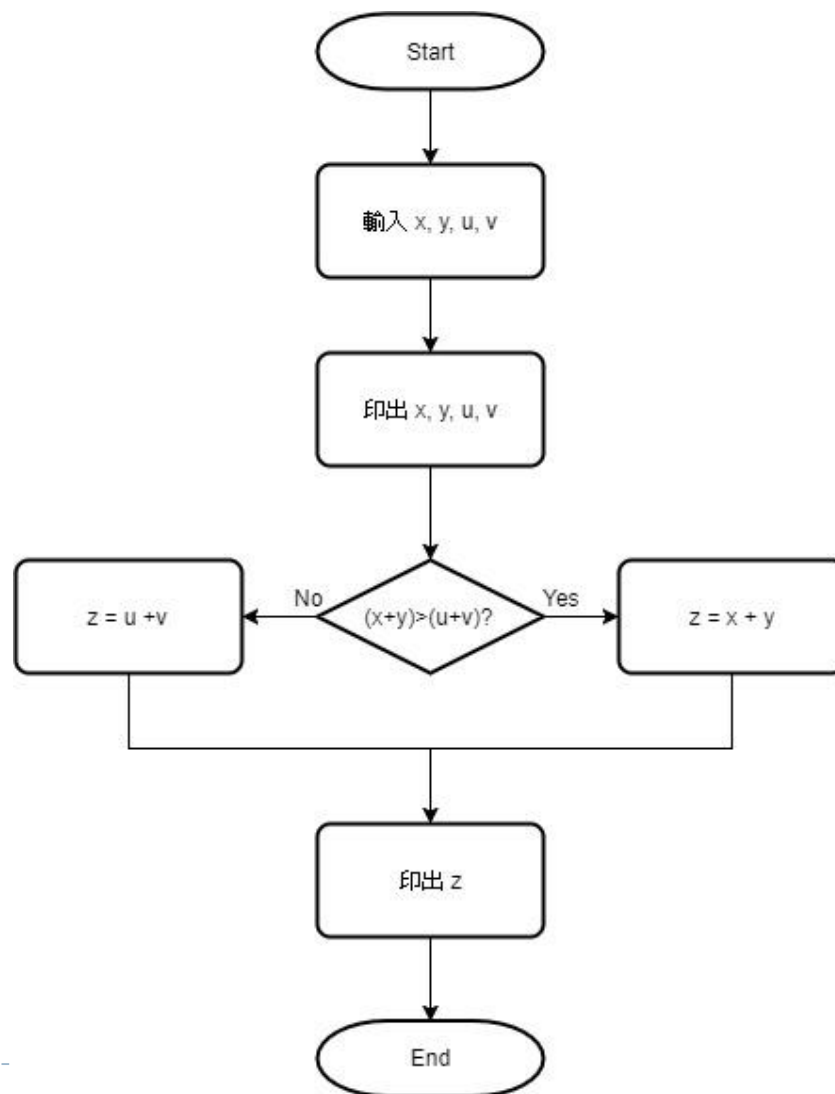
if x>0 and y>0:
    z = 1
elif x<0 and y<0:
    z = -1
else:
    z = 0

print("z = ",z)
```



Ch2. 習題

► 2-3 流程圖：



Ch2. 習題

▶ 2-3 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
u=int(input("Please input the value of variable u: "))
v=int(input("Please input the value of variable v: "))
print("x = ",x)
print("y = ",y)
print("u = ",u)
print("v = ",v)

if (x+y) > (u+v):
    z = x + y
else:
    z = u + v

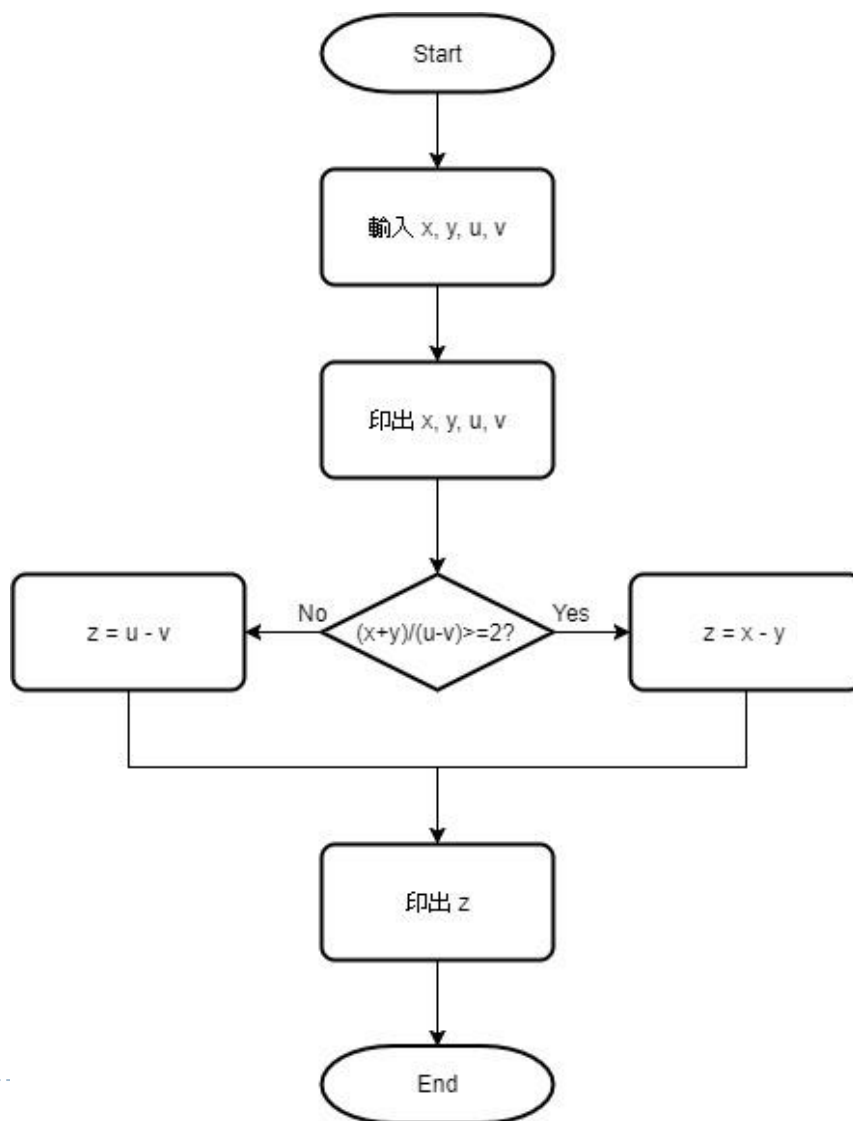
print("z = ",z)
```


Ch2. 習題

- ▶ 2-4. 寫一程式，輸入 x 、 y 、 u 、 v ，如果 $\frac{x+y}{u-v} \geq 2$ ，令 $z=x-y$ ，否則令 $z=u-v$ 。
- ▶ 2-5. 寫一程式，輸入 x 和 y ，如果 $x \geq y$ ，令 $z=x^2$ ，否則令 $z=y^2$

Ch2. 習題

► 2-4 流程圖：



Ch2. 習題

▶ 2-4 參考程式：

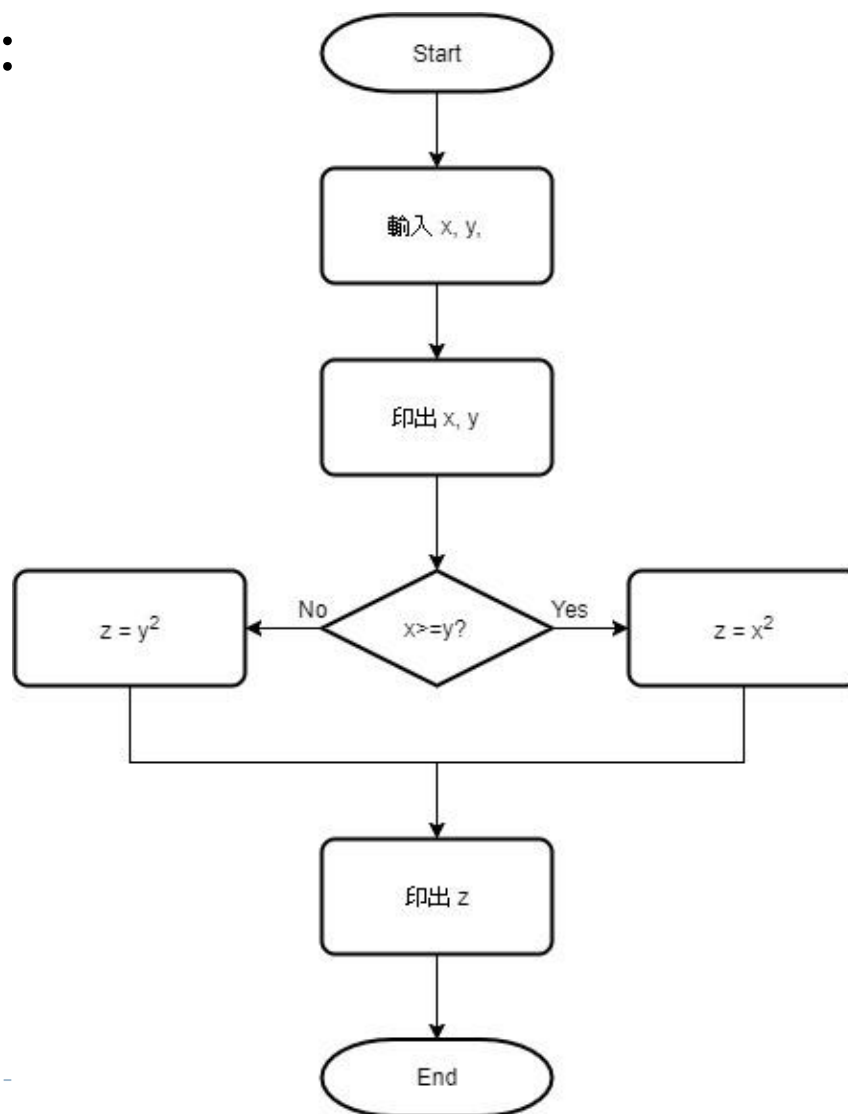
```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
u=int(input("Please input the value of variable u: "))
v=int(input("Please input the value of variable v: "))
print("x = ",x)
print("y = ",y)
print("u = ",u)
print("v = ",v)

if (x+y)/(u-v) >= 2:
    z = x - y
else:
    z = u - v

print("z = ",z)
```

Ch2. 習題

► 2-5 流程圖：



Ch2. 習題

▶ 2-5 參考程式：

```
x=int(input("Please input the value of variable x: "))
y=int(input("Please input the value of variable y: "))
print("x = ",x)
print("y = ",y)

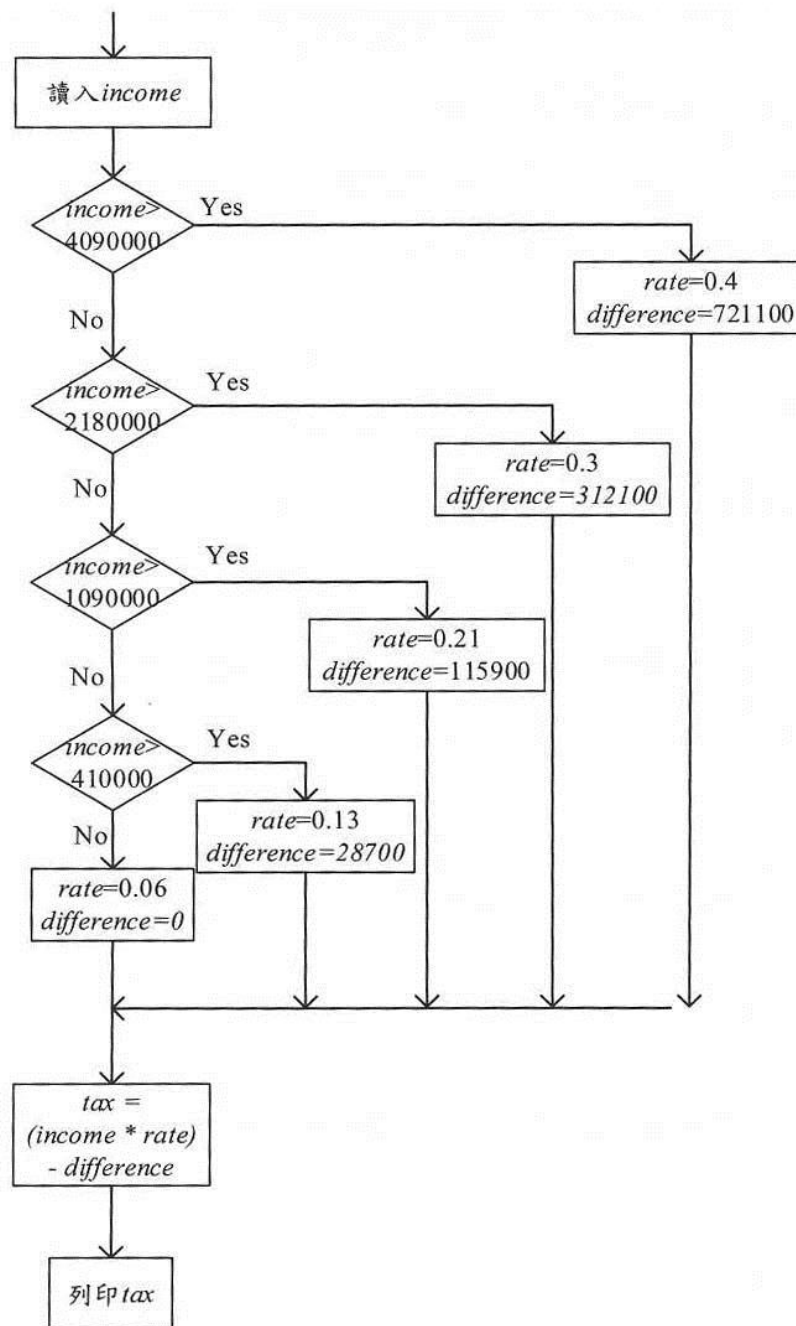
if x >= y:
    z = x**2
else:
    z = y**2

print("z = ",z)
```



Ch2. 習題

- 2-6. 依照以下的流程圖，
寫一程式。



Ch2. 習題

► 2-6 參考程式：

```
income = int(input("Please input your income: "))
if income > 4090000:
    rate = 0.4
    difference = 721100
elif income > 2180000:
    rate = 0.3
    difference = 312100
elif income > 1090000:
    rate = 0.21
    difference = 115900
elif income > 410000:
    rate = 0.13
    difference = 28700
else:
    rate = 0.06
    difference = 0
tax = (income * rate) - difference
print("tax = ", tax)
```

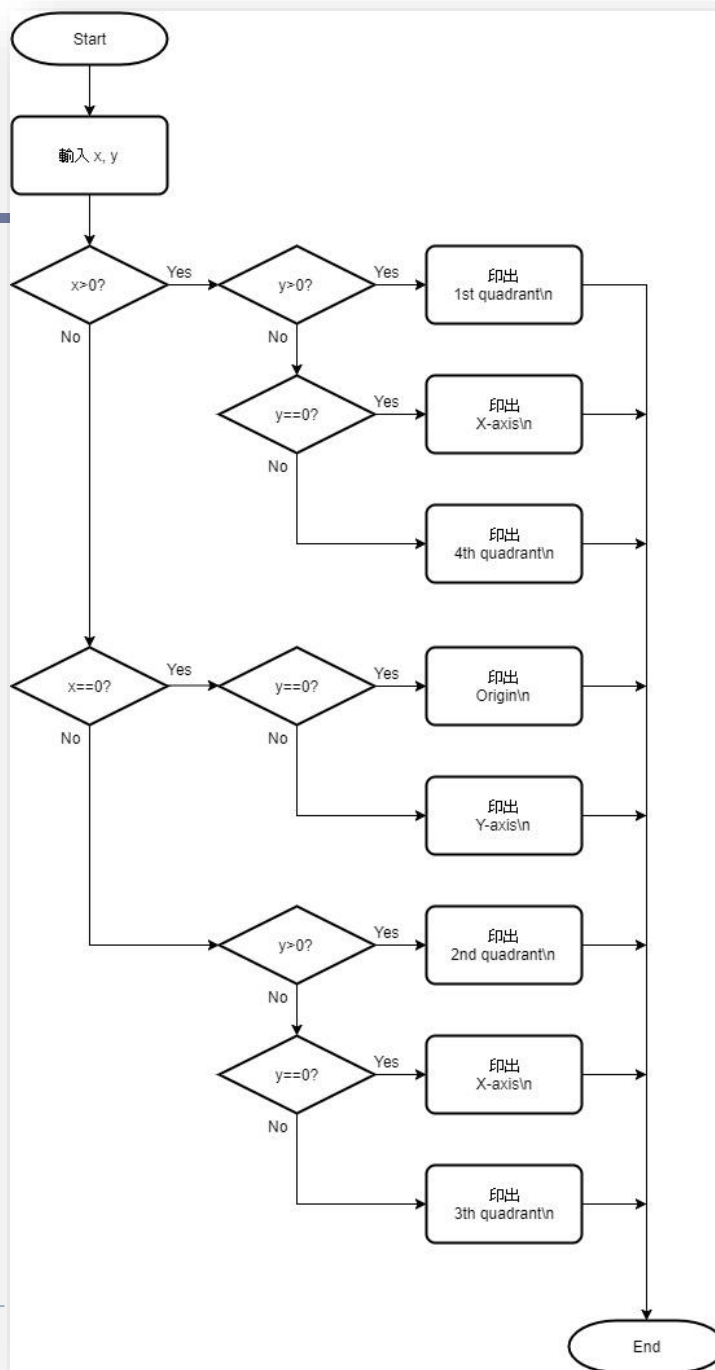
Ch2. 習題

- ▶ 2-7. 將以下程式的流程圖畫出來。

```
x=int(input("Enter x: "))
y=int(input("Enter y: "))
if x > 0:
    if y > 0:
        print("1st quadrant\n")
    elif y == 0:
        print("X=axis\n")
    else:
        print("4th quadrant\n")
elif x == 0:
    if y == 0:
        print("Origin\n")
    else:
        print("Y=axis\n")
else:
    if y > 0:
        print("2nd quadrant\n")
    elif y == 0:
        print("X-axis\n")
    else:
        print("3rd quadrant\n")
```


Ch2. 習題

► 2-7. 流程圖



休息一下~



Ch3. for迴圈指令

- ▶ 如果在程式中遇到要**重複同樣動作**的指令，就要使用迴圈指令



Ch3. for迴圈指令

- ▶ 讀入五個整數，
求這五個整數的和。

- ▶ 流程圖：

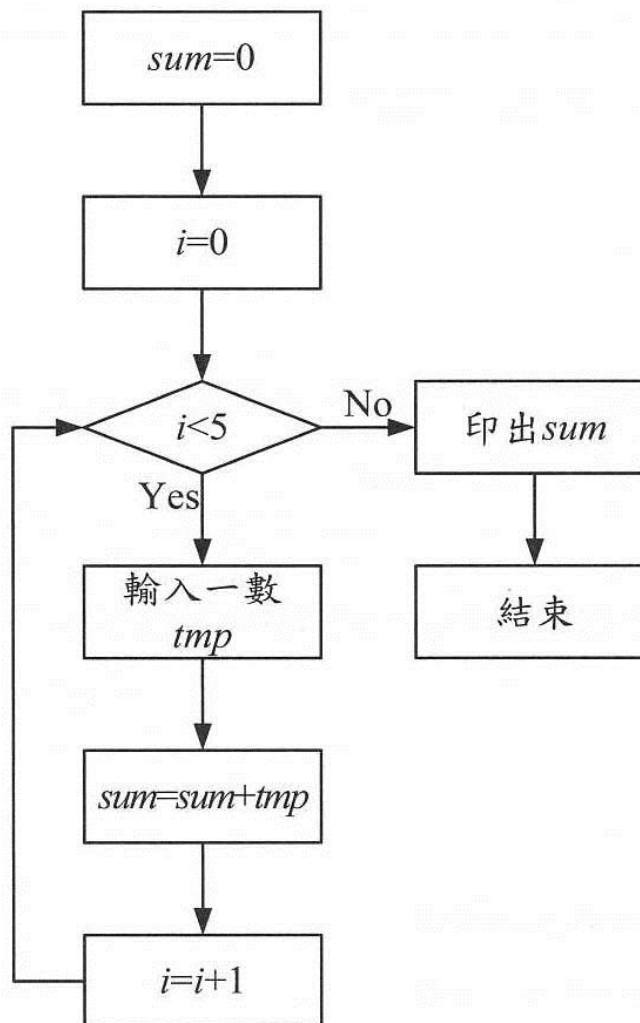


圖 3.1 ▶ 例題 3.1 的流程圖 (五個整數的總和)

Ch3. for迴圈指令

▶ 參考程式：

```
sum = 0

for i in range(5):
    tmp = int(input("Please input the value: "))
    sum = sum + tmp

print("sum = ",sum)
```

- ▶ *i* 會從0開始，每次加1，直到*i*=4為止(共5次)，在這個範圍內for迴圈內的敘述就要重複執行。



Ch3. for迴圈指令

- ▶ 求 N 個整數的和， N 由使用者輸入。

- ▶ 流程圖：

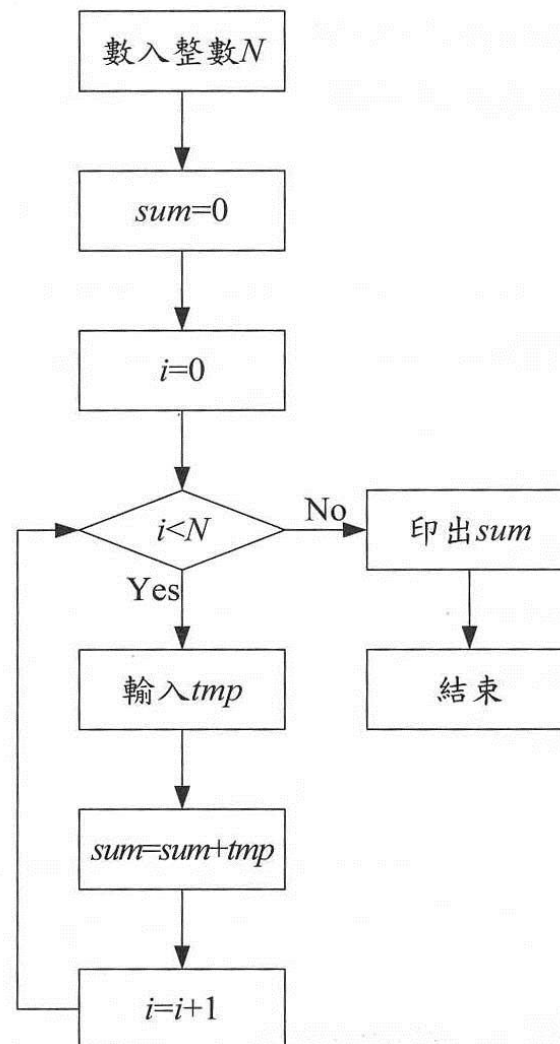


圖 3.2 ▶ 例題 3.2 的流程圖 (N 個數的和)

Ch3. for迴圈指令

▶ 參考程式：

```
N = int(input("Please input the value of variable N: "))
sum = 0

for i in range(N):
    tmp = int(input("Please input the value: "))
    sum = sum + tmp

print("sum = ",sum)
```

▶ 迴圈重複的次數由輸入的N決定，從0到N-1



Ch3. for迴圈指令

- ▶ 假設有N個正整數，求其中的最大值。
- ▶ 先假設最大值(MAX)等於0，然後逐一檢查讀入的每一個數字，如果比MAX還大，就令MAX等於這個數字，如果沒有比MAX大，就不做任何事情，最後MAX一定會是這一串數字中的最大值。



Ch3. for迴圈指令

► 流程圖：

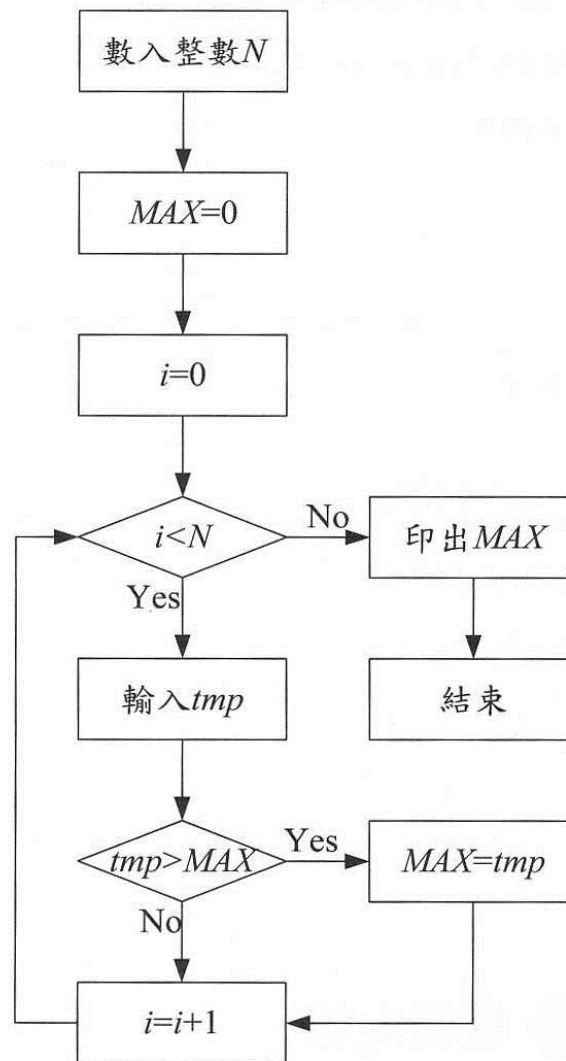


圖 3.3 ► 例題 3.3 的流程圖(求最大值)

Ch3. for迴圈指令

▶ 參考程式：

```
N = int(input("Please input the value of variable N: "))  
MAX = 0
```

```
for i in range(N):  
    tmp = int(input("Please input the value: "))  
    if tmp > MAX:  
        MAX = tmp
```

```
print("MAX = ",MAX)
```



Ch3. for迴圈指令

- ▶ 求最大奇數值。
- ▶ 先假設最大值(MAX)等於0，然後逐一檢查讀入的每一個數字，如果是奇數且比MAX還大，就令MAX等於這個數字，如果不是，就不做任何事情，最後MAX一定會是這一串數字中的奇數最大值。
- ▶ 可以用%(取餘數)運算，例如將n除以2之後若餘0，為偶數，若餘1則為奇數。

Ch3. for迴圈指令

► 流程圖：

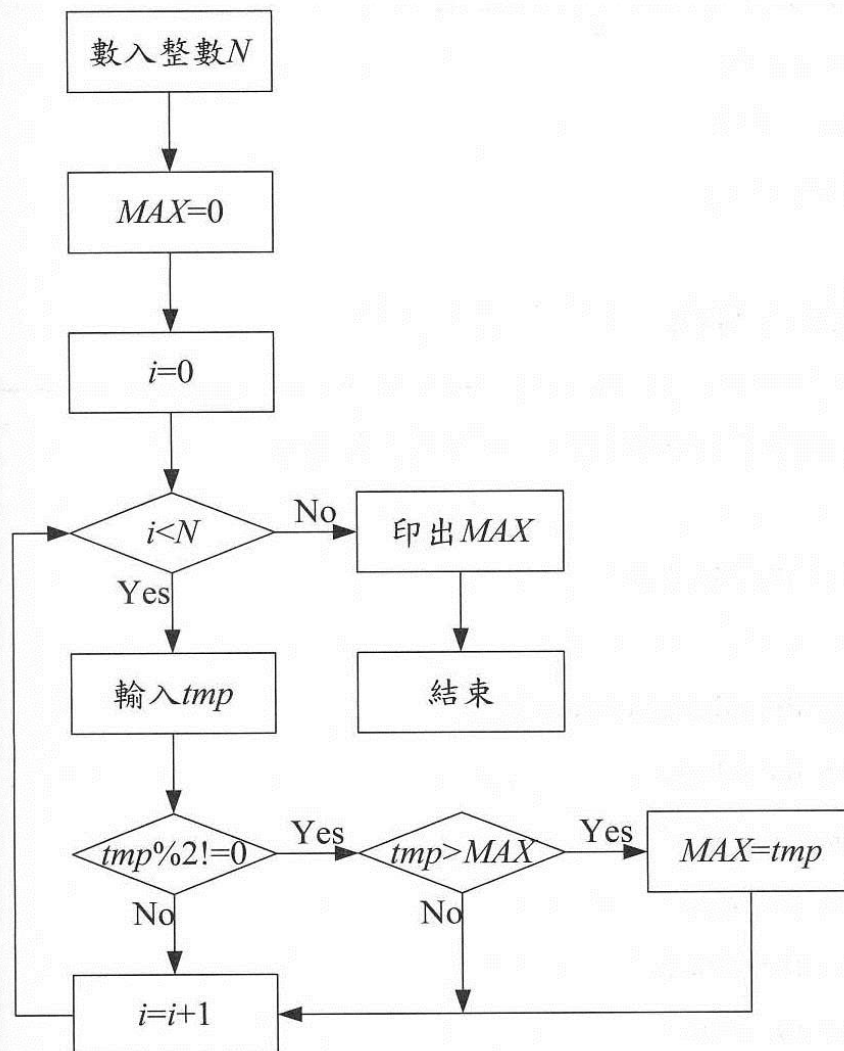


圖 3.4 ► 例題 3.4 的流程圖: 求最大奇數值

Ch3. for迴圈指令

▶ 參考程式：

```
N = int(input("Please input the value of variable N: "))  
MAX = 0
```

```
for i in range(N):  
    tmp = int(input("Please input the value: "))  
    if tmp % 2 != 0:  
        if tmp > MAX:  
            MAX = tmp
```

```
print("MAX = ",MAX)
```



Ch3. for迴圈指令

- ▶ 求N階層，即求 $1 \times 2 \times 3 \times \dots \times N$ 。
- ▶ 請求輸入N，讓for迴圈自2依序乘到N，所以要寫成

```
for i in range(2,N+1)
```

- ▶ 流程圖：

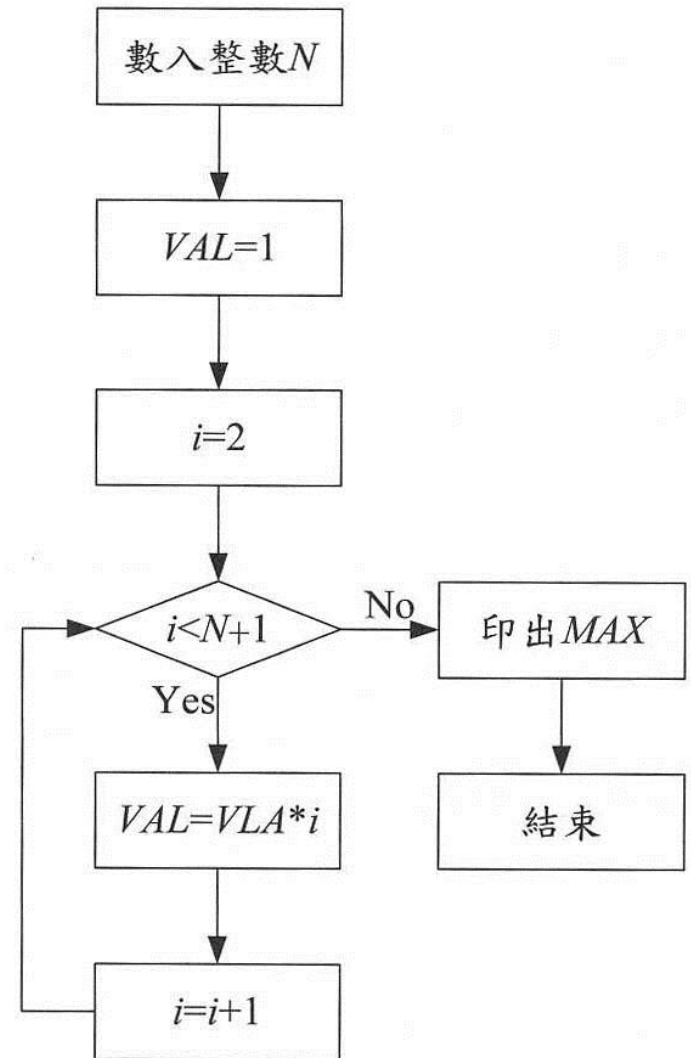


圖 3.5 ▶ 例題 3.5 的流程圖 (求 N 的階層)

Ch3. for迴圈指令

▶ 參考程式：

```
N = int(input("Please input the value of variable N: "))  
VAL = 1
```

```
for i in range(2,N+1):  
    VAL = VAL * i
```

```
print("VAL = ",VAL)
```

- ▶ $VAL = VAL * i$ 也可寫成 $VAL *= i$
- ▶ 注意迴圈若要執行到N，要寫成N+1，否則是執行到N-1



Ch3. for迴圈指令

- ▶ 輸出一個九九乘法表。
- ▶ 利用兩層迴圈，又稱巢狀迴圈。
- ▶ 迴圈的層數沒有限制，但越多效率越差。
- ▶ 流程圖：

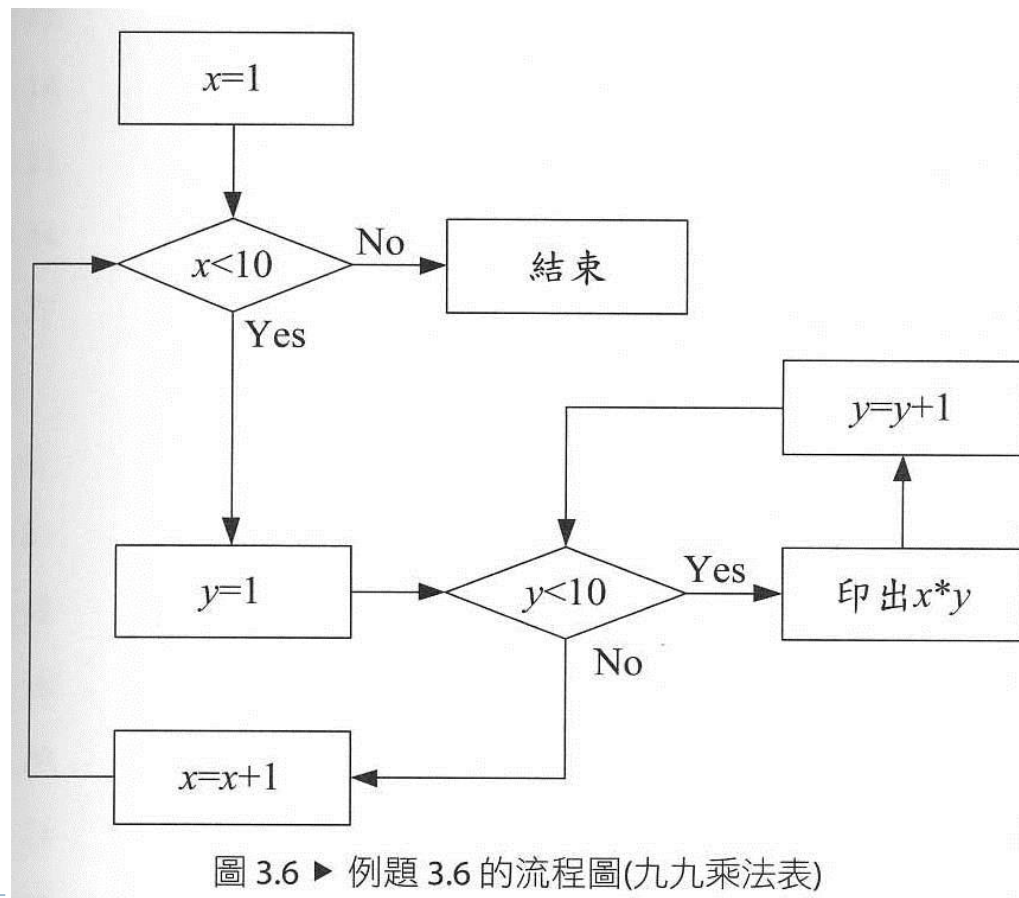


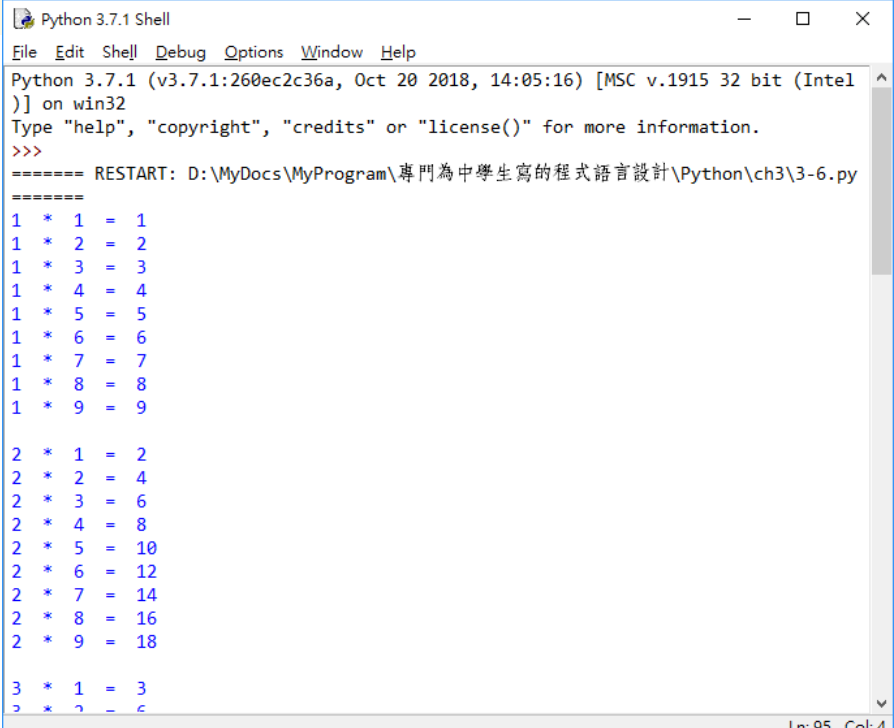
圖 3.6 ▶ 例題 3.6 的流程圖(九九乘法表)

Ch3. for迴圈指令

▶ 參考程式：

```
for x in range(1,10):  
    for y in range(1,10):  
        print(x, " * ", y, " = ", x*y),  
    print()
```

▶ 執行結果：

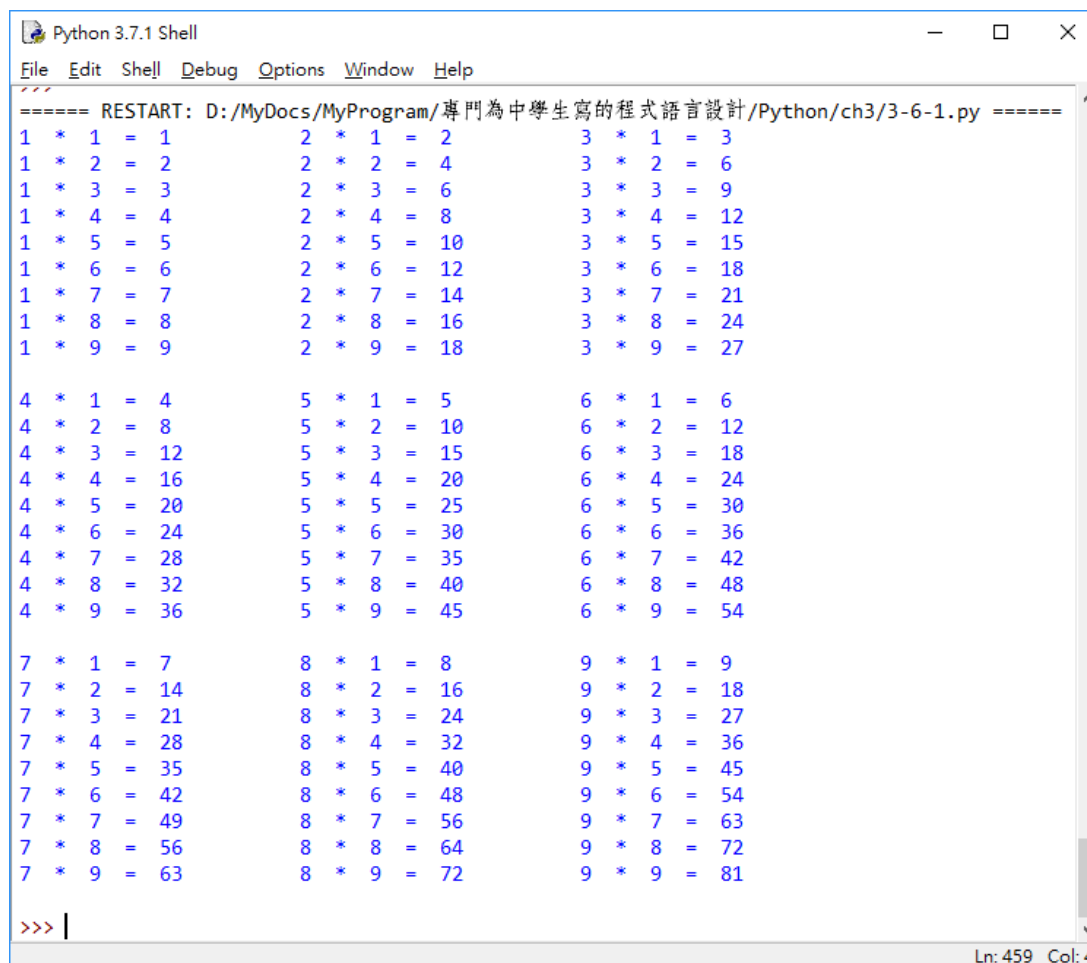


```
Python 3.7.1 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel  
)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:\MyDocs\MyProgram\專門為中學生寫的程式語言設計\Python\ch3\3-6.py  
=====  
1 * 1 = 1  
1 * 2 = 2  
1 * 3 = 3  
1 * 4 = 4  
1 * 5 = 5  
1 * 6 = 6  
1 * 7 = 7  
1 * 8 = 8  
1 * 9 = 9  
  
2 * 1 = 2  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18  
  
3 * 1 = 3  
3 * 2 = 6
```

Ln: 95 Col: 4

Ch3. for迴圈指令

- ▶ 請將程式改成如下輸出：



```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:/MyDocs/MyProgram/專門為中學生寫的程式語言設計/Python/ch3/3-6-1.py =====
1 * 1 = 1      2 * 1 = 2      3 * 1 = 3
1 * 2 = 2      2 * 2 = 4      3 * 2 = 6
1 * 3 = 3      2 * 3 = 6      3 * 3 = 9
1 * 4 = 4      2 * 4 = 8      3 * 4 = 12
1 * 5 = 5      2 * 5 = 10     3 * 5 = 15
1 * 6 = 6      2 * 6 = 12     3 * 6 = 18
1 * 7 = 7      2 * 7 = 14     3 * 7 = 21
1 * 8 = 8      2 * 8 = 16     3 * 8 = 24
1 * 9 = 9      2 * 9 = 18     3 * 9 = 27

4 * 1 = 4      5 * 1 = 5      6 * 1 = 6
4 * 2 = 8      5 * 2 = 10     6 * 2 = 12
4 * 3 = 12     5 * 3 = 15     6 * 3 = 18
4 * 4 = 16     5 * 4 = 20     6 * 4 = 24
4 * 5 = 20     5 * 5 = 25     6 * 5 = 30
4 * 6 = 24     5 * 6 = 30     6 * 6 = 36
4 * 7 = 28     5 * 7 = 35     6 * 7 = 42
4 * 8 = 32     5 * 8 = 40     6 * 8 = 48
4 * 9 = 36     5 * 9 = 45     6 * 9 = 54

7 * 1 = 7      8 * 1 = 8      9 * 1 = 9
7 * 2 = 14     8 * 2 = 16     9 * 2 = 18
7 * 3 = 21     8 * 3 = 24     9 * 3 = 27
7 * 4 = 28     8 * 4 = 32     9 * 4 = 36
7 * 5 = 35     8 * 5 = 40     9 * 5 = 45
7 * 6 = 42     8 * 6 = 48     9 * 6 = 54
7 * 7 = 49     8 * 7 = 56     9 * 7 = 63
7 * 8 = 56     8 * 8 = 64     9 * 8 = 72
7 * 9 = 63     8 * 9 = 72     9 * 9 = 81

>>> |
```

Ln: 459 Col: 4

Ch3. for迴圈指令

▶ 參考程式：

```
for x in range(1,10,3):  
    for y in range(1,10):  
        print(x, " * ", y, " = ", x*y, end="\t\t")  
        print(x+1, " * ", y, " = ", (x+1)*y, end="\t\t")  
        print(x+2, " * ", y, " = ", (x+2)*y)  
    print()
```

- ▶ “\t”稱跳脫字元，是Tab的意思。
- ▶ 常用的還有“\n”，是換行的意思。
- ▶ print()函數內定結尾會自動加上換行，加上end=“ ”參數後就會依我們加入的文字當結尾。

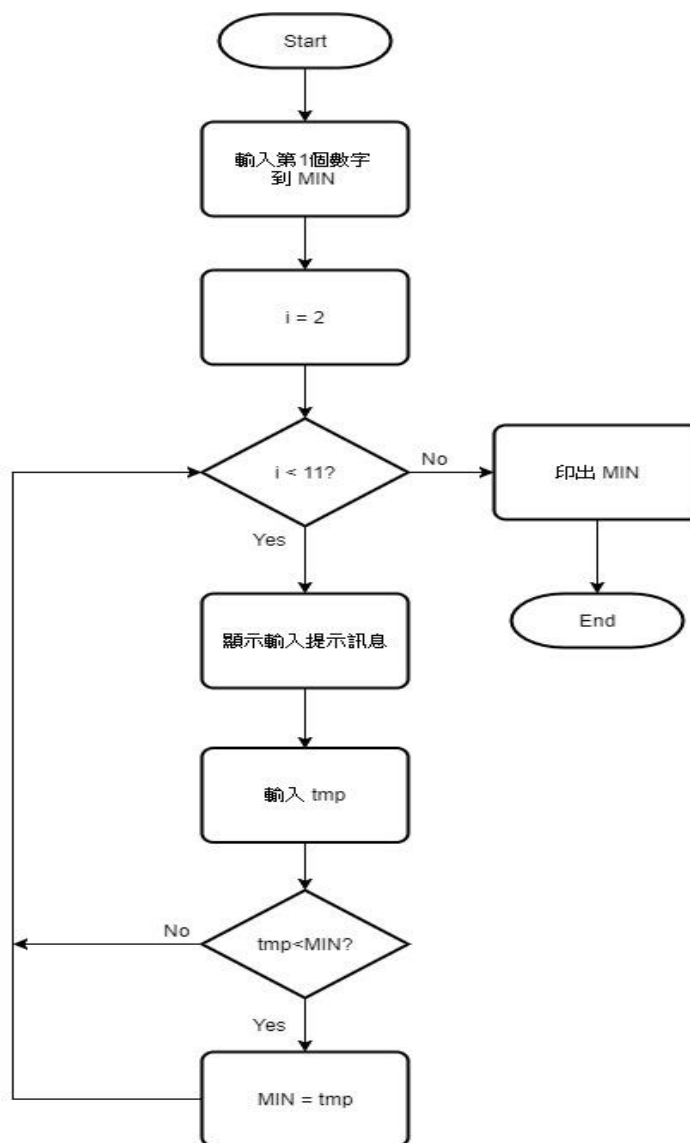
Ch3. 習題

- ▶ 3-1. 寫一程式，輸入10個整數，求其最小值。
- ▶ 3-2. 寫一程式，輸入N個整數，求其最小值。
- ▶ 3-3. 寫一程式，輸入10個整數，列出其中所有大於12的數字。
- ▶ 3-4. 寫一程式，輸入10個整數，列出其中所有大於12的數字的總和。



Ch3. 習題

▶ 3-1 流程圖：



Ch3. 習題

▶ 3-1 參考程式：

```
MIN = int(input("Please input the value of 1 : "))

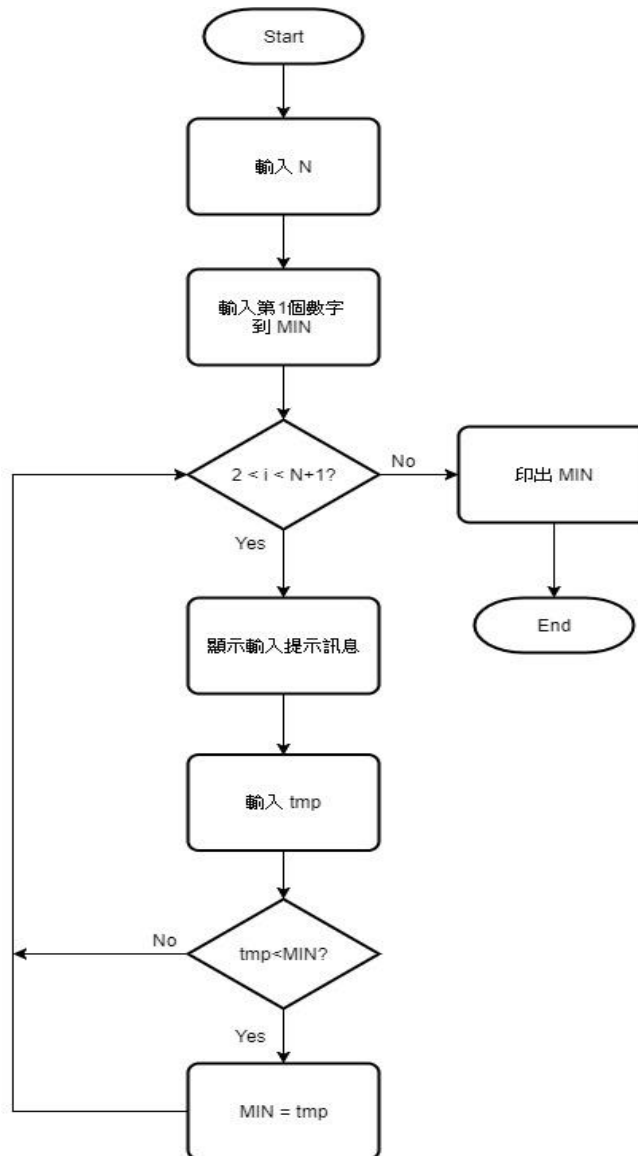
for i in range(2,11):
    print("Please input the value of",i,": ",end="")
    tmp = int(input())
    if tmp < MIN:
        MIN = tmp

print("MIN = ",MIN)
```

- ▶ 先將第一個輸入的值當成最小的，再依序跟後續輸入的做比較。

Ch3. 習題

▶ 3-2 流程圖：



Ch3. 習題

▶ 3-2 參考程式：

```
N = int(input("Please input the value of variable N: "))
MIN = int(input("Please input the value of 1 : "))

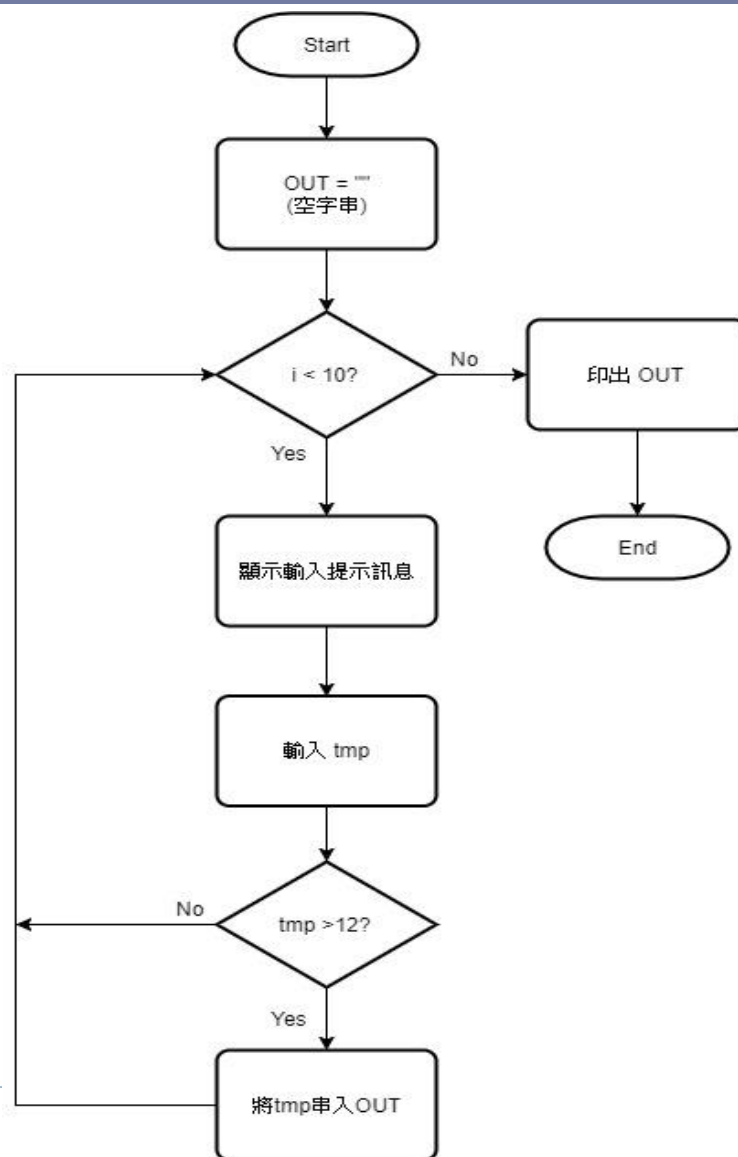
for i in range(2,N+1):
    print("Please input the value of",i,": ",end="")
    tmp = int(input())
    if tmp < MIN:
        MIN = tmp

print("MIN = ",MIN)
```



Ch3. 習題

► 3-3 流程圖：



Ch3. 習題

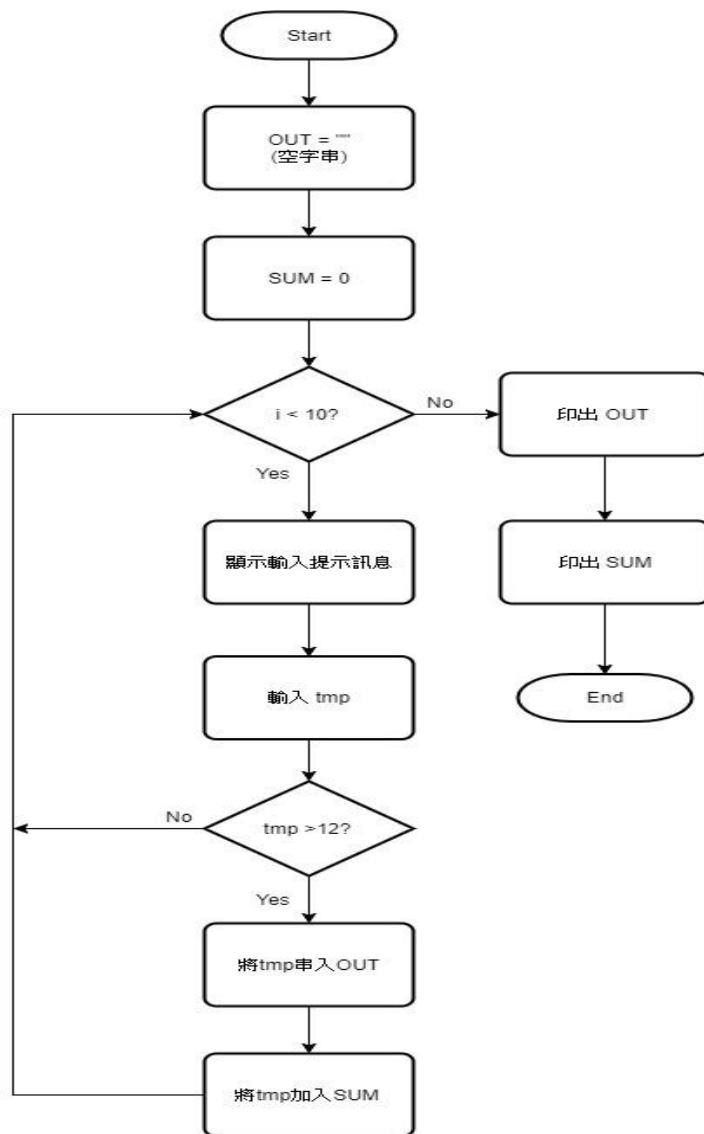
▶ 3-3參考程式：

```
OUT = ""
for i in range(10):
    print("Please input number",i+1,": ",end="")
    tmp = int(input())
    if tmp > 12:
        OUT = OUT + str(tmp) + " "
print("Number bigger then 12: ",OUT)
```

- ▶ 將符合大於12的數字轉成文字串入OUT，最後將OUT印出來。

Ch3. 習題

▶ 3-4 流程圖：



Ch3. 習題

▶ 3-4 參考程式：

```
OUT = ""
SUM = 0

for i in range(10):
    print("Please input number",i+1,": ",end="")
    tmp = int(input())
    if tmp > 12:
        OUT = OUT + str(tmp) + " "
        SUM += tmp

print("Number bigger then 12: ",OUT)
print("Sum = ",SUM)
```

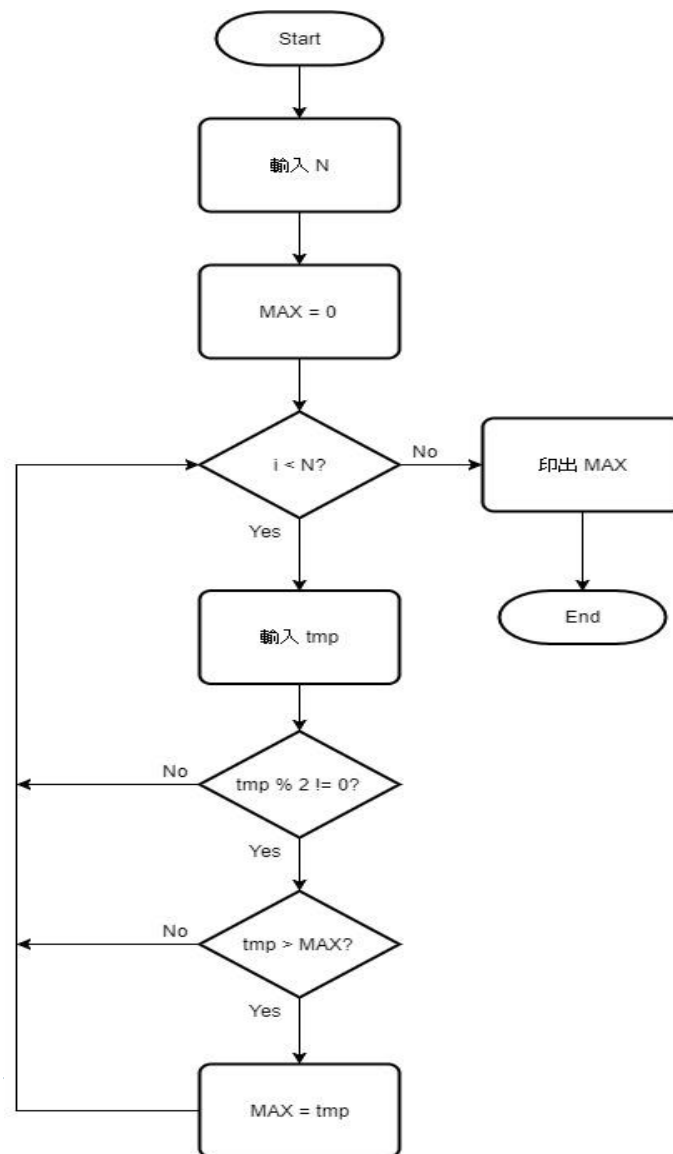


Ch3. 習題

- ▶ 3-5. 寫一程式，輸入N個數字，求其所有奇數中的最大值。例如輸入11,12,3,24,15，答案是15。
- ▶ 3-6. 寫一程式，輸入N個數字，求其所有正數之平方的加總。例如輸入1,-2,3,-4,5五個數字，得到
$$1^2 + (-2)^2 + 3^2 + (-4)^2 + 5^2 = 1 + 9 + 25 = 35$$
- ▶ 3-7. 寫一程式，輸入N個數字，其中有些是負數，將這些負數轉換成正數，例如-7會被轉換成7。

Ch3. 習題

▶ 3-5. 流程圖：



Ch3. 習題

▶ 3-5. 參考程式：

```
N = int(input("Please input the value of variable N: "))
MAX = 0

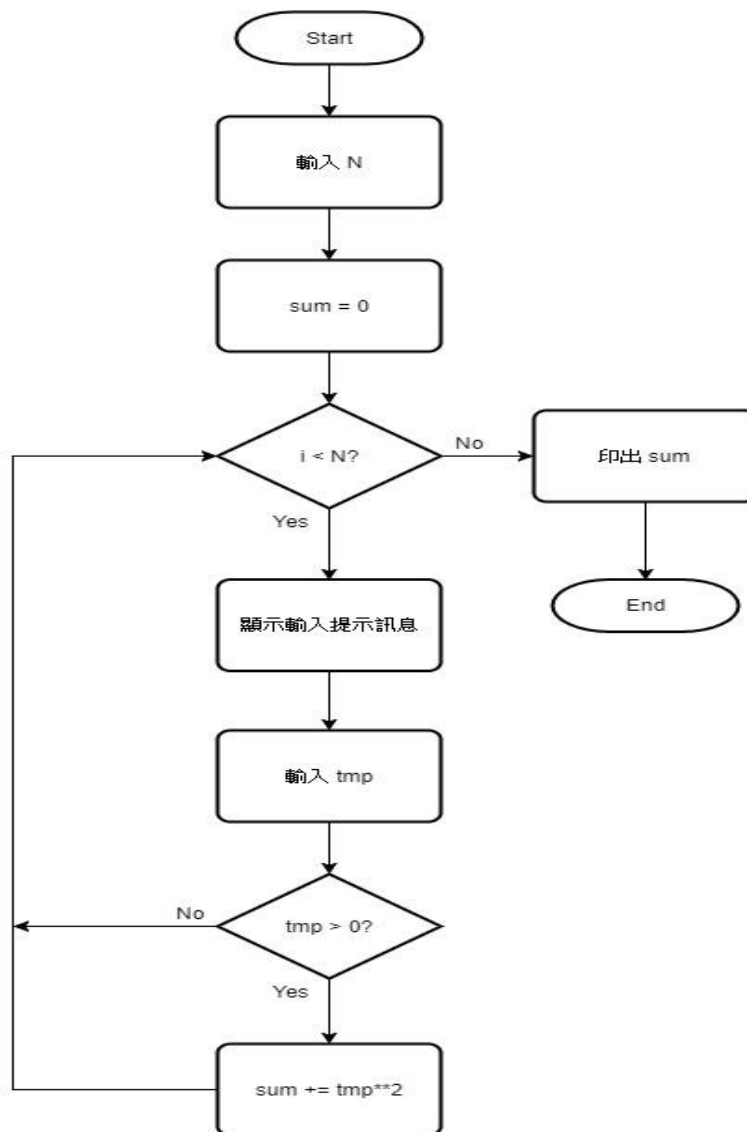
for i in range(N):
    tmp = int(input("Please input the value: "))
    if tmp % 2 != 0:
        if tmp > MAX:
            MAX = tmp

print("MAX = ",MAX)
```



Ch3. 習題

► 3-6. 流程圖：



Ch3. 習題

▶ 3-6. 參考程式：

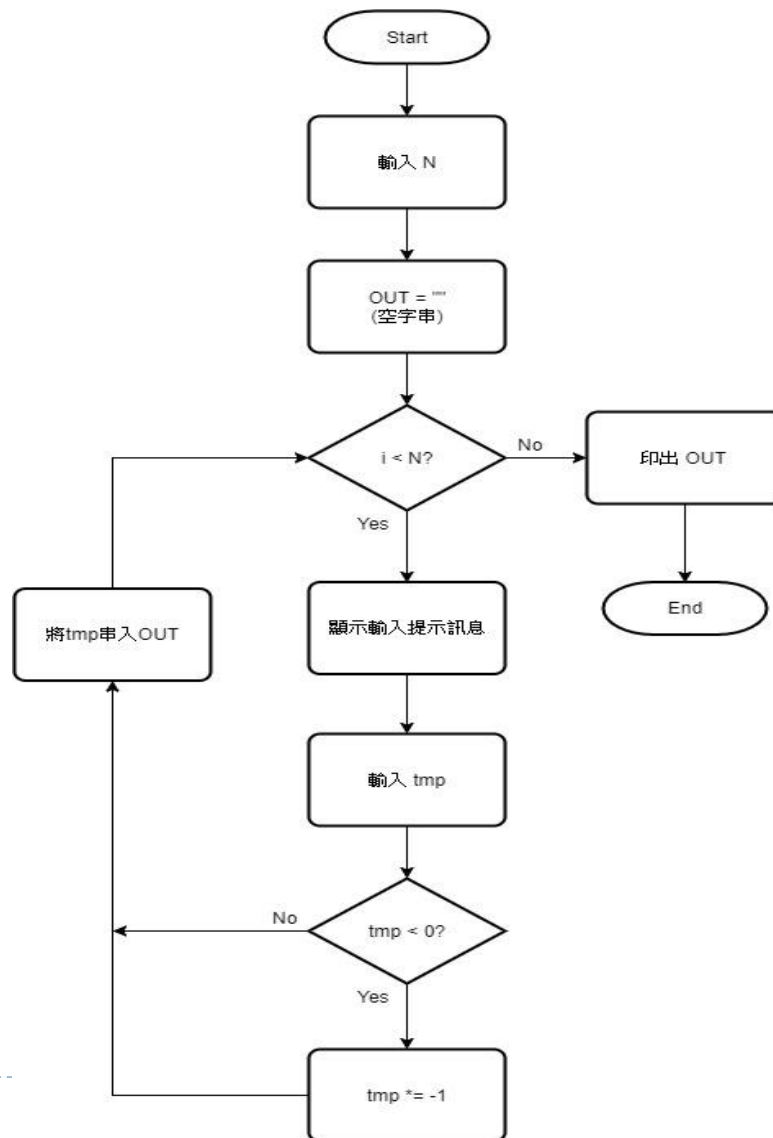
```
N = int(input("Please input the value of variable N: "))
sum = 0

for i in range(N):
    print("Please input number",i+1,": ",end="")
    tmp = int(input())
    if tmp > 0:
        sum += tmp**2

print("Sum = ",sum)
```

Ch3. 習題

▶ 3-7. 流程圖：



Ch3. 習題

▶ 3-7. 參考程式：

```
N = int(input("Please input the value of variable N: "))
OUT = ""

for i in range(N):
    print("Please input number",i+1,": ",end="")
    tmp = int(input())
    if tmp < 0:
        tmp *= -1
    OUT = OUT + str(tmp) + " "

print("Result =",OUT)
```

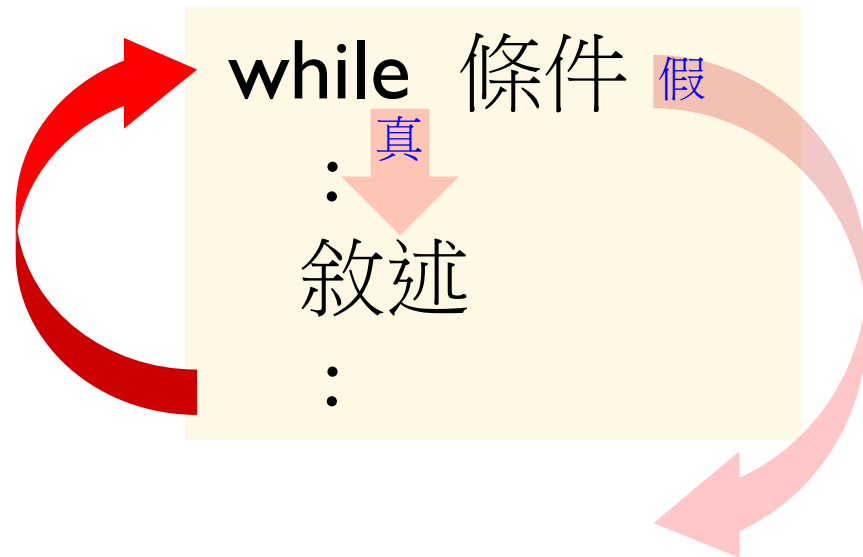


休息一下~



Ch4. while迴圈指令

- ▶ 通常for圈都用在確定要重複執行多少次的情況，若重複的次數不定，我們可以用while迴圈。
- ▶ while以“是”、“否”來做為終止條件。
- ▶ 當條件為真時執行迴圈，為假時結束迴圈



Ch4. while迴圈指令

- ▶ 讀入一連串的數字，但事先不知道數字的數目。
- ▶ 設定一個特殊的數字，一旦讀到這個數字就終止迴圈。
- ▶ 假設輸入的都是正數，一旦讀到負數就終止。
- ▶ 流程圖：

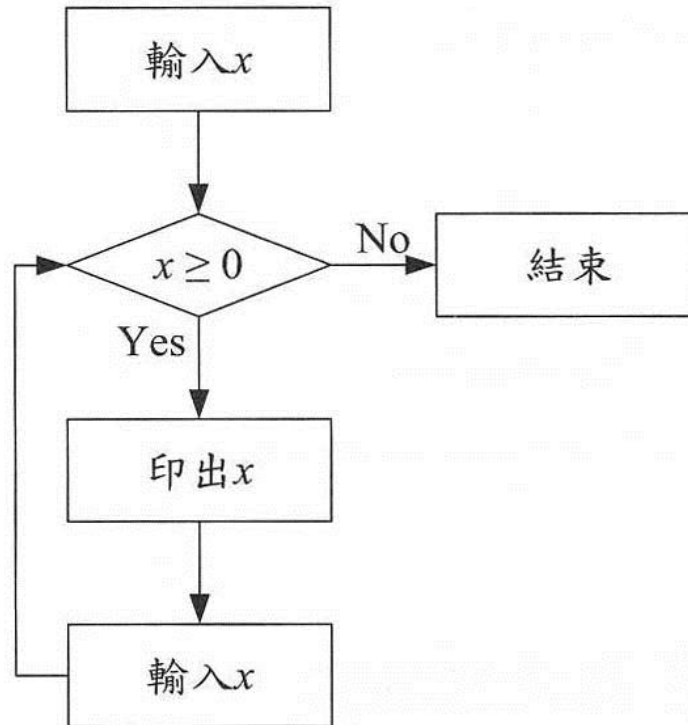


圖 4.1 ▶ 例題 4.1 的流程圖

Ch4. while迴圈指令

▶ 參考程式：

```
x = int(input("Please input the value of variable: "))  
  
while x >= 0:  
    print("x = ",x)  
    x = int(input("Please input the value of variable: "))
```

- ▶ 當 $x \geq 0$ 成立時，就執行迴圈內敘述，部成立時就終止迴圈。
- ▶ **while**(條件) 或 **while** 條件 都可以，即括符可加可不加。

Ch4. while迴圈指令

- ▶ 求最大公約數(Greatest Common Divisor，簡寫G.C.D.，或稱最大公因數)。
- ▶ 指能夠整除多個整數的最大正整數。而多個整數不能都為零。例如8和12的最大公因數為4。
- ▶ 輾轉相除法：兩數相除，取餘數重複進行相除，直到餘數為0時，前一個除數即為最大公約數。



Ch4. while 迴圈指令

► 流程圖：

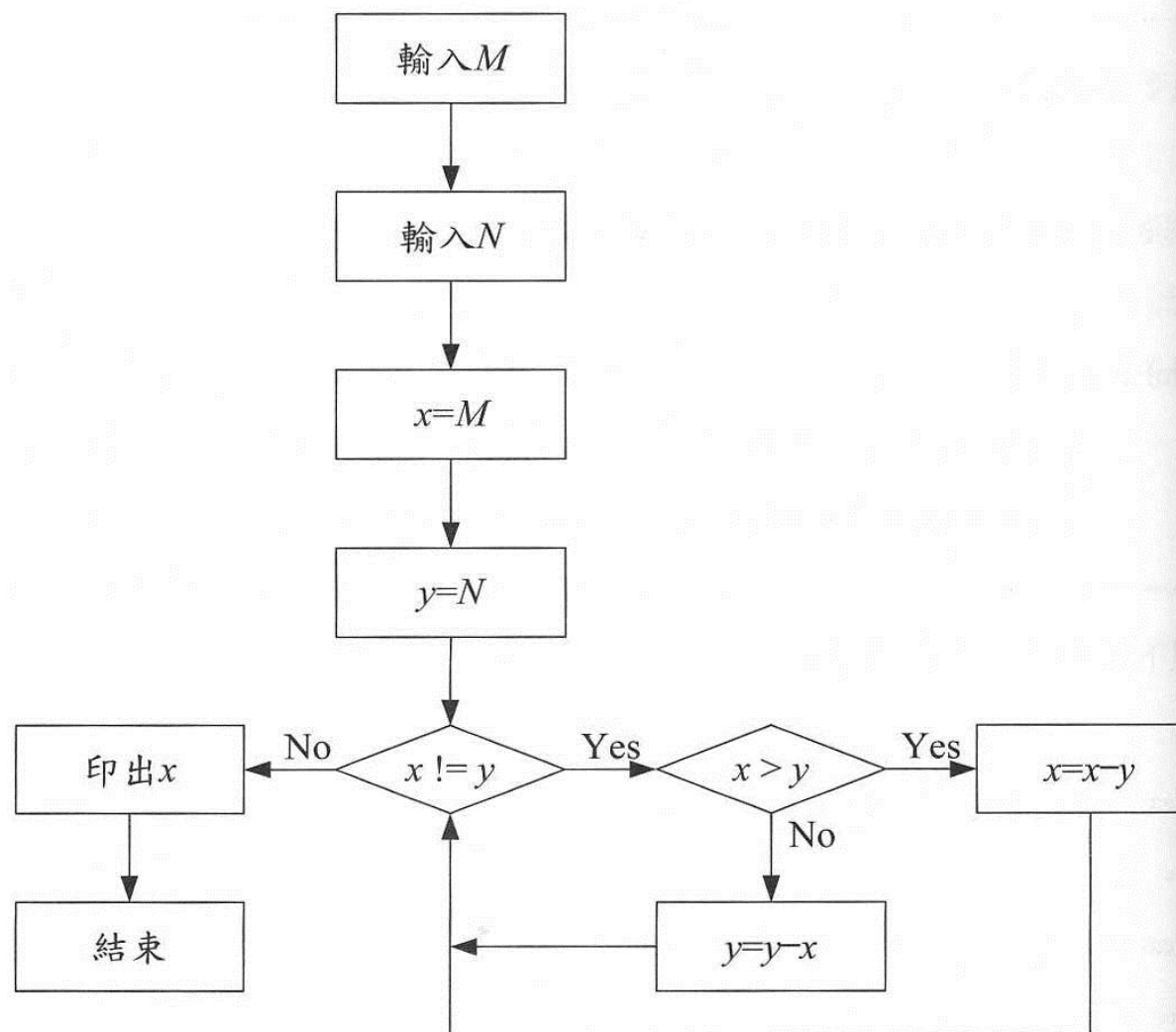


圖 4.2 ► 例題 4.2 的流程圖(求最大公約數)

Ch4. while迴圈指令

▶ 參考程式：

```
M = int(input("Please input the value of variable M: "))
N = int(input("Please input the value of variable N: "))
x = M
y = N
```

```
while (x != y):
    if (x > y):
        x = x-y
    else:
        y = y-x
```

```
print("x = ",x)
```



Ch4. while迴圈指令

▶ 參考程式二：

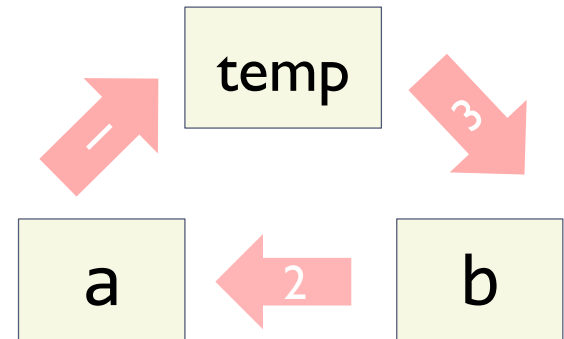
```
a = int(input("Please input the value of variable a: "))
b = int(input("Please input the value of variable b: "))
# 如果 a < b，把a和b對調
if a < b:
    a, b = b, a

r = a % b
# While迴圈會計算到r=0才會停止
while r != 0:
    a = b
    b = r
    r = a % b
print("GCD是 " + str(b))
```

Ch4. while迴圈指令

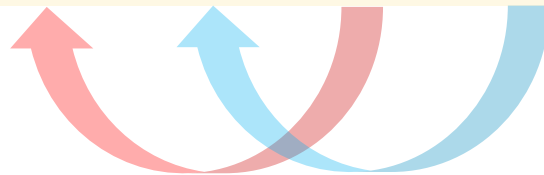
- ▶ 關於兩變數交換內容的方式：
- ▶ 其他語言都須借助第三個變數當中介，例如要交換a,b的內容，則需寫成

```
temp = a  
a = b  
b = temp
```



- ▶ 在python語言只需要一行即可

```
a , b = b , a
```



Ch4. while迴圈指令

- ▶ 計算 N 個數字的和。
- ▶ 檢查 i 有沒有超過 N ，超過就不做了。
- ▶ 流程圖：

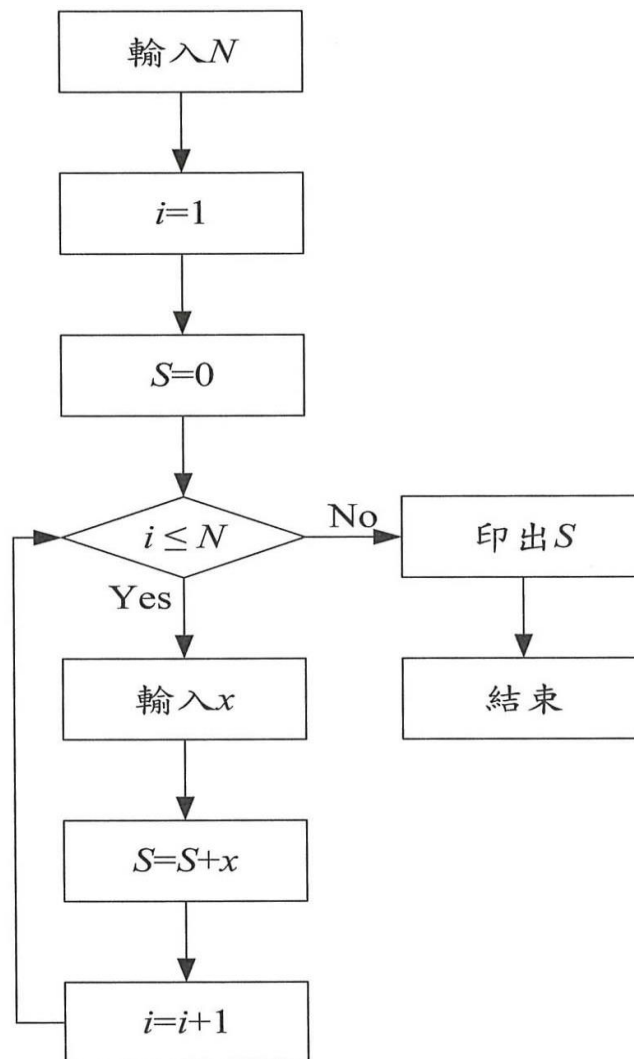


圖 4.3 ▶ 例題 4.3 的流程圖(求 N 個數字的和)

Ch4. while迴圈指令

▶ 參考程式：

```
N = int(input("Please input the value of variable N: "))
i = 1
S = 0

while i <= N:
    x = int(input("Please input the value of variable x: "))
    S = S + x
    i = i + 1

print("S = ",S)
```



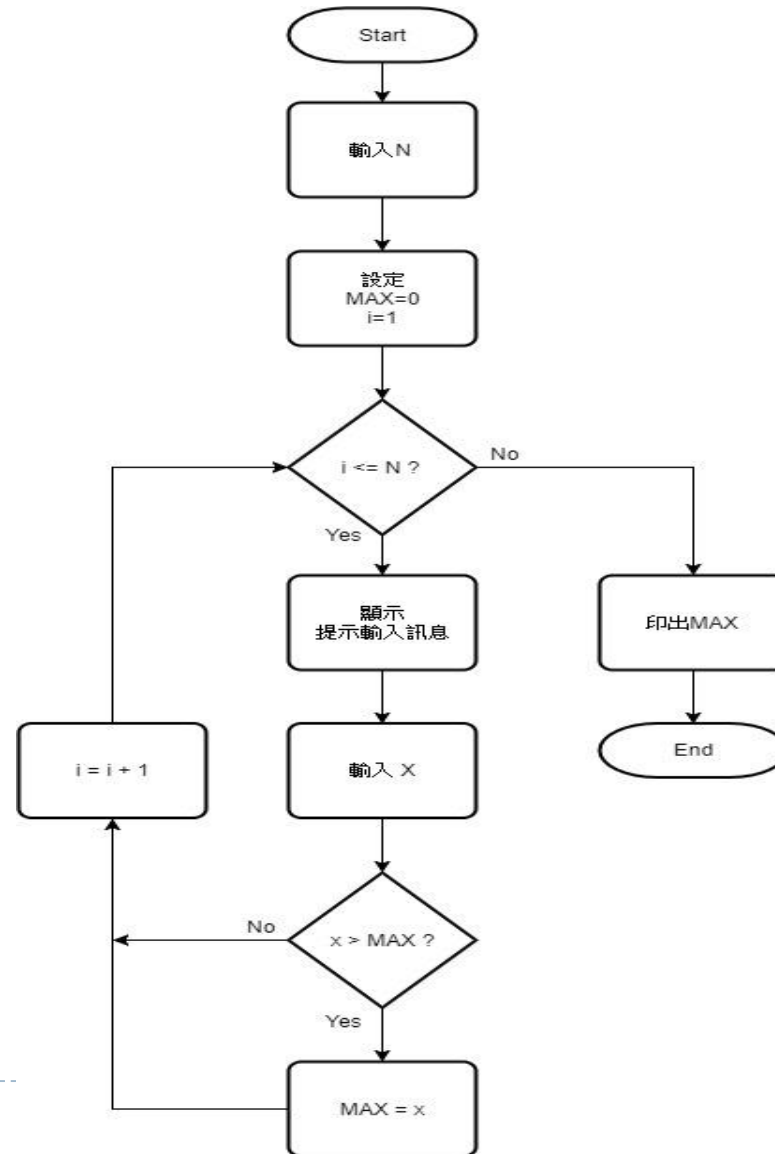
Ch4. 習題

- ▶ 4-1. 利用 `while` 寫一程式求 N 個數字的最大值。
- ▶ 4-2. 利用 `while` 寫一程式求一個等差級數數字的和，一共有 N 個數字，程式應該輸入最小的起始值以及數字間的差。(即從起始值開始，間格差，共 N 個 的總和)
- ▶ 4-3. 利用 `while` 寫一程式，讀入 N 個數字，然後找出所有小於 13 的數，再求這些數字的和。



Ch4. 習題

► 4-1. 流程圖：



Ch4. 習題

► 4-1.參考程式：

```
N = int(input("How many number? "))
MAX = 0
i = 1

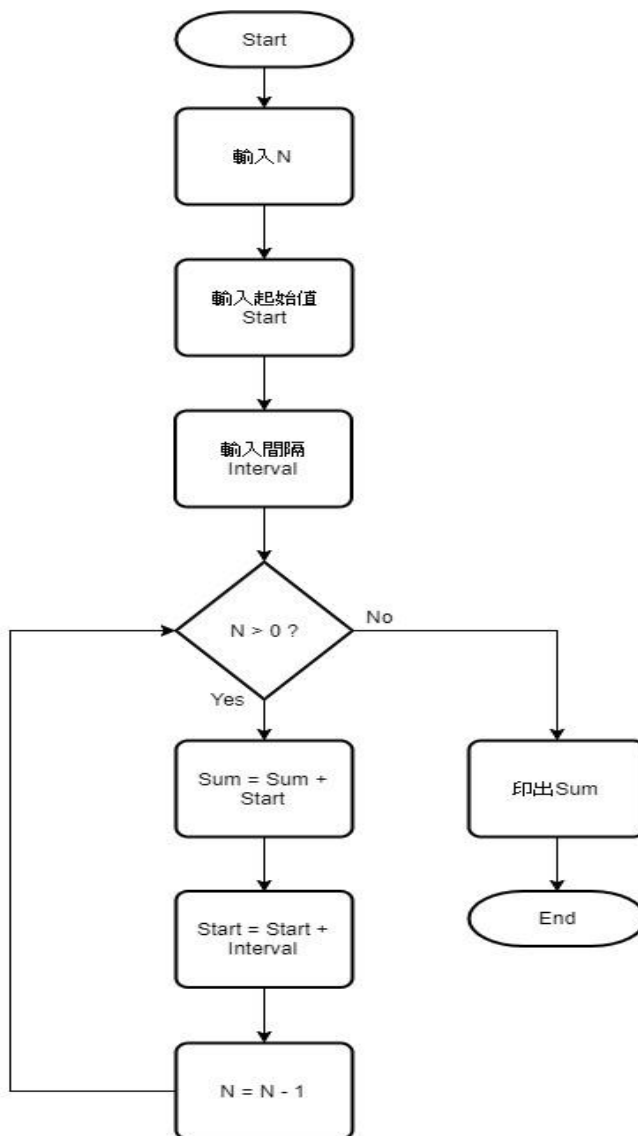
while i <= N:
    print("Please input the value of number",i,": ",end="")
    x = int(input())
    if x > MAX:
        MAX = x
    i = i + 1

print("MAX = ",MAX)
```



Ch4. 習題

► 4-2. 流程圖：



Ch4. 習題

▶ 4-2. 參考程式：

```
N = int(input("Please input N: "))
Start = int(input("Please input the start number: "))
Interval = int(input("Please input the interval: "))
Sum = 0

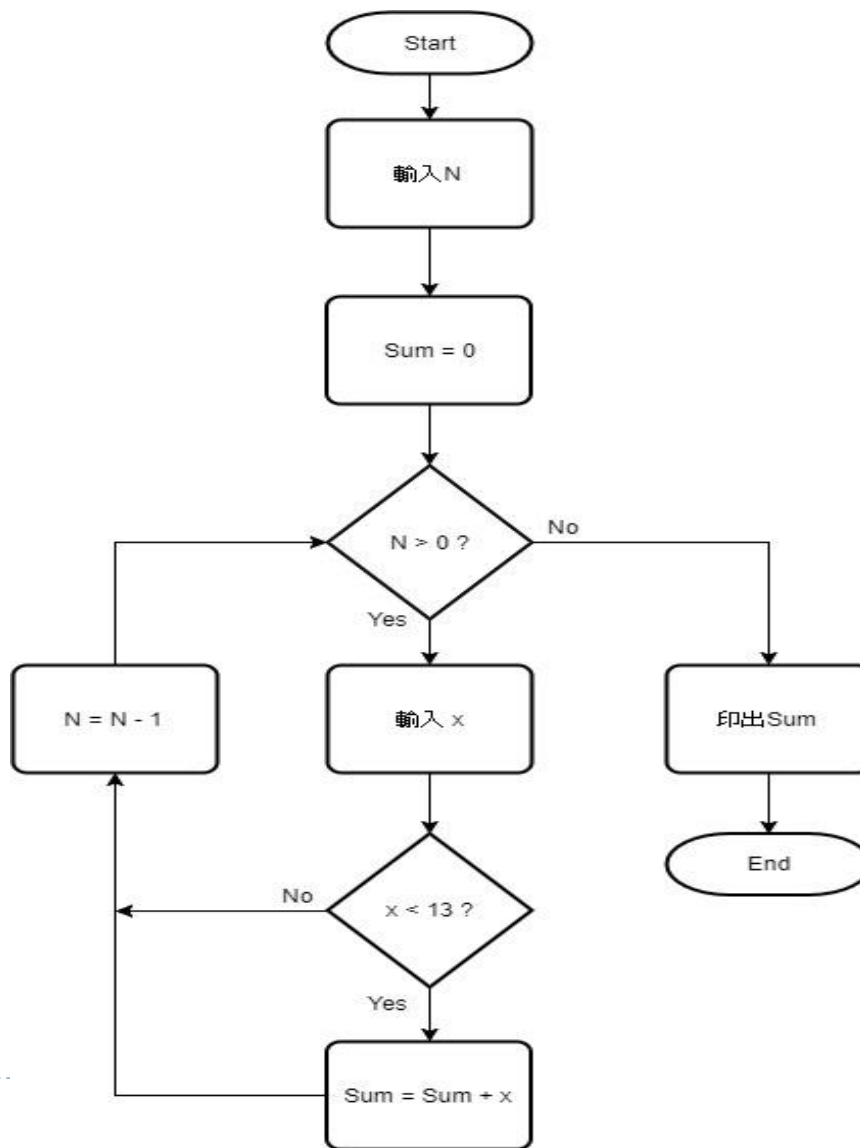
while N>0:
    Sum += Start
    Start += Interval
    N -= 1

print("Total = ",Sum)
```



Ch4. 習題

► 4-3. 流程圖：



Ch4. 習題

► 4-3. 參考程式：

```
N = int(input("Please input N: "))
Sum = 0

while N > 0:
    x = int(input("Please input the value: "))
    if x < 13:
        Sum += x
    N -= 1

print("Total = ", Sum)
```

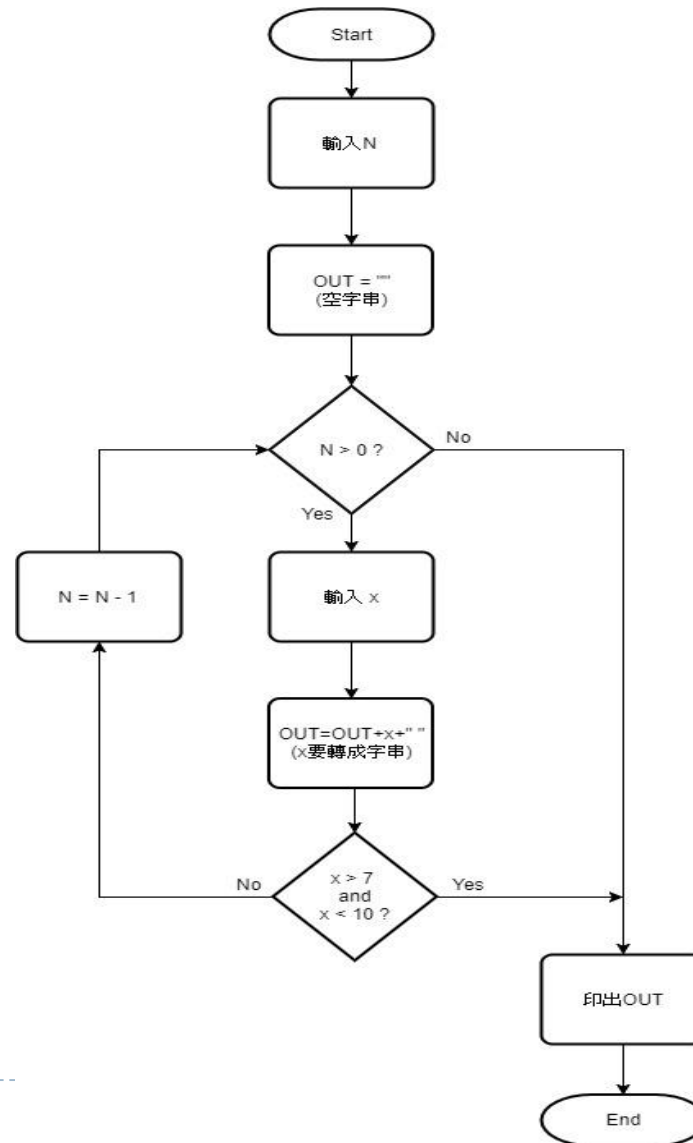
Ch4. 習題

- ▶ 4-4. 利用 `while` 寫一程式，讀入 N 個數字，找到第一個大於 7 而小於 10 的數字就停止，而且列印出這個數字。
- ▶ 4-5. 利用 `while` 寫一程式，讀入 a_1, a_2, \dots, a_5 和 b_1, b_2, \dots, b_5 。找到第一個 $a_i > b_i$ ，即停止，並列印出 a_i 及 b_i 。



Ch4. 習題

► 4-4. 流程圖：



Ch4. 習題

▶ 4-4. 參考程式：

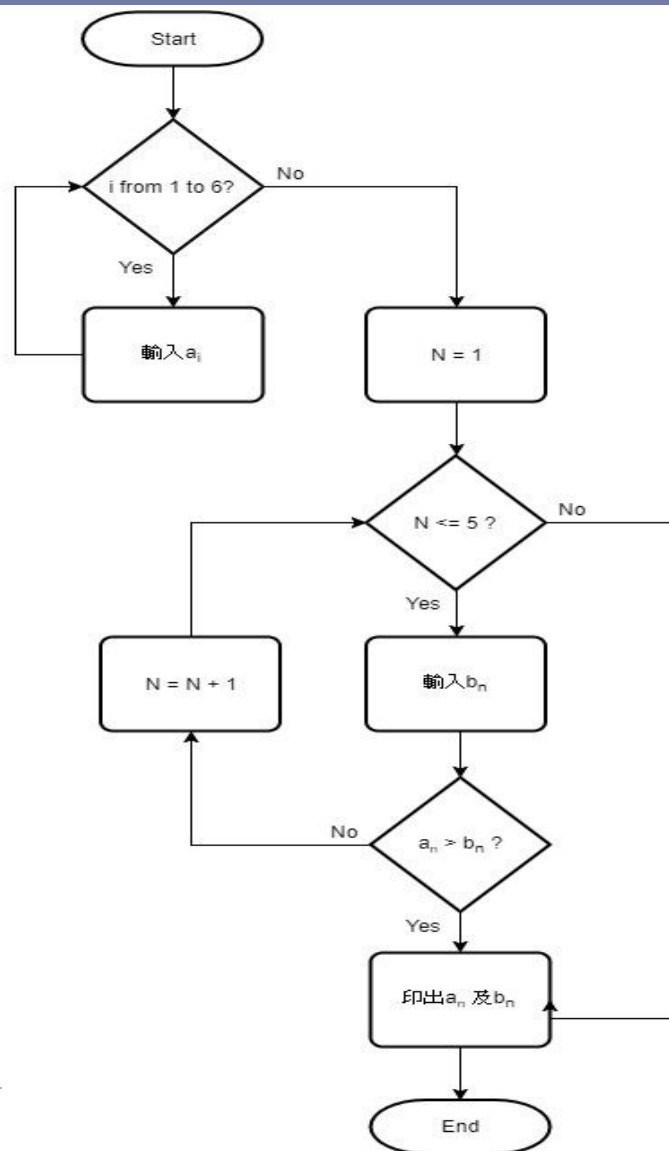
```
N = int(input("Please input N: "))

while N > 0:
    x = int(input("Please input the value: "))
    if x>7 and x<10:
        break
    N -= 1

print("STOP ON: ",x)
```


Ch4. 習題

► 4-5. 流程圖：



Ch4. 習題

▶ 4-5. 參考程式(依序輸入a和b)

```
N = 1
while N <=5:
    a = int(input("Please input a" + str(N) + ": "))
    b = int(input("Please input b" + str(N) + ": "))
    if a > b:
        print("a"+str(N)+" =",a," >  b"+str(N)+" =",b)
        break
    N += 1
```

Ch4. 習題

► 4-5. 參考程式(用陣列來做，但進度在下一章)：

```
a = [1 for i in range(6)]
b = [1 for i in range(6)]

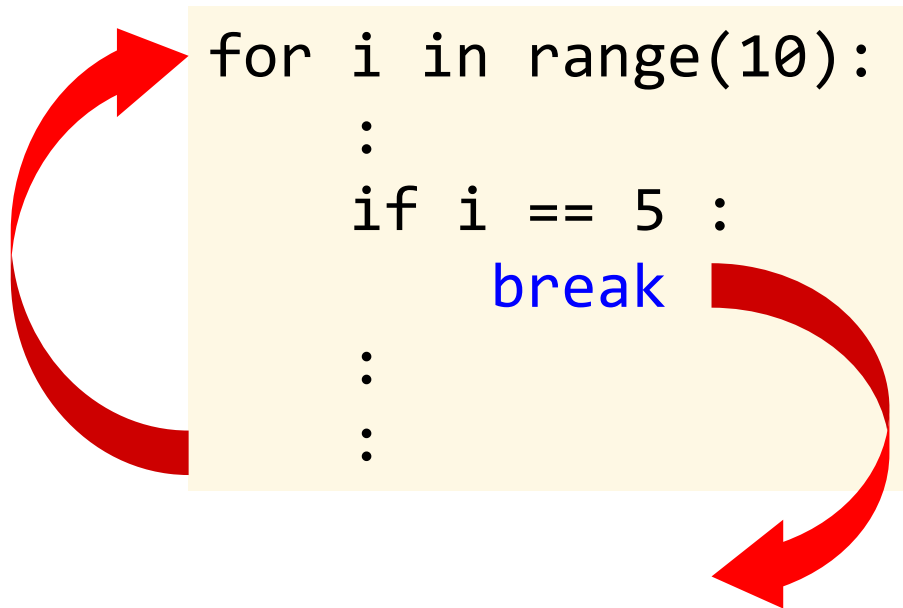
for i in range(1,6):
    a[i] = int(input("Please input a" + str(i) + ": "))

N = 1
while N <=5:
    b[N] = int(input("Please input b" + str(N) + ": "))
    if a[N] > b[N]:
        break
    N += 1

print("a"+str(N)+" =",a[N]," >  b"+str(N)+" =",b[N])
```

迴圈補充說明

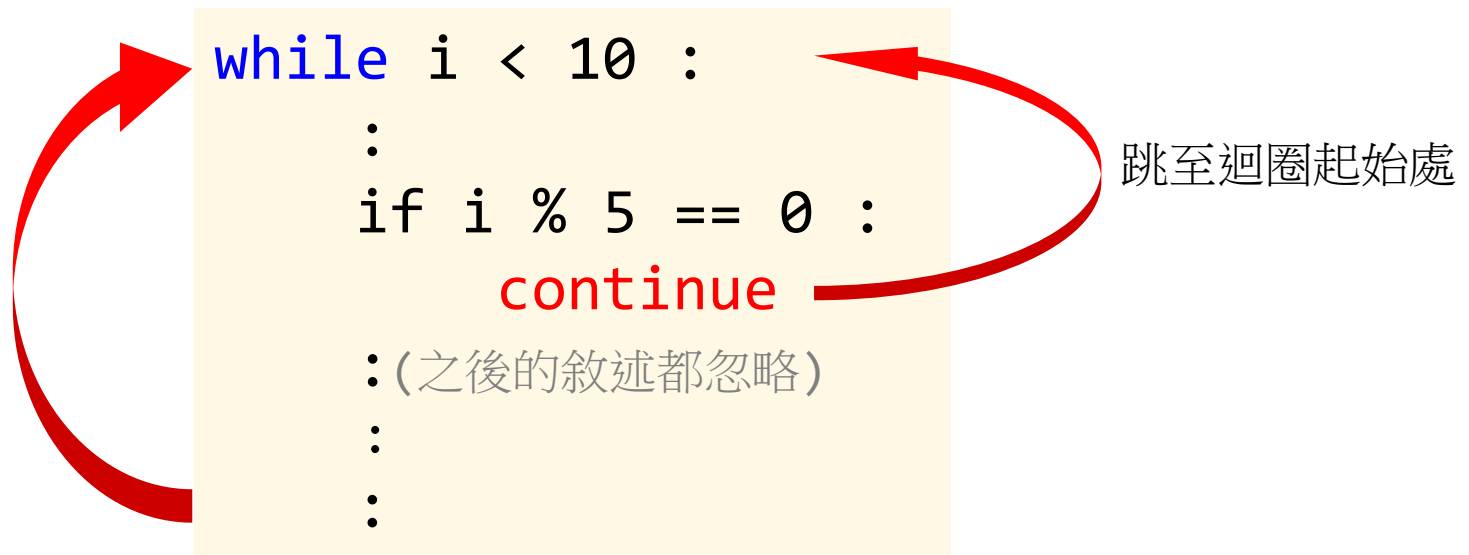
- ▶ **break**敘述：強制終止迴圈，跳到迴圈之後的敘述去執行。



跳出迴圈，執行迴圈外的下一道敘述

迴圈補充說明

- ▶ **continue**敘述：強制回到迴圈初始的地方，忽略之後的敘述。



休息一下~

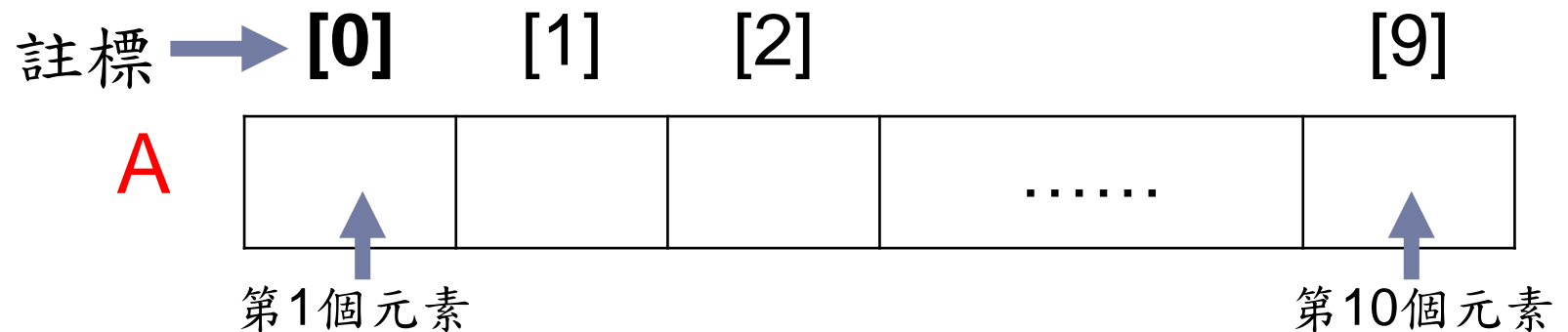


Ch5.陣列

- ▶ 在過去的章節，我們的做法是每次讀入一個資料，就立刻去處理，當讀入的資料一多就很不方便，應該先儲存至記憶體，以便後續處理，最普通的辦法是使用陣列。
- ▶ 目的：
 - ▶ 減少變數的個數,使程式的編寫更容易
 - ▶ 陣列可配合迴圈來處理大量資料

Ch5.陣列

- 下圖為一維陣列的概念，陣列名稱為A，依陣列的編號[n]來存取。



注意：陣列註標為正整數,而且是從0開始

Ch5.陣列

► 利用一維陣列求10個數字的計算平均值。

► 流程圖：

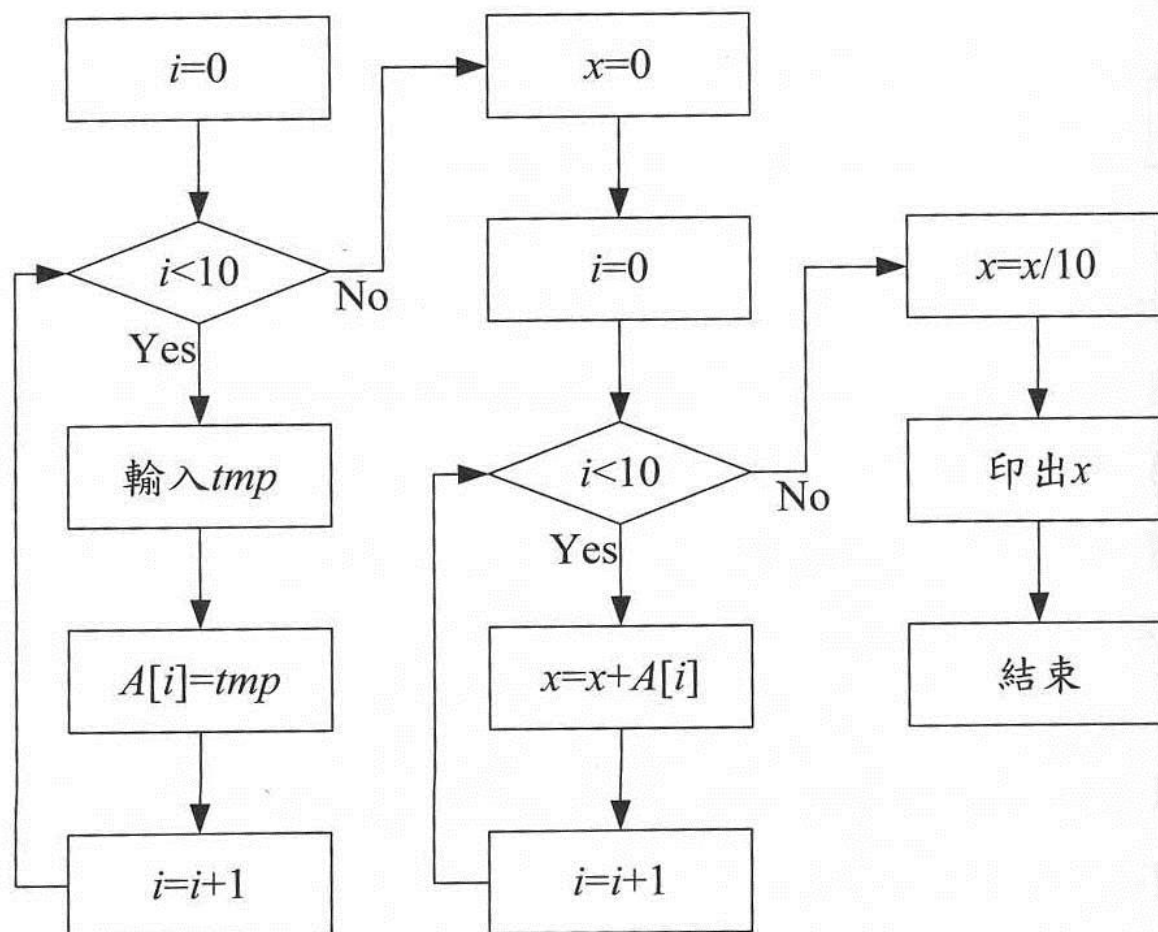


圖 5.3 ► 例題 5.1 的流程圖(求一個陣列中 10 個數字的平均數)

Ch5.陣列

▶ 參考程式：

```
A = [0 for i in range(10)]
```

宣告一維陣列A[10]，
並填入初值0

```
for i in range(10):  
    tmp = int(input("Please input the value: "))  
    A[i] = tmp
```

將輸入的數值存入陣列

```
x = 0  
for i in range(10):  
    x = x + A[i]
```

```
x = x / 10  
print("x = ",x)
```

將陣列裡的值依序加總

Ch5.陣列

► 利用一維陣列求10個數字的最大值。

► 流程圖：

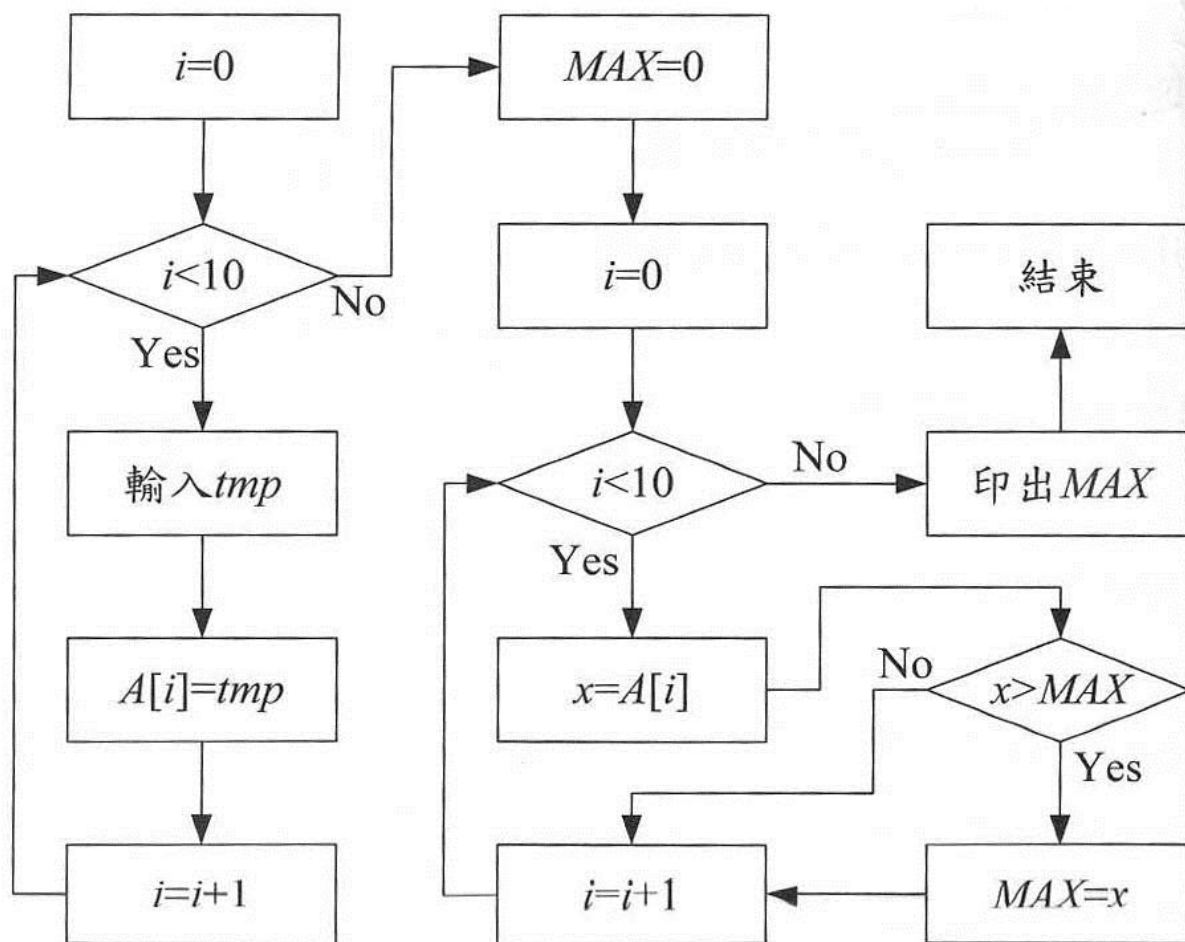


圖 5.4 ► 例題 5.2 的流程圖(求一個陣列中 10 個數字的最大)

Ch5.陣列

▶ 參考程式：

```
A = [0 for i in range(10)]
```

```
for i in range(10):  
    tmp = int(input("Please input the value: "))  
    A[i] = tmp
```

```
MAX = 0
```

```
for i in range(10):  
    x = A[i]  
    if x > MAX:  
        MAX = x
```

```
print("MAX = ",MAX)
```



Ch5.陣列

- ▶ 搜尋問題：輸入10個數字至 $A[]$ ，再輸入 x ，判斷 x 是否存在於 A 陣列中，如果存在，輸出所在的註標(索引)值，若不存在，則告知不存在。

- ▶ 流程圖：

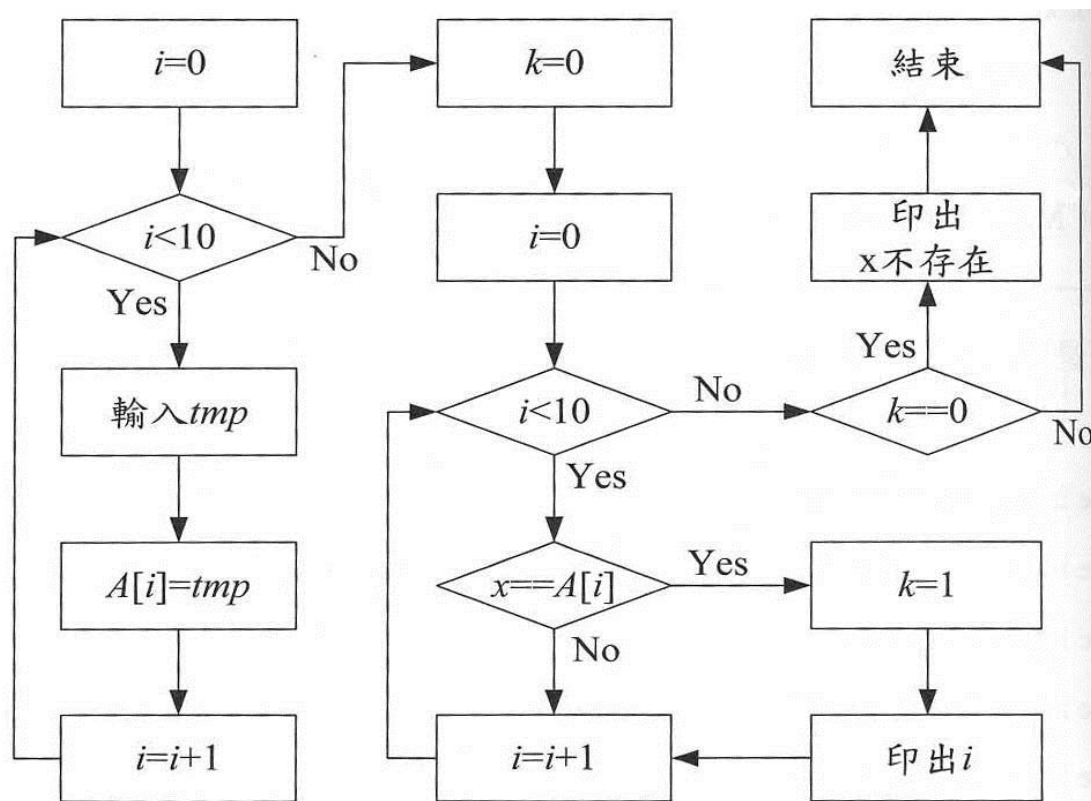


圖 5.5 ▶ 例題 5.3 的流程圖(搜尋問題)

Ch5.陣列

▶ 參考程式：

```
A = [0 for i in range(10)]
```

```
for i in range(10):  
    tmp = int(input("Please input the value: "))  
    A[i] = tmp
```

```
x = int(input("Please input the value youy want to search: "))  
k = 0
```

```
for i in range(10):  
    if x == A[i]:  
        k = 1  
        print("index = ",i)
```

```
if k == 0:  
    print(x, " does not exist")
```



Ch5. 陣列

- ▶ 二維陣列：使用兩個註標來宣告並使用陣列。
- ▶ 宣告並指定初值：

```
A = [[0 for i in range(4)] for j in range(3)]
```

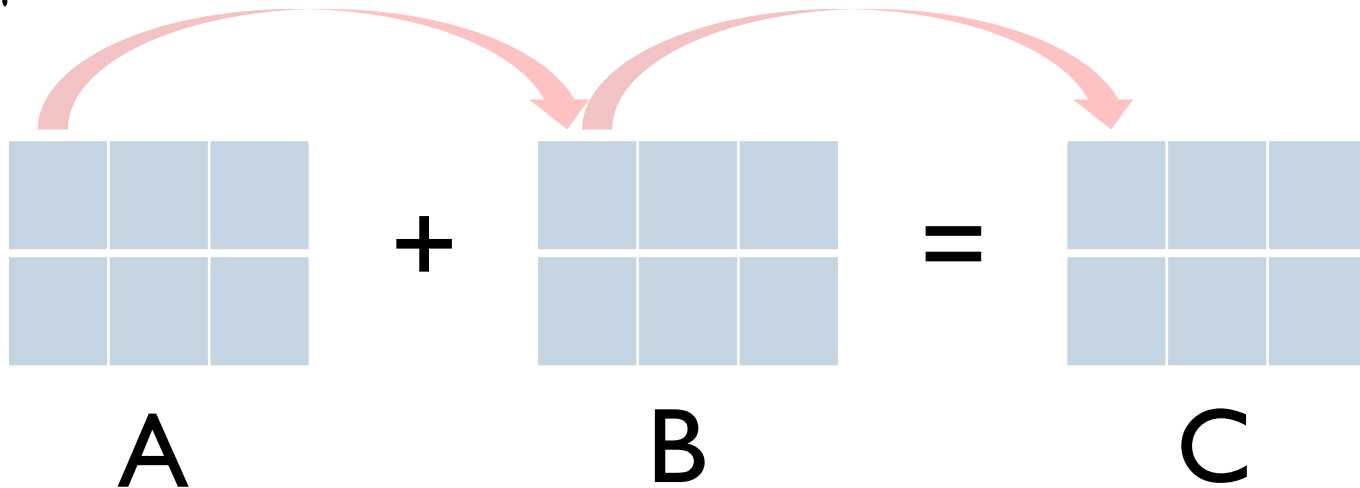
Diagram illustrating the mapping of the nested list comprehension to a 2D array structure:

	0	1	2	3
0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

The diagram shows a 3x4 grid of elements. The rows are indexed 0, 1, 2 and the columns are indexed 0, 1, 2, 3. The elements are labeled as A[row][column]. The row index is labeled 'row' and the column index is labeled 'column'.

Ch5.陣列

- ▶ 兩個二維矩陣相加。以陣列模擬矩陣，求兩個2乘3的二維矩陣相加之結果，第一個及第二個矩陣分別以A及B表示，相加之結果存入C矩陣，最後將C矩陣內容顯示出來。
- ▶ 圖示：



Ch5.陣列

► 流程圖：

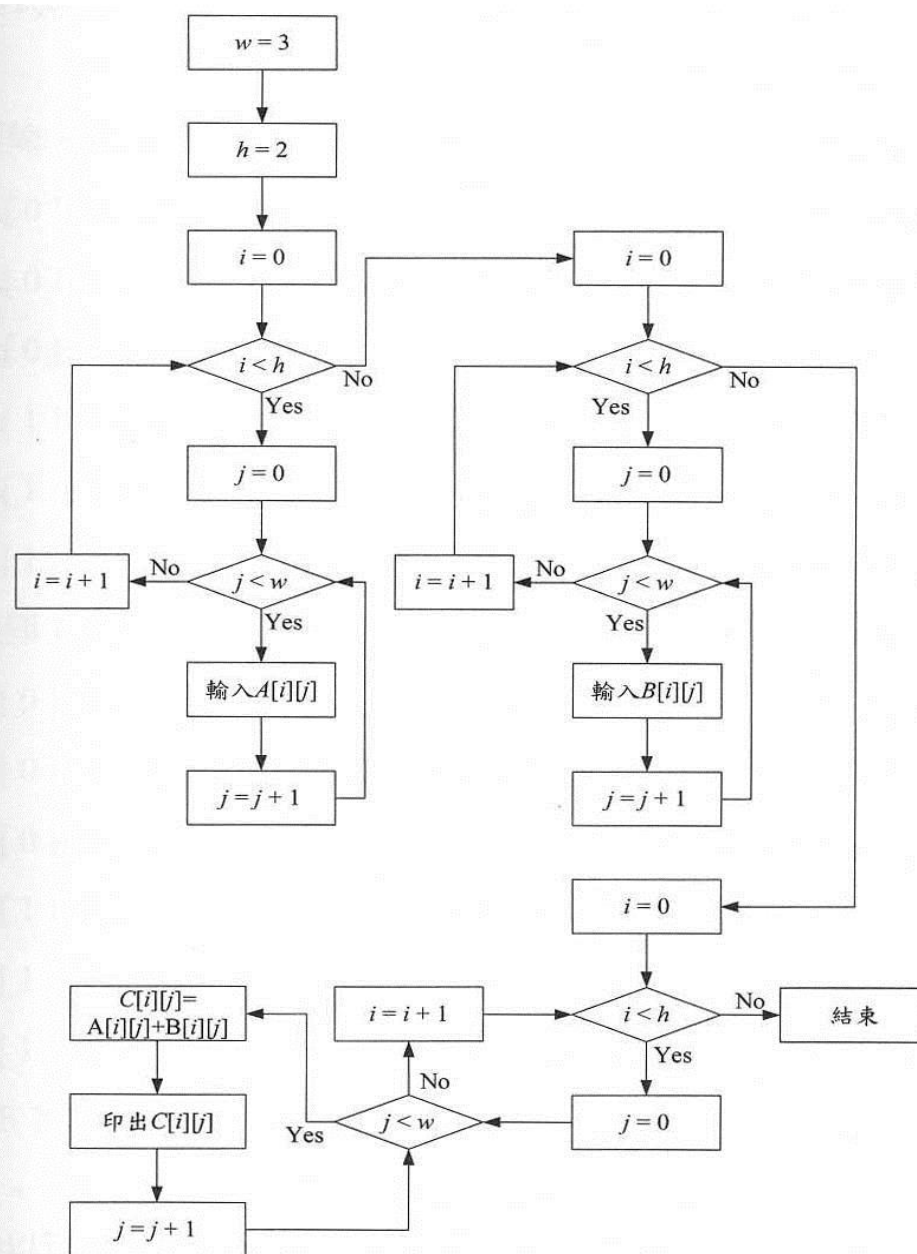


圖 5.6 ► 例題 5.4 的流程圖

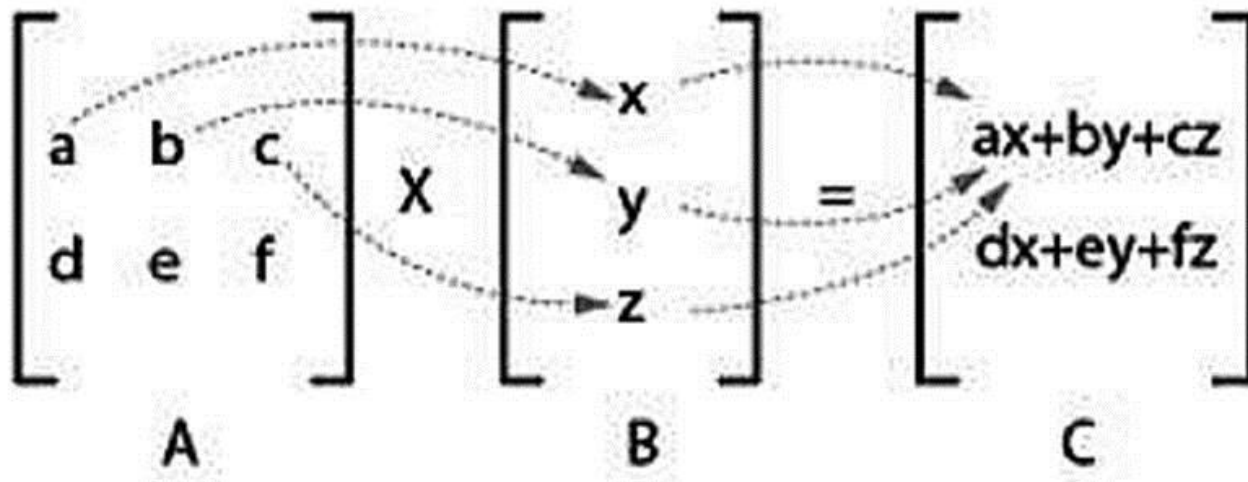
Ch5.陣列

▶ 參考程式：

```
w = 3
h = 2
#宣告A是一個h*w = 2*3的陣列
A = [[0 for i in range(w)] for j in range(h)]
print("請輸入矩陣A的值...")
for i in range(h):
    for j in range(w):
        print("請輸入A[" + i + "][" + j + "]的值: ", end=" ")
        A[i][j] = int(input())
#宣告B是一個h*w = 2*3的陣列
B = [[0 for i in range(w)] for j in range(h)]
print("請輸入矩陣B的值...")
for i in range(h):
    for j in range(w):
        print("請輸入B[" + i + "][" + j + "]的值: ", end=" ")
        B[i][j] = int(input())
print("矩陣A、B相加結果儲存於矩陣C中: ")
#宣告C是一個h*w = 2*3的陣列
C = [[0 for i in range(w)] for j in range(h)]
for i in range(h):
    for j in range(w):
        #計算A[i][j]+B[i][j]，將結果存入C[i][j]並印出
        C[i][j] = A[i][j] + B[i][j]
        print("C[" + i + "][" + j + "] = ", C[i][j])
```

Ch5.陣列

- ▶ 兩個矩陣相乘。求2乘3的A矩陣乘以3乘1的B矩陣，結果存入2乘1的C矩陣，最後將C矩陣的內容顯示出來。
- ▶ 圖示：



Ch5.陣列

▶ 參考程式：

```
#ha、wa代表陣列A的行列數目
ha = 2
wa = 3
#宣告A是一個ha*wa = 2*3的陣列
A = [[0 for i in range(wa)] for j in range(ha)]
print("請輸入矩陣A的值...")
for i in range(ha):
    for j in range(wa):
        print("請輸入A[" ,i, "][" ,j, "]的值: ",end=" ")
        A[i][j] = int(input())
#hb、wb代表陣列B的行列數目
hb = 3
wb = 1
#宣告B是一個hb*wb = 3*1的陣列
B = [[0 for i in range(wb)] for j in range(hb)]
print("請輸入矩陣B的值...")
for i in range(hb):
    for j in range(wb):
        print("請輸入B[" ,i, "][" ,j, "]的值: ",end=" ")
        B[i][j] = int(input())
print("矩陣A、B相乘結果儲存於矩陣C中: ")
```

Ch5.陣列

▶ 參考程式：

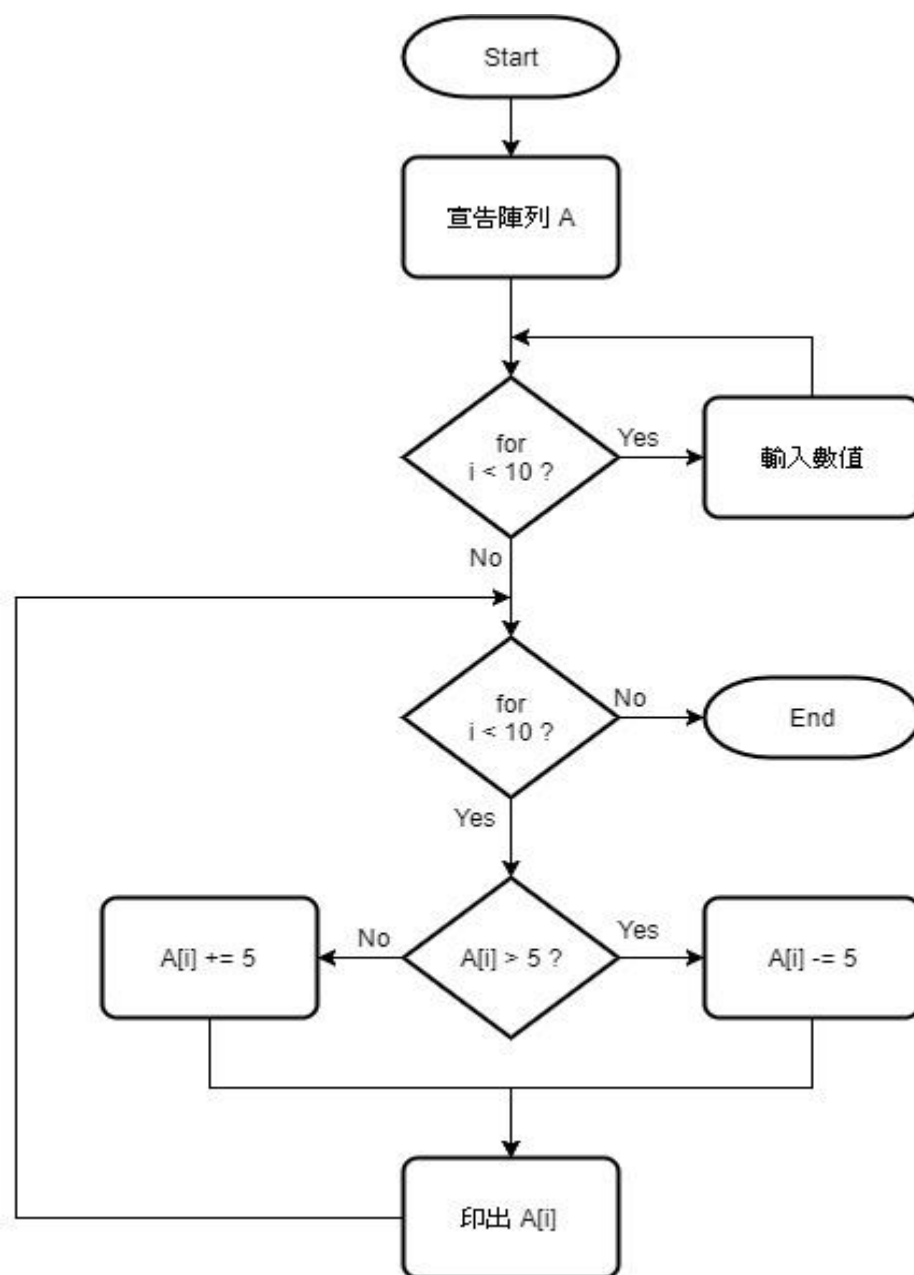
```
#宣告C是一個ha*wb = 2*1的陣列
C = [[0 for i in range(wb)] for j in range(ha)]
for i in range(ha):
    for j in range(wb):
        C[i][j] = 0
        for k in range(wa):
            #將A[i][k]*B[k][j]的結果累加入C[i][j]
            C[i][j] = C[i][j] + A[i][k] * B[k][j]
        #最後印出C[i][j]
        print("C[" + i + "][" + j + "] = " + C[i][j])
```

Ch5. 習題

- ▶ 5-1. 寫一程式，將10個數字讀入A陣列，然後逐一檢查此陣列，如 $A[i] > 5$ ，則令 $A[i] = A[i] - 5$ ，否則 $A[i] = A[i] + 5$ 。
- ▶ 5-2. 寫一程式，將10個數字讀入A陣列，對每一個數字，令 $A[i] = A[i] + i$ 。
- ▶ 5-3. 寫一程式，將10個數字讀入A陣列，並建立一個B陣列，如 $A[i] \geq 0$ ，令 $B[i] = 1$ ，否則令 $B[i] = 0$ 。
- ▶ 5-4. 寫一程式，將15數字存入 3×5 的二維陣列A中，求每一行及每一列數字的和。

Ch5. 習題

► 5-1 流程圖：



Ch5. 習題

► 5-1. 參考程式：

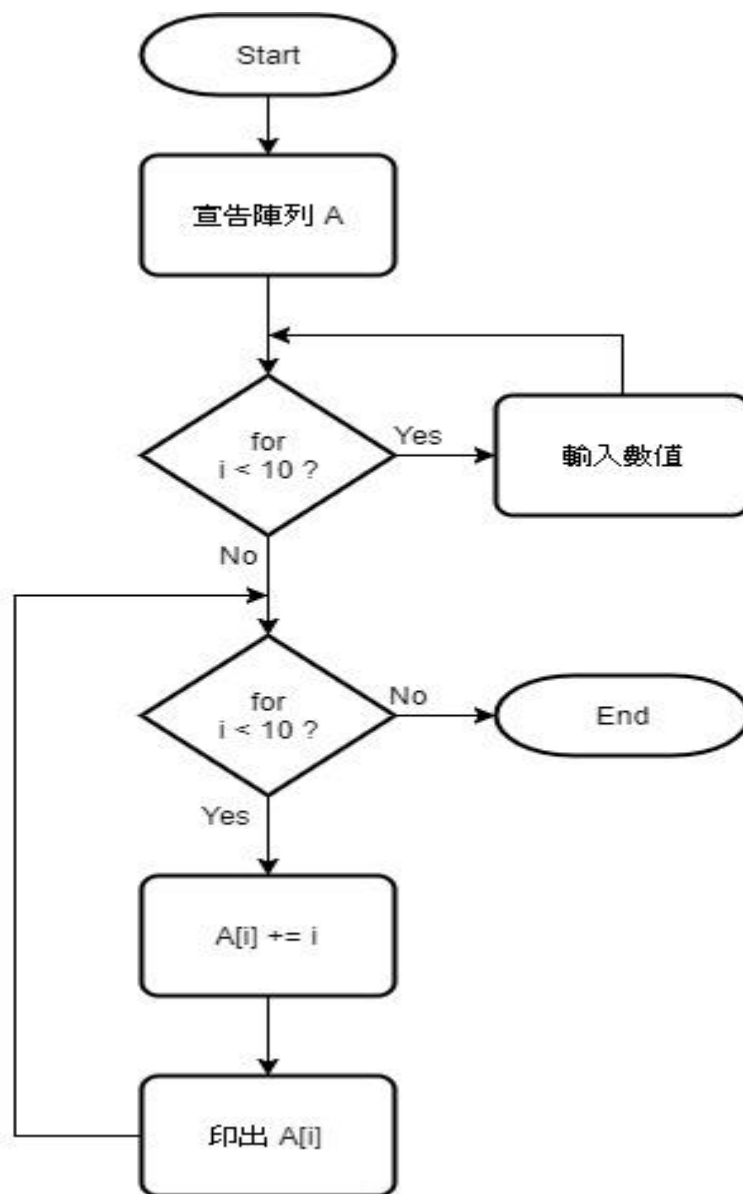
```
A = [0 for i in range(10)]

for i in range(10):
    A[i] = int(input("Please input the value: "))

for i in range(10):
    if A[i] > 5:
        A[i] -= 5
    else:
        A[i] += 5
    print("A[" + str(i) + "] = " + str(A[i]))
```


Ch5. 習題

► 5-2 流程圖：



Ch5. 習題

► 5-2. 參考程式：

```
A = [0 for i in range(10)]

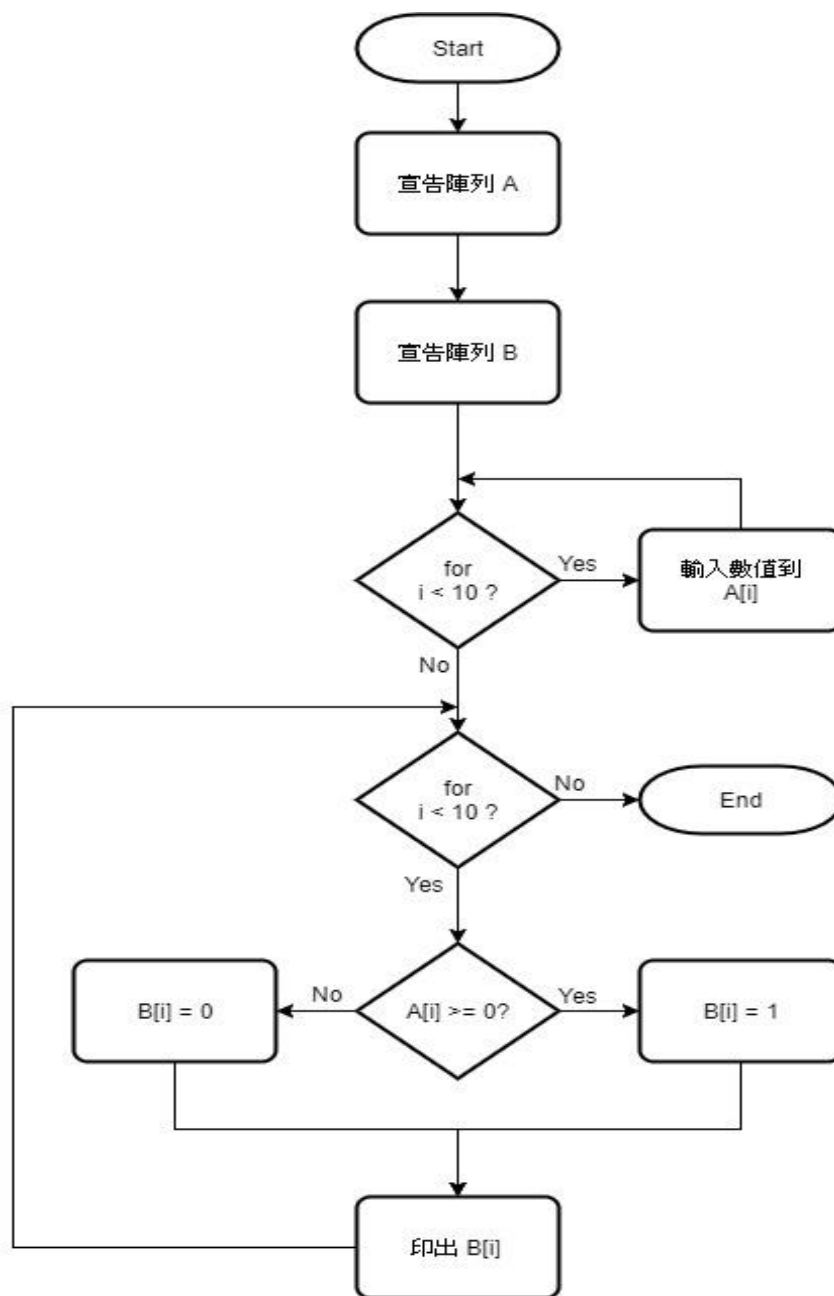
for i in range(10):
    A[i] = int(input("Please input the value: "))

for i in range(10):
    A[i] += i
    print("A[" + i + "] = ", A[i])
```



Ch5. 習題

► 5-3 流程圖：



Ch5. 習題

► 5-3. 參考程式：

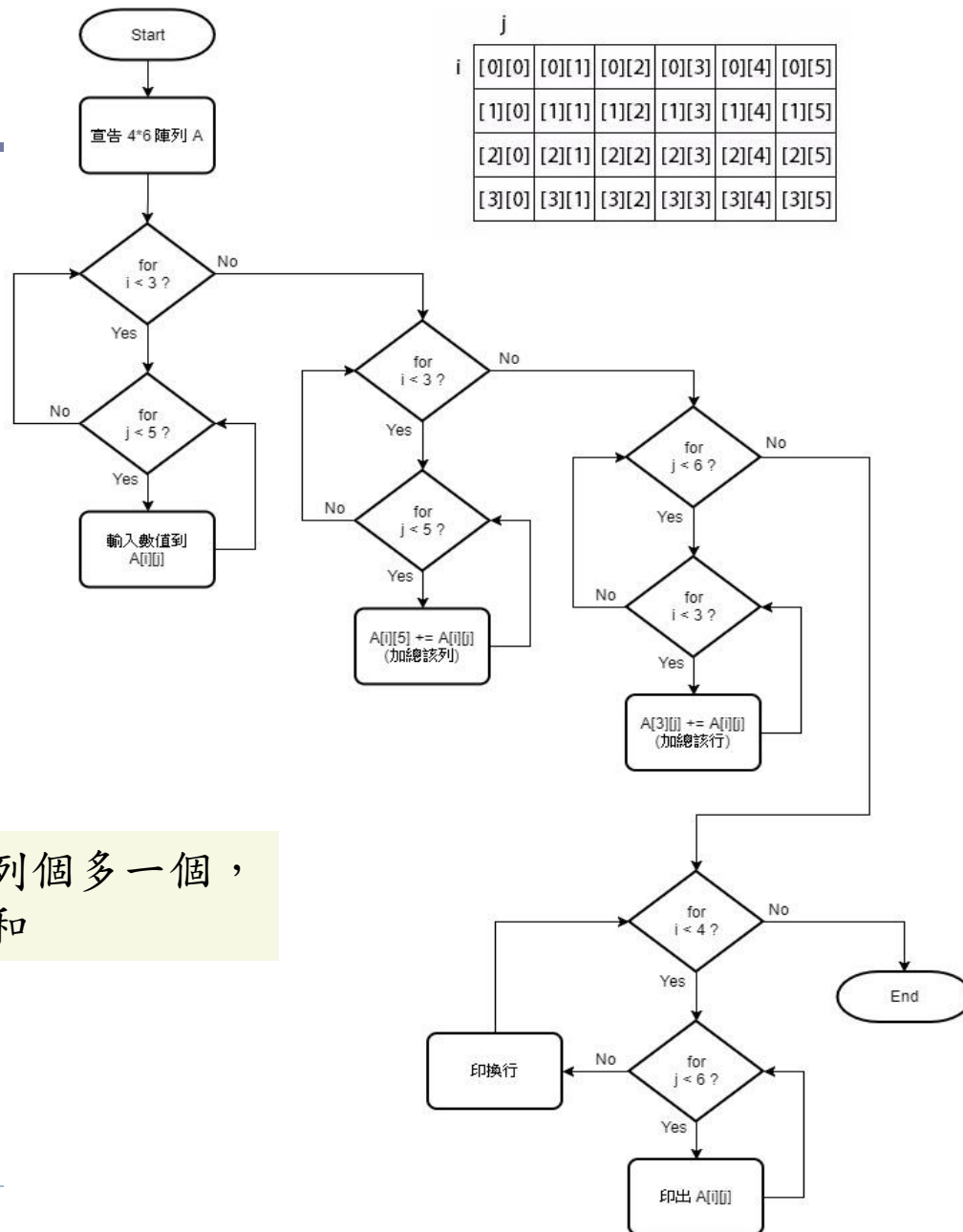
```
A = [0 for i in range(10)]
B = [0 for i in range(10)]

for i in range(10):
    A[i] = int(input("Please input the value: "))

for i in range(10):
    if A[i] >= 0:
        B[i] = 1
    else:
        B[i] = 0
    print("B[" + i + "] = " + str(B[i]))
```

Ch5. 習題

► 5-4 流程圖：



將陣列宣告時，行列個多一個，
用來儲存行和列的和

Ch5. 習題

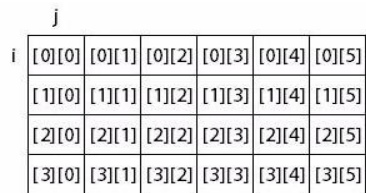
► 5-4. 參考程式：

```
A = [[0 for j in range(6)] for i in range(4)]
#請求輸入
for i in range(3):
    for j in range(5):
        print("Please input A[",i,""][",j,"]: ",end=" ")
        A[i][j] = int(input())
#計算列，將列的和存入每列最後一個元素
for i in range(3):
    for j in range(5):
        A[i][5] += A[i][j]
#計算行，將行的和存入每行最後一個元素
for j in range(6):
    for i in range(3):
        A[3][j] += A[i][j]
#印出矩陣和結果
for i in range(4):
    for j in range(6):
        print(A[i][j],"\t",end="")
    print()
```

Ch5. 習題

- ▶ 5-5. 寫一程式，將15數字存入 3×5 的二維陣列A中，求每一行及每一列數字的最小值。
- ▶ 5-6. 寫一程式，輸入兩組數字： a_1, a_2, \dots, a_5 和 b_1, b_2, \dots, b_5 。求 $a_i + b_i$ ， $i=1$ 到 $i=5$ 。
- ▶ 5-7. 寫一程式，輸入兩組數字： a_1, a_2, \dots, a_5 和 b_1, b_2, \dots, b_5 。令x為a中的最大值，令y為b中的最大值，求x與y中較小者。

► 5-5 流程圖：



Ch5. 習題

► 5-5. 參考程式：

```
A = [[0 for j in range(6)] for i in range(4)]
#請求輸入
for i in range(3):
    for j in range(5):
        print("Please input A[",i,""][",j,"]: ",end=" ")
        A[i][j] = int(input())

#尋找列，將列的最小值存入每列最後一個元素
for i in range(3):
    min = A[i][0]
    for j in range(1,5):
        if A[i][j] < min:
            min = A[i][j]
    A[i][5] = min
```



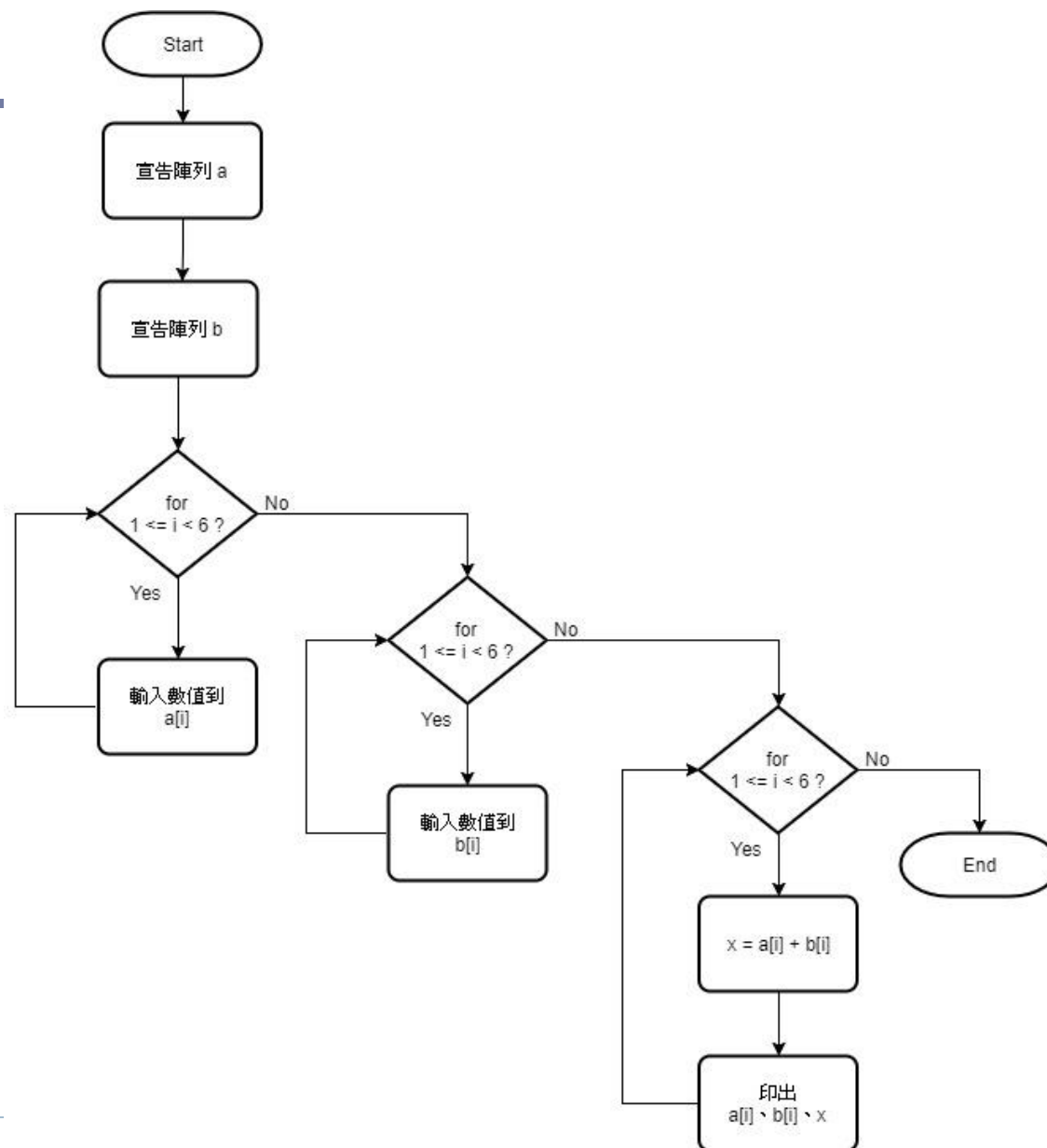
Ch5. 習題

► 5-5. 參考程式：

```
#尋找行，將行的最小值存入每行最後一個元素
for j in range(6):
    min = A[0][j]
    for i in range(1,3):
        if A[i][j] < min:
            min = A[i][j]
    A[3][j] = min
#印出矩陣和結果
for i in range(4):
    for j in range(6):
        print(A[i][j], "\t", end="")
    print()
```

Ch5. 習題

► 5-6 流程圖：



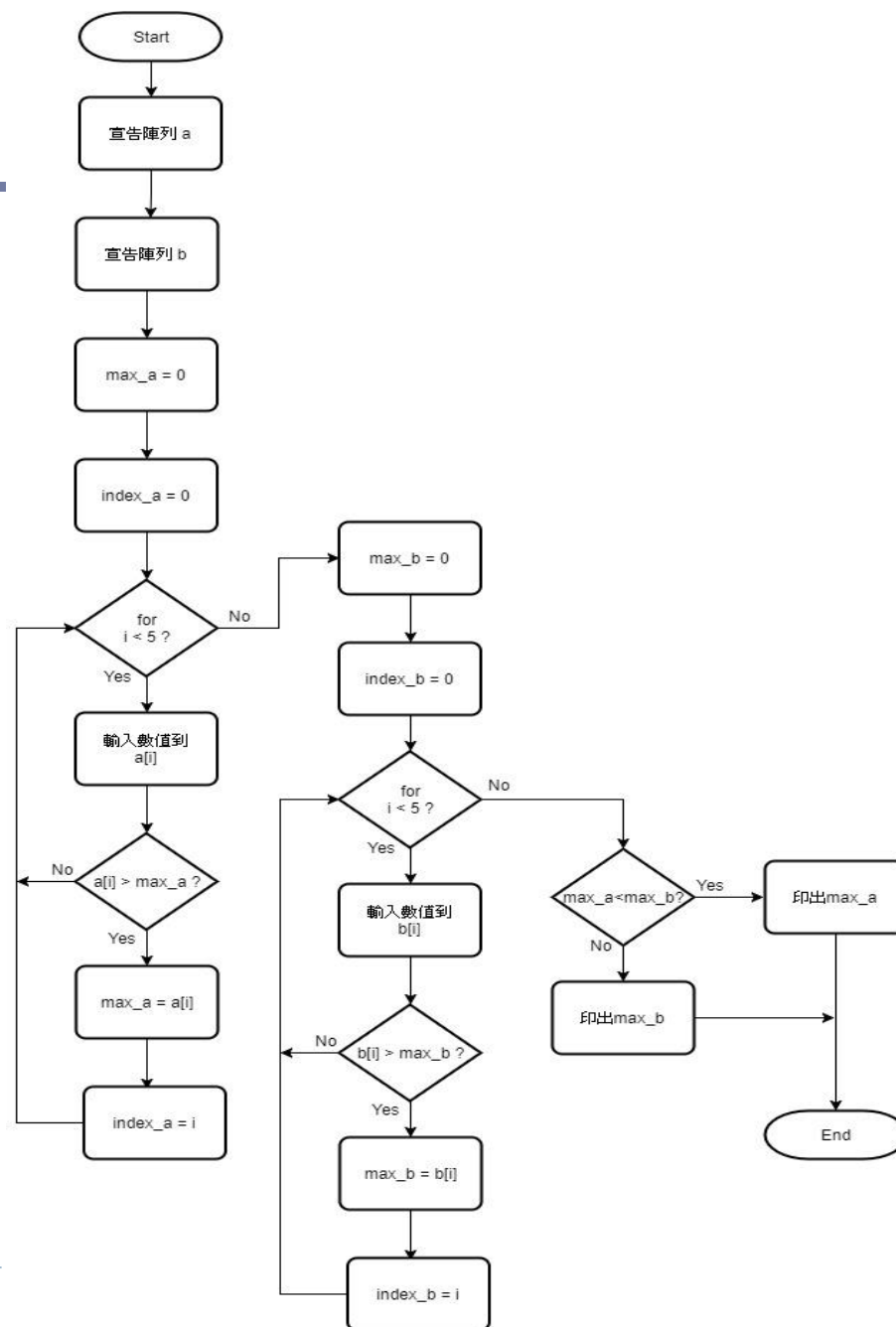
Ch5. 習題

► 5-6. 參考程式：

```
a = [0 for i in range(6)]
b = [0 for i in range(6)]
#輸入a陣列
for i in range(1,6):
    a[i] = int(input("Please input a["+str(i)+"]: "))
#輸入b陣列
for i in range(1,6):
    b[i] = int(input("Please input b["+str(i)+"]: "))
#計算a[i]+b[i]
for i in range(1,6):
    x = a[i] + b[i]
    print("a["+str(i)+"] + b["+str(i)+"] =",x)
```

Ch5. 習題

► 5-7 流程圖：



Ch5. 習題

► 5-7. 參考程式：

```
a = [0 for i in range(5)]
b = [0 for i in range(5)]
max_a = 0
index_a = 0
for i in range(5):    #輸入a陣列並記錄最大值
    a[i] = int(input("Please input a["+str(i)+"]: "))
    if a[i] > max_a:
        max_a = a[i]
        index_a = i
max_b = 0
index_b = 0
for i in range(5):    #輸入b陣列並記錄最大值
    b[i] = int(input("Please input b["+str(i)+"]: "))
    if b[i] > max_b:
        max_b = b[i]
        index_b = i
if max_a < max_b:      #顯示較小者
    print("a[",index_a,"] =",max_a)
else:
    print("b[",index_b,"] =",max_b)
```

休息一下~



Ch6. 副程式

- ▶ 將程式的步驟依功能分成幾個小部分，再由主程式去呼叫即可，可是較大的程式簡單明瞭。
- ▶ 程式中重複的段落可以寫成副程式的方式，代入不同的參數重複呼叫，可以精簡程式並容易維護。
- ▶ 有時亦稱函數
- ▶ 副程式需要傳回結果時用return指令。
- ▶ 宣告方式：

```
def 副程式名稱(參數1,參數2,...):  
    :  
    敘述  
    :
```


Ch6. 副程式

- ▶ 兩個一維陣列 相加：
 - ▶ 1.將一組數字讀入A陣列。
 - ▶ 2.將一組數字讀入B陣列。
 - ▶ 3.將A陣列與B陣列相加，成為C陣列，列印C陣列。

- ▶ 我們需要三個副程式：
 - ▶ 1.將數字讀入一個陣列的副程式。
 - ▶ 2.將兩個陣列相加的副程式。
 - ▶ 3.列印一個陣列的副程式。



Ch6. 副程式

- ▶ 1. 將數字讀入一個陣列的副程式。

```
def input_array(x,n):  
    print("---開始讀取陣列---")  
    i = 0  
    while i < n:  
        input_number = int(input("輸入數字: "))  
        x[i] = input_number  
        i += 1
```

- ▶ 副程式名稱為input_array，並需要傳入x及n兩個參數，x是陣列名稱，n是要輸入的數字的數目。



Ch6. 副程式

- ▶ 2. 將兩個陣列相加的副程式。

```
def add_array(x,y,z,n):  
    print("---開始陣列相加---")  
    i = 0  
    while i < n:  
        z[i] = x[i] + y[i]  
        i += 1
```

- ▶ 副程式名稱為add_array，並需要傳入x、y、z三個陣列名稱及數字的數目n。



Ch6. 副程式

- ▶ 3. 列印一個陣列的副程式。

```
def print_array(x,name,n):  
    print("---開始印出陣列---")  
    i = 0  
    while i < n:  
        print(name,"[",i,"]",x[i])  
        i += 1
```

- ▶ 副程式名稱為print_array，並需要傳入陣列名稱x，及要印出的陣列名稱的字母name，及數字的數目n。



Ch6. 副程式

▶ 主程式：

```
num = int(input("請輸入陣列大小:"))
```

```
#宣告陣列
```

```
A = [0 for i in range(num)]
```

```
B = [0 for i in range(num)]
```

```
C = [0 for i in range(num)]
```

```
#依序呼叫各個動作的副程式
```

```
input_array(A,num)
```

```
input_array(B,num)
```

```
add_array(A,B,C,num)
```

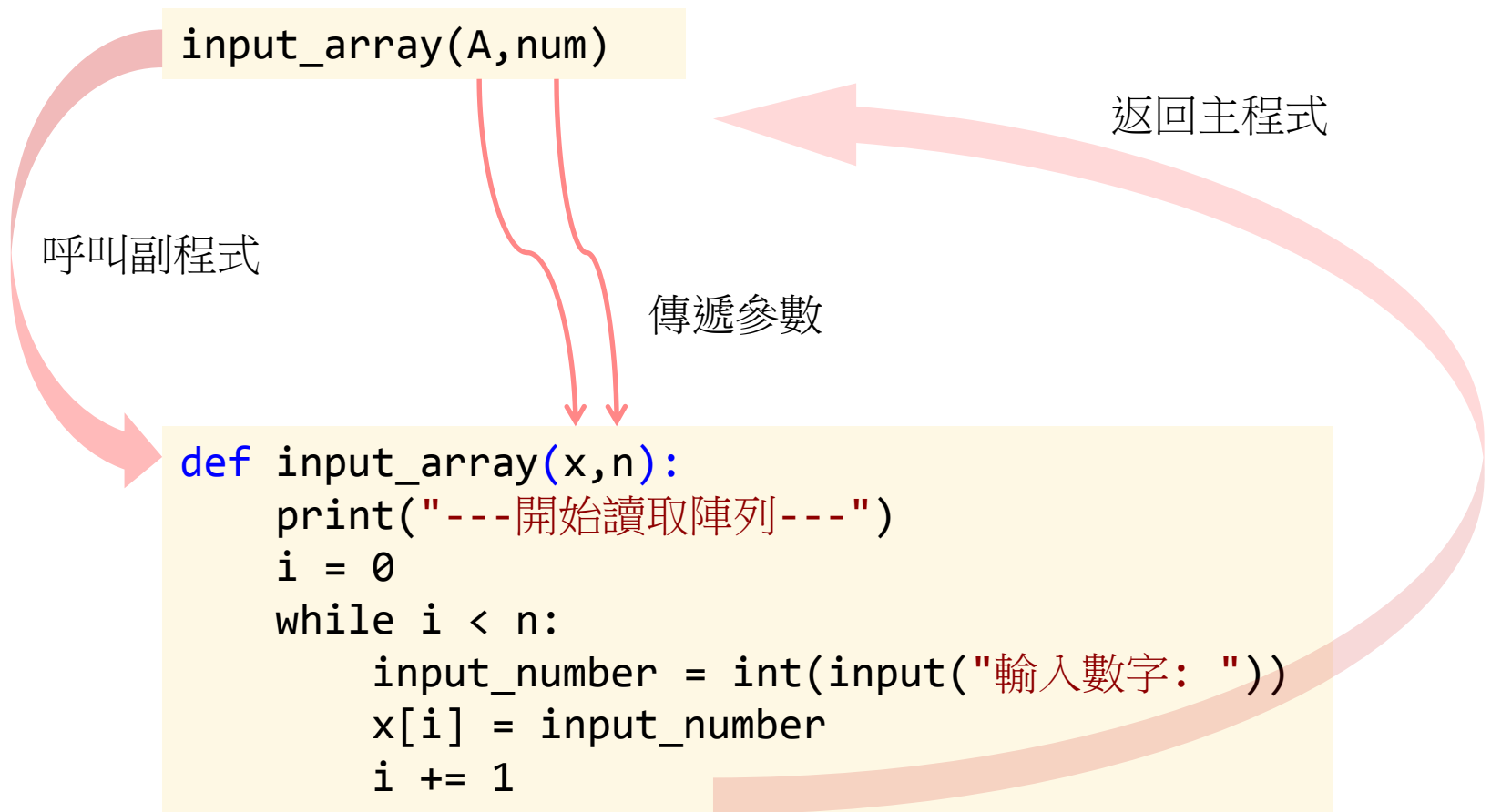
```
print_array(A,"A",num)
```

```
print_array(B,"B",num)
```

```
print_array(C,"C",num)
```

Ch6. 副程式

▶ 呼叫副程式方式：



Ch6. 副程式

► 完整程式：

```
def input_array(x,n):
    print("---開始讀取陣列---")
    i = 0
    while i < n:
        input_number = int(input("輸入數字: "))
        x[i] = input_number
        i += 1

def add_array(x,y,z,n):
    print("---開始陣列相加---")
    i = 0
    while i < n:
        z[i] = x[i] + y[i]
        i += 1

def print_array(x,name,n):
    print("---開始印出陣列---")
    i = 0
    while i < n:
        print(name,"[",i,"]",x[i])
        i += 1
```

```
#main
num = int(input("請輸入陣列大小:"))
A = [0 for i in range(num)]
B = [0 for i in range(num)]
C = [0 for i in range(num)]

input_array(A,num)
input_array(B,num)
add_array(A,B,C,num)
print_array(A,"A",num)
print_array(B,"B",num)
print_array(C,"C",num)
```

Ch6. 副程式

- ▶ 求陣列中最大之數：將5個數字讀入陣列，求陣列中最大之數。我們需要兩個副程式。
 - ▶ 1.將5個數字讀入一個陣列的副程式。
 - ▶ 2.求這個陣列中數字最大值的副程式。
- ▶ 這裡使用return敘述將結果回傳給呼叫它的敘述，所以呼叫這個副程式時需準備接收傳回值。
- ▶ Python可以直接傳遞陣列，與傳遞變數相同。
- ▶ len()函數會傳回陣列或字串的長度(個數)



Ch6. 副程式

▶ 參考程式：

```
def createArray():  
    A = [0 for i in range(5)]  
    for i in range(5):  
        x = int(input("Input number:"))  
        A[i] = x  
    return A
```

```
def getMax(A):  
    max = 0  
    for i in range(len(A)):  
        if A[i] > max:  
            max = A[i]  
    return max
```

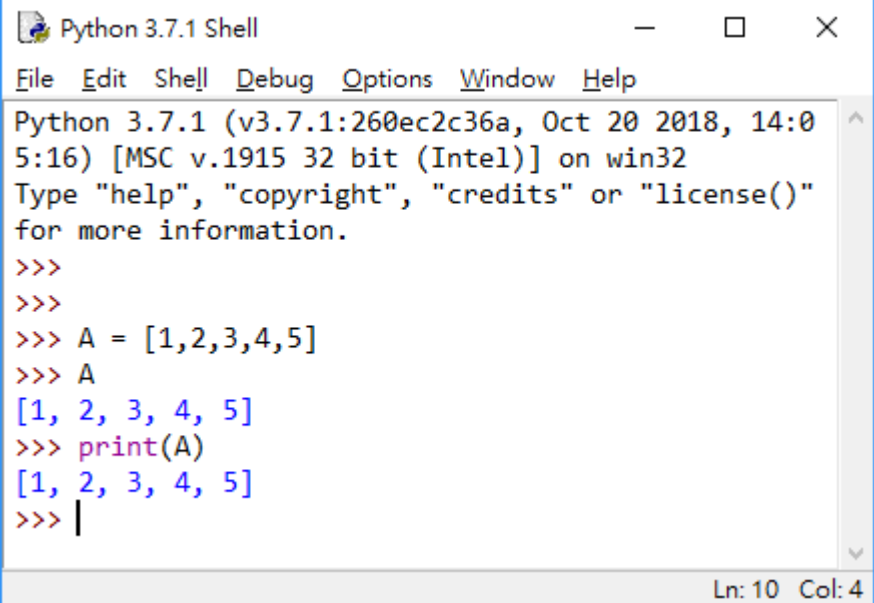
```
#main  
array = createArray()  
print(getMax(array))
```

Ch6. 副程式

▶ Array + 1 :

- ▶ 1.讀入5個數字，將這5個數字放入一個陣列中。
- ▶ 2.對每一個陣列中的數字都加1，然後列印出來。

▶ Python使用print()可以直接列印陣列，不須使用迴圈。



```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> A = [1,2,3,4,5]
>>> A
[1, 2, 3, 4, 5]
>>> print(A)
[1, 2, 3, 4, 5]
>>> |
```

Ln: 10 Col: 4

Ch6. 副程式

▶ 參考程式：

```
def createArray():  
    A = [0 for i in range(5)]  
    for i in range(5):  
        x = int(input("Input number:"))  
        A[i] = x  
    return A
```

```
def addition(A):  
    for i in range(len(A)):  
        A[i] += 1  
    return A
```

```
#main  
array = createArray()  
print("After addition:", addition(array))
```

Ch6. 副程式

- ▶ 求小於 N 的所有質數。
 - ▶ 質數 (Prime number)，又稱素數，指在大於1的自然數中，除了1和該數自身外，無法被其他自然數整除的數
 - ▶ 例如假設 $N=20$ ，則小於20的質數有1,2,3,5,7,11,13,17,19。
- ▶ 我們需要一個副程式，這個副程式輸入一個正整數 x ，然後判斷 x 是否是一個質數。



Ch6. 副程式

► 流程圖：

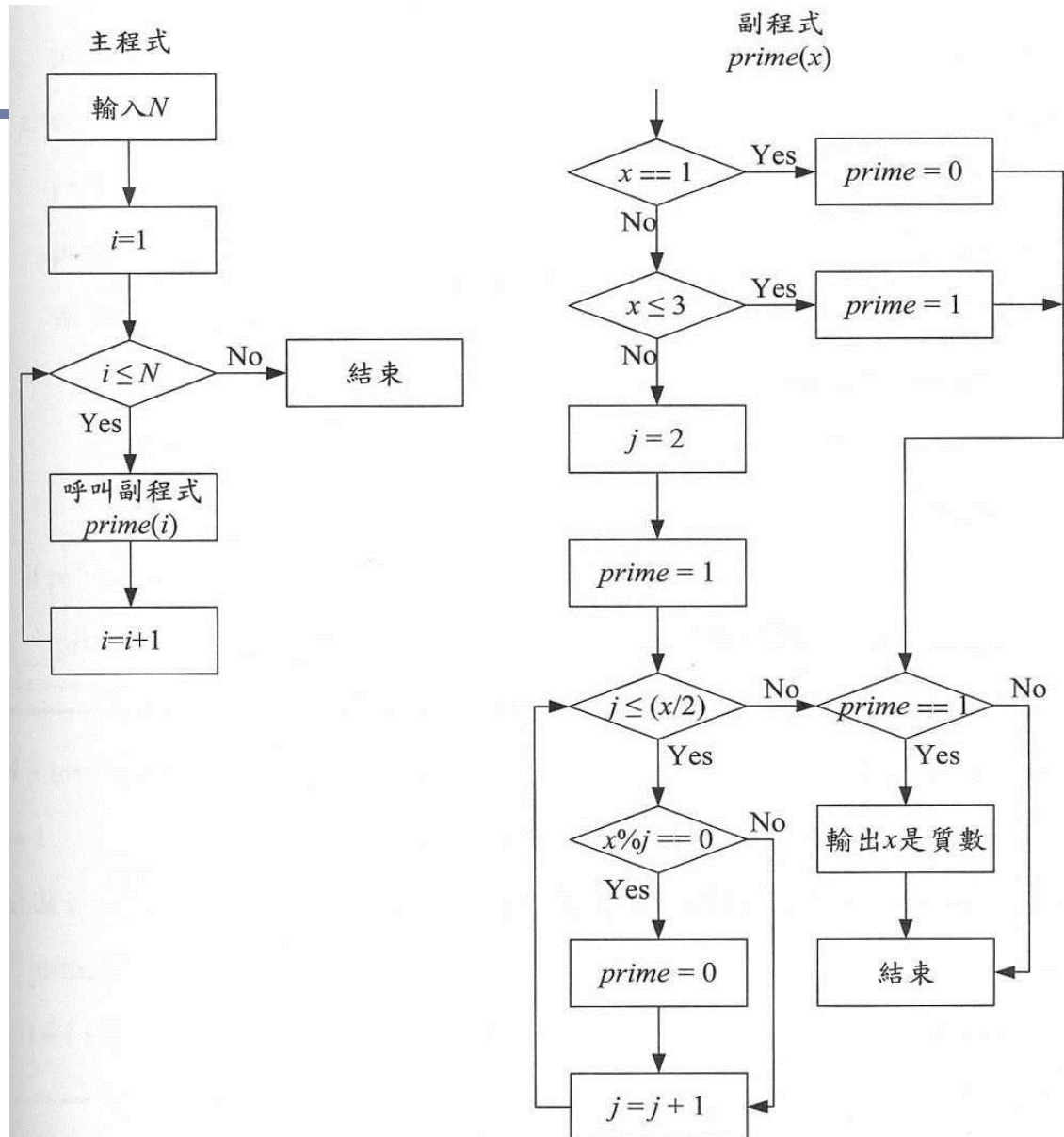


圖 6.1 ► 例題 6.4 的流程圖(決定 x 是否為質數)

Ch6. 副程式

▶ 參考程式：

```
def prime(x):  
    if x == 1:  
        prime = 0  
    elif x <= 3:  
        prime = 1  
    else:  
        j = 2  
        prime = 1  
        while j <= (x / 2):  
            if x % j == 0:  
                prime = 0  
                j += 1  
        if prime == 1:  
            print(x, "是一個質數")  
  
#main  
N = int(input("請輸入一個正整數N:"))  
i = 1  
while i <= N:  
    prime(i)  
    i += 1
```

1不列入質數

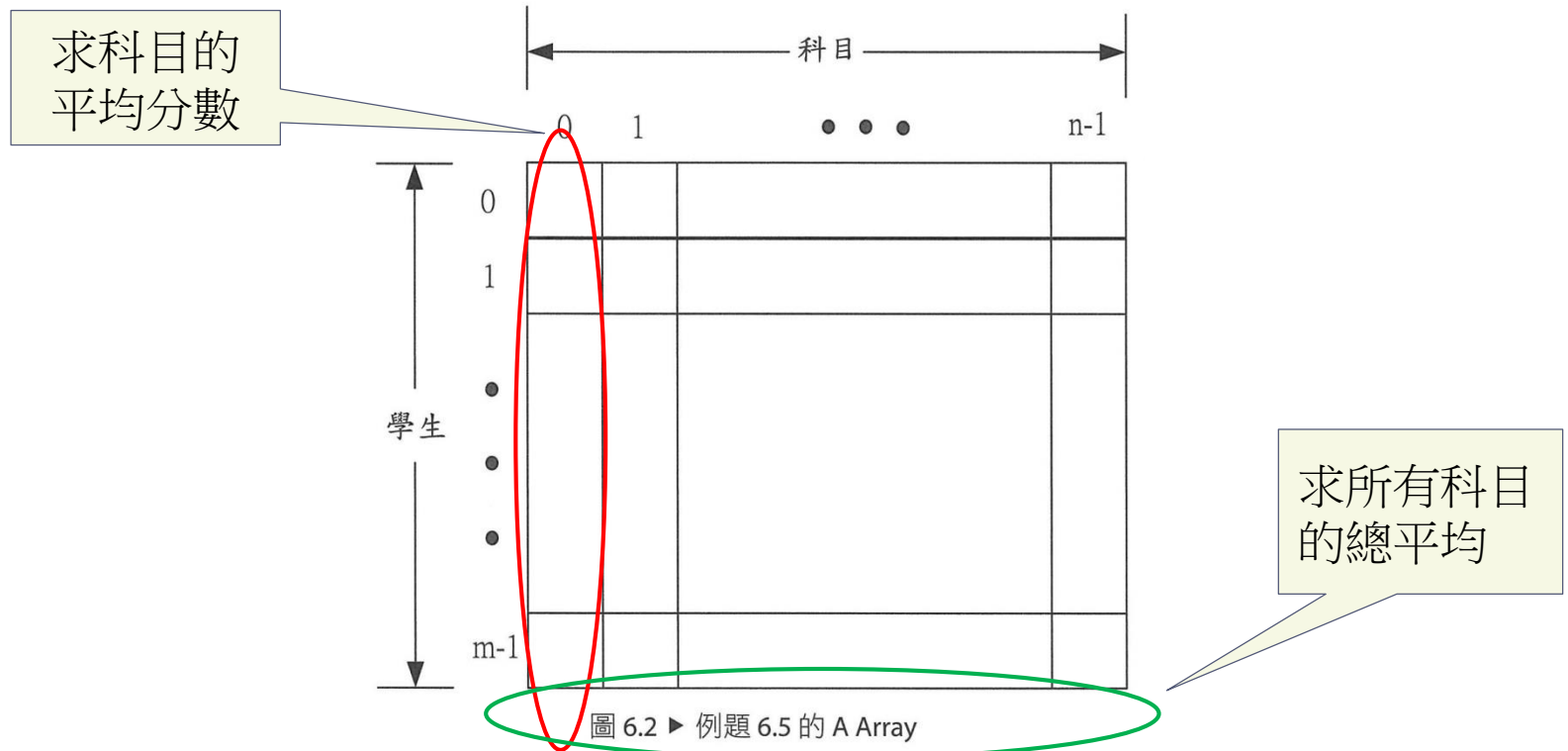
2,3是質數，不用計算

算到x的1/2即可，
再加上去也不可能
整除x了

不能輸入0或1，
會沒有輸出訊息

Ch6. 副程式

- ▶ 求出各科平均分數及總平均分數。
- ▶ 我們有 m 個學生，每一個學生有 n 門課，我們的任務是對每一門課，求這門課的平均分數，然後求所有科目的總平均。



Ch6. 副程式

▶ 參考程式：

```
def subjectAverage(i,m,Array): #求第i科平均成績
    sum = 0
    for j in range(m):
        sum += Array[j][i]
    sum /= m
    print("第",i+1,"科平均為:",sum)
    return sum

def getAverage(m,n,Array): #求各科總平均
    sum = 0
    for i in range(n):
        sum = sum + subjectAverage(i,m,Array)
    sum /= n
    print("總平均:",sum)

def inputArray(n): #輸入某位學生n科成績
    B = [0 for i in range(n)]
    for i in range(n):
        print("輸入第",i+1,"科:")
        x = int(input())
        B[i] = x
    return B
```


Ch6. 副程式

▶ 參考程式：

```
def input2DArray(m,n): #輸入m個學生的n科成績
    A = [[0 for i in range(n)] for j in range(m)]
    for i in range(m):
        print("--請輸入第",i+1,"位學生的成績--")
        B = inputArray(n)
        A[i] = B
    return A

#main
m = int(input("請輸入學生數目:"))
n = int(input("請輸入科目數:"))
Array = input2DArray(m,n)
getAverage(m,n,Array)
```

▶ 副程式裡可以再呼叫其他副程式，也可以呼叫自己(這情況稱遞迴(Recursive))

Ch6. 副程式

- ▶ 判斷日期先後順序。
 - ▶ 輸入日期A與日期B，若：
 - ▶ 日期A在日期B之前，輸出「日期A在日期B之前」。
 - ▶ 日期A在日期B之後，輸出「日期A在日期B之後」。
 - ▶ 日期A在日期B相同，輸出「日期A在日期B相同」。
- ▶ 假設輸入格式為(日/月/年)「21/01/2018」存入變數temp，則使用split()方法可將字串拆開。

```
date = temp.split("/")
```
- ▶ 以「/」為分隔，拆成三組字串，存入date陣列，使得date[0]="21"，date[1]="01"，date[2]="2018"

Ch6. 副程式

► 流程圖：

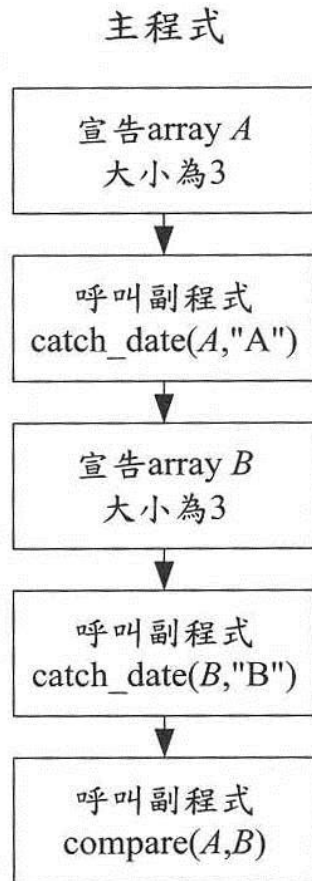


圖 6.3 ► 例題 6.6 主程式的流程圖

副程式 *catch_date(X, name)*

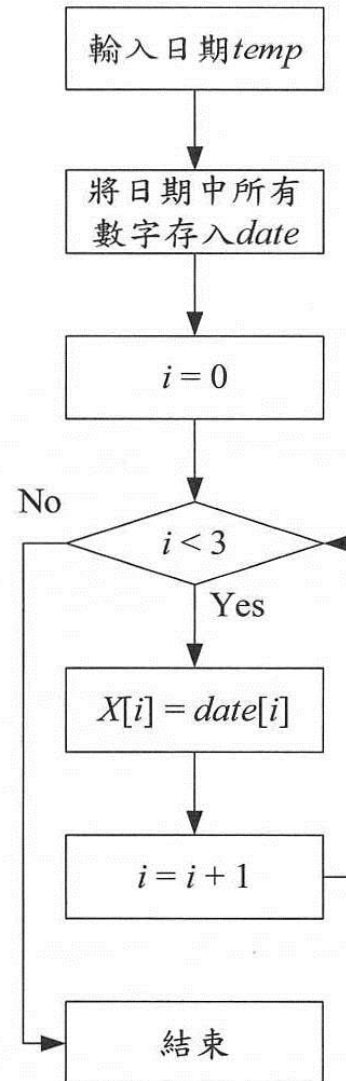


圖 6.4 ► 例題 6.6 副程式 catch-date 的流程圖

Ch6. 副程式

► 流程圖：

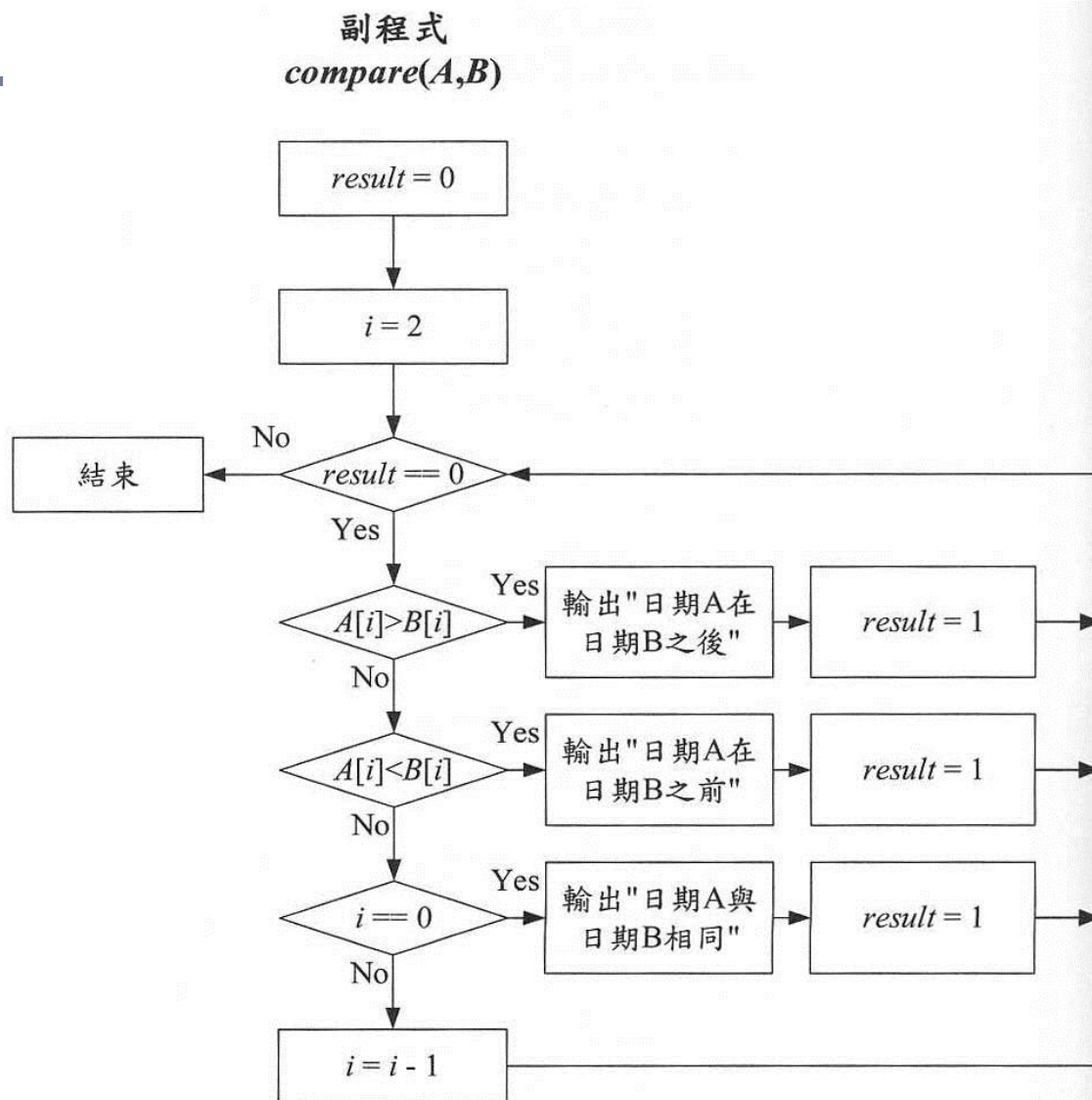


圖 6.5 ► 例題 6.6 副程式 *compare* 的流程圖

Ch6. 副程式

▶ 參考程式：

```
#main
A = [0 for i in range(3)]
catch_date(A, "A")
B = [0 for i in range(3)]
catch_date(B, "B")
compare(A,B)
```

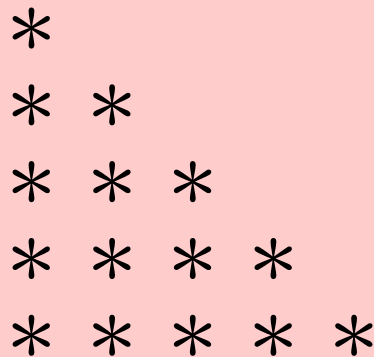
- ▶ 程式中副程式要寫在前面，最後再寫主程式。

```
def catch_date(X,day): #將日期分開存入陣列副程式
    print("請輸入日期",day,"(日/月/年):",end="")
    temp = input()
    date = temp.split("/")
    for i in range(3):
        X[i] = int(date[i])

def compare(A,B): #比較日期副程式
    result = 0
    i = 2
    while result == 0:
        if A[i] > B[i]:
            print("日期A在日期B之後")
            result = 1
        elif A[i] < B[i]:
            print("日期A在日期B之前")
            result = 1
        elif i == 0:
            print("日期A與日期B相同")
            result = 1
    else:
        i -= 1
```

Ch6. 副程式

- ▶ 輸出「*」三角形，如圖。
- ▶ 印完後要顯示"Enter x to quit, or any other key to continue:"，如果使用者按X鍵，程式就要停止，其他任何鍵就會再印一次星號



```
*  
* *  
* * *  
* * * *  
* * * * *
```

Ch6. 副程式

► 流程圖：

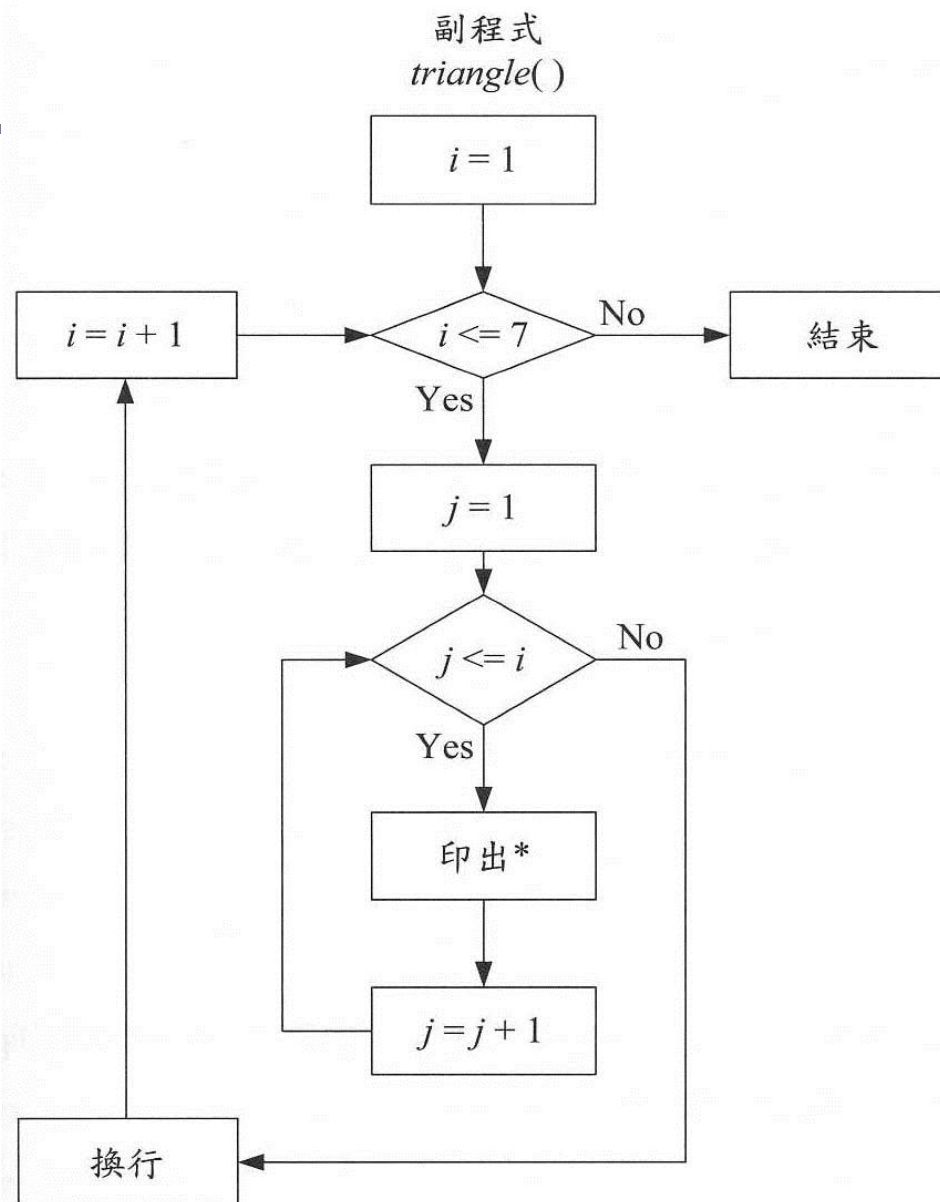


圖 6.6 ► 例題 6.7 副程式的流程圖

Ch6. 副程式

▶ 參考程式：

```
def triangle():
    for i in range(7):
        for j in range(i):
            print("*",end="")
        print("")

#main
triangle()
ch = input("Enter x to quit, or any other key to continue:")
while ch != "x":
    triangle()
    ch = input("Enter x to quit, or any other key to continue:")
```

▶ print("")會單純的換行而已



Ch6.副程式

- ▶ 密文解碼。
- ▶ 為了保密，可以將原文加密，加密的方法有很多種，最簡單的英文加密方法是將每一個字母做位移，假設只考慮26個英文大寫字母，且位移規則如下：

原來字母	A	B	C	...	X	Y	Z
轉換後字母	D	E	F	...	A	B	C

小寫字母規則亦同。可以看出皆是往後移三個字母。

- ▶ 還原的動作叫解密，假設加密後的文字為DSSOH和CRR，解密後的文字就是APPLE和ZOO。

Ch6.副程式

► 流程圖：

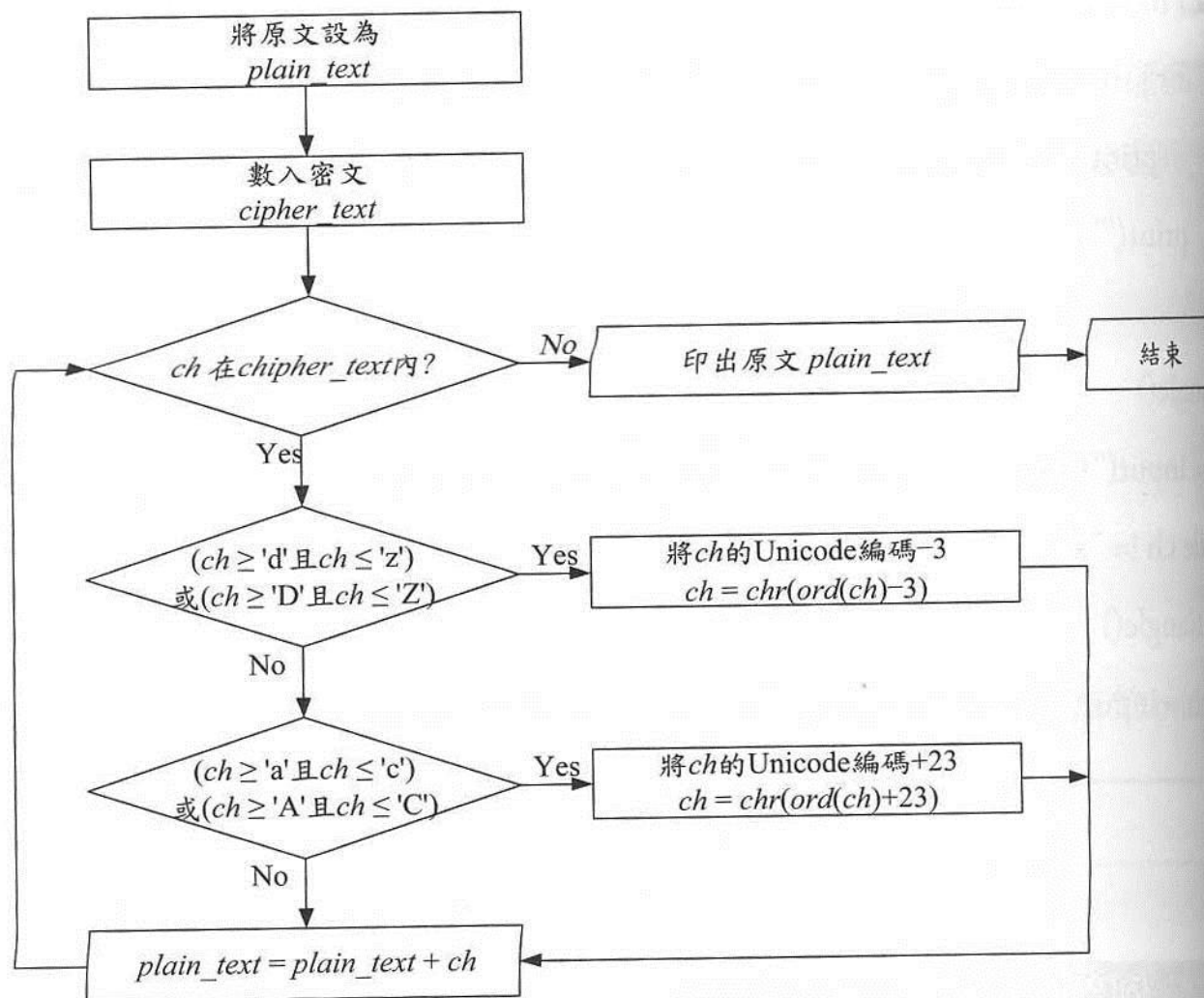


圖 6.7 ► 例題 6.8 的流程圖

Ch6. 副程式

▶ 參考程式：

```
plain_text = ""
cipher_text = input("請輸入密文:")

for ch in cipher_text:
    if (ch>="d" and ch<="z") or (ch>="D" and ch<="Z"):
        ch = chr(ord(ch) - 3)
    elif (ch>="a" and ch<="c") or (ch>="A" and ch<="C"):
        ch = chr(ord(ch) + 23)
    plain_text += ch

print("原文為:" + plain_text)
```



Ch6. 副程式

- ▶ 說明：
- ▶ 在ASCII碼中，字母A的編碼為 65_{10} ，則函數

`ord("A")` 會傳回A的編碼65

`chr(65)` 會傳回字母A

- ▶ 所以字母也可以加或減，會得到另一個字母。

Ch6. 習題

- ▶ 6-1. 在主程式中接受使用者輸入梯形之上底、下底與高的值，並呼叫一副程式，將之上底、下底與高的值傳入該副程式後，在副程式中計算並印出梯形面積的值。
- ▶ 6-2. 寫一副程式，接受主程式傳進的陣列與陣列大小，分別計算陣列上第奇數個元素與第偶數個元素之平均值並印出。
- ▶ 6-3. 在主程式中接受使用者輸入A、B、C的值並呼叫一副程式，將A、B、C的值傳入副程式中，在副程式中判斷 $|A|$ 、 $|B|$ 、 $|C|$ 之大小順序，回傳絕對值最大者並在主程式中印出， $|A|$ 、 $|B|$ 、 $|C|$ 之最大值。

Ch6. 習題

▶ 6-1. 參考程式：

```
def trapezoid(A,B,H):  
    return ((A + B)* H) / 2  
  
a = float(input("請輸入上底:"))  
b = float(input("請輸入下底:"))  
h = float(input("請輸入高:"))  
print("此梯形面積為:",trapezoid(a,b,h))
```

- ▶ 副程式 `trapezoid()` 的傳回值直接交給 `print()` 列印。

Ch6. 習題

► 6-2. 參考程式：

```
def averageElement(X,L):
    even = 0
    odd = 0
    c1 = 0
    c2 = 0
    for i in range(1,L):
        if i % 2 == 0:
            even += X[i]
            c2 += 1
        else:
            odd += X[i]
            c1 += 1
    print("奇數元素平均:",odd/c1)
    print("偶數元素平均:",even/c2)

#main
L = int(input("請輸入有多少個元素:")) + 1
A = [0 for i in range(L)]
for i in range(1,L):
    print("請輸入元素",i,":",end="")
    A[i] = int(input())
averageElement(A,L)
```

Ch6. 習題

▶ 6-3. 參考程式：

```
def maxABS(a,b,c):
```

```
    a = abs(a)
```

```
    b = abs(b)
```

```
    c = abs(c)
```

```
    if a > b:
```

```
        max = a
```

```
    else:
```

```
        max = b
```

```
    if c > max:
```

```
        max = c
```

```
    return max
```

```
#main
```

```
A = int(input("Please input number A:"))
```

```
B = int(input("Please input number B:"))
```

```
C = int(input("Please input number C:"))
```

```
print("Max of absolute value is:",maxABS(A,B,C))
```

abs() 函數會傳回數值的絕對值

Ch6. 習題

- ▶ 6-4. 在主程式中接受使用者輸入首項 a_1 ，公比 r 與項數 n ，呼叫一副程式並將 a_1 、 r 、 n 的值傳入，在副程式中計算等比級數第 n 項的值並回傳，在主程式中印出該值。
- ▶ 6-5. 在主程式中接受使用者輸入一個陣列的值，將陣列的值與陣列的大小傳入一副程式中，此副程式將會計算該陣列之中位數並回傳，主程式在收到此副程式的回傳值之後印出。

Ch6. 習題

► 6-4. 參考程式：

```
def progression(a1,r,n):  
    return a1*(r**(n-1))  
  
#main  
A1 = int(input("請輸入首項："))  
R = int(input("請輸入公比："))  
N = int(input("請輸入項數："))  
print(progression(A1,R,N))
```

► 等比級數第n項的值為：

$$a_n = a_1 r^{n-1}$$

Ch6. 習題

▶ 6-5. 參考程式：

```
def median(x,L):  
    x.sort()    #先將陣列排序  
    if L % 2 != 0:  
        return x[int(L/2)]  
    else:  
        return (x[int(L/2-1)] + x[int(L/2)]) / 2  
  
#main  
L = int(input("請輸入陣列長度:"))  
X = [0 for i in range(L)]  
for i in range(L):  
    print("請輸入數值",i+1,":",end="")  
    X[i] = int(input ())  
print("中數:",median(X,L))
```

Ch6. 習題

- ▶ 6-5.說明：
- ▶ 若陣列元素個數為奇數，則中位數為正中間元素的值。

0	1	2	3	4	5	6
---	---	---	---	---	---	---



$$Q = \text{第 } \frac{n}{2} \text{ 個元素的值}$$

- ▶ 若陣列元素個數為偶數，則中位數為中間兩個元素的平均值。

0	1	2	3	4	5
---	---	---	---	---	---



$$Q = \text{第 } \frac{n}{2} \text{ 及 } \frac{n}{2} + 1 \text{ 兩個元素的平均值}$$

休息一下~



头条号 / fa芽儿