

第七單元

決策運算

人類的生活必須不斷面對決策問題，連我家一個不到三歲的小孩，也常要思考他手裡的十元是要坐電動車還是買棒棒糖。程式語言是協助解決人類問題的工具，當然也有決策指令。本單元即是探討 Python 決策所需使用的運算子與指令。請先鍵入以下程式，並觀察執行結果。

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

鍵入以上程式，請留意冒號『:』與縮排問題。冒號『:』不可遺漏，打完冒號『:』，電腦會自動縮排。鍵入『else:』前，要先前進再打字。其次，請將 a=66 換成 55，並觀察執行結果。以上『>=』稱為『關係運算子』、『if else』稱為決策指令。請再鍵入以下程式，並觀察執行結果。

```
x=3
y=4
if(x>0 and y>0):
    print('第一象限')
```

以上『and』稱為『邏輯運算子』，它可以讓我們同時連結兩個以上的關係運算子，這些都在本單元介紹。

決策運算子

決策運算子包含關係運算子與邏輯運算子，分別說明如下：

關係運算子 (Relational Operators)

關係運算子又稱為比較 (Comparison) 運算子，用於資料之間的大小比較，比較的結果可得到 True 或 False，下表是 Python 的關係運算子符號，此與 C/C++ 相同。

運算子	定義	優先順序	結合律
<	小於	9	由左至右
>	大於	9	由左至右
<=	小於等於	9	由左至右
>=	大於等於	9	由左至右
==	等於	9	由左至右
!=	不等於	9	由左至右

例如，

```
x=3;y=4
print(x==y) #False
print(x!=y) #True
print(x<y) #True
print(x=y) #錯，此為指派運算子
```

邏輯運算子 (Logical Operators)

當同一個運算式要同時存在兩個以上的關係運算子時，每兩個關係運算子之間必須使用邏輯運算子連結，例如，您要找『男生』且『年齡大於 40』，此一選擇就同時含有兩個關係運算式，此時就要使用邏輯運算子連結。下表是 Python 的邏輯運算子，C/C++ 用符號『! ,&&,||』表示，Python 直接用單字表示，如下表，這樣反而比較好記。

運算子	定義	優先順序	結合律
not	邏輯否定運算	10	由右至左
and	邏輯 and 運算	11	由左至右
or	邏輯 or 運算	12	由左至右

not

邏輯 not 是將『True』轉為『False』，『False』轉為『True』。

and

邏輯 and 是兩件事都 True，結果才是 True，其餘都是 False，其真值表如下：

事件1	事件2	結果
True	True	True
True	False	False
False	True	False
False	False	False

or

邏輯 or 是兩件事只要有一件為 True，那就為 True，只有兩件事全為 False，才為 False，其真值表如下：

事件1	事件2	結果
True	True	True
True	False	True
False	True	True
False	False	False

例如：

```
x=3;y=4
print(not (x==y)) #True
```

```
print(x>0 and y>0) #True
print(x>0 or x==y) #True
```

範例7a

若有一數學式，判斷 x 是否滿足 $1 < x \leq 6$ ，請轉換為 Python 語言敘述。

題解

1. $1 < x \leq 6$ 是數學語言，Python 的表達方式是
 $x > 1$ and $x \leq 6$
2. 請鍵入以下程式，並觀察執行結果。

```
x=3
print(x>1 and x<=6)
x=8
print(x>1 and x<=6)
```

自我練習

若有一數學式，判斷 x 是否滿足 $x > 3$ or $x < -2$ ，請轉換為 Python 語言敘述。

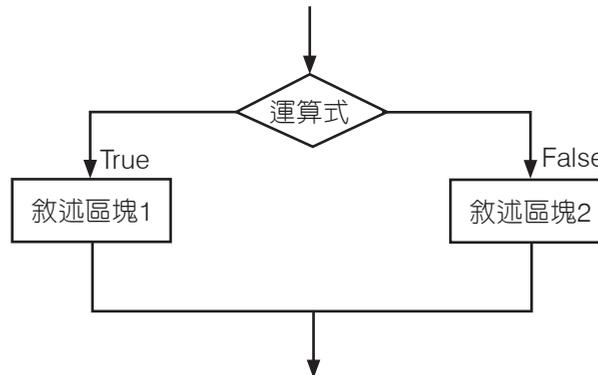
if~else ~

生活上常會面對『假如~否則~』，Python 的指令是『if~else~』，其語法如下：

```
if 運算式:
    敘述區塊1;
[else :
    敘述區塊2;]
```

請特別留意冒號『:』與『程式縮排』，因為冒號『:』是 Python 特有；『縮排』在其他語言是美觀性質，但 Python 是語法也就是規定，因為縮排才是屬於 if else 要執行的敘述子區塊（若使用 Spyder

編寫程式，Spyder 會幫您補上冒號與縮排)，以上程式其流程圖如下：



例如：

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

以上 if 條件程式成立所要執行的敘述區塊與 else 的程式區塊一定要縮排，因為，程式縮排在其它語言是美觀問題，但 Python 是語法，所以沒有縮排就不行。例如，以下程式就不行。

```
a=66
if a>=60:
b="Pass"
else:
b="Fail"
print(b)
```

但若是僅執行一個敘述，這樣也行，但是視覺效果很差。

```
a=55
if a>=60:b="Pass"
else:b="Fail"
print(b)
```

請留意，以下兩個程式的執行結果就不同了，右圖的『print(b)』屬於 else 成立時的敘述區塊。

<pre>a=66 if a>=60: b="Pass" else: b="Fail" print(b)</pre>	<pre>a=66 if a>=60: b="Pass" else: b="Fail" print(b)</pre>
---	---

C 等用大括號『{ }』來表示程式區塊的範圍，Python 直接用縮排表示，那就更簡潔了。其次，也可以將否則的部分放在前面當預設值，那程式如下：

```
a=46;b="Fail"
if a>=60:
    b="Pass"
print(b)
```

或

```
a=66;b="Fail"
if a>=60:b="Pass"
print(b)
```

if 內可否放 if，當然可以，此稱為巢狀 if，請看範例 7b。

■ 冒號(:)

冒號(:) 也是 Python 特有的符號，往後的 for、while、函式、類別定義，都需要這個符號。

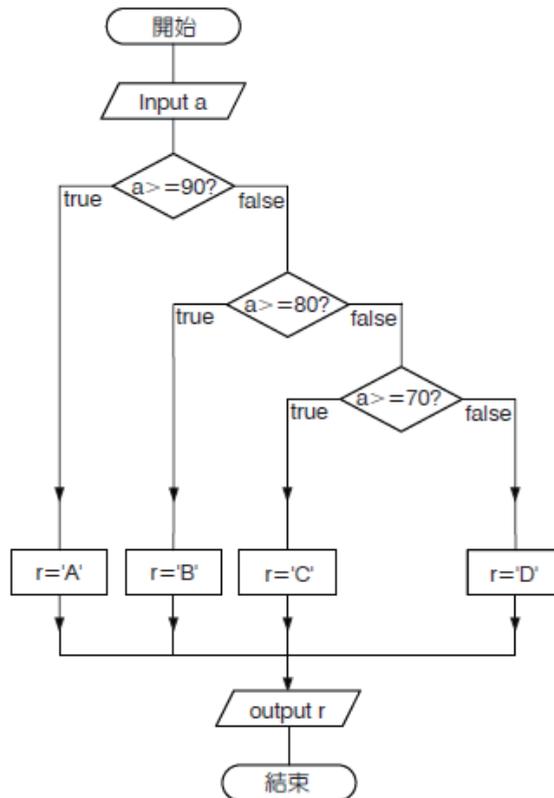
範例7b

請寫一個程式，完成以下要求：

1. 輸入一個 0~100 的分數。
2. 當分數大於 90 分時，輸出 A。
3. 當分數介於 80~89 時，輸出 B。
4. 當分數介於 70~79 時，輸出 C。
5. 當分數介於 0~69 時，輸出 D。

流程分析

1. 使用流程圖分析如下：



2. 以上每一個決策流程點，都有兩個分歧點，所以適用 if~else。

3. 每一個 else 後面均需進一步決策流程，所以可在每一 else 後面再放置 if。

輸出結果

```
input a grade: 92
the grade is A
```

程式列印

```
a=int(input("input a grade: "))#請留意input傳回字串型態
if(a>=90) :          # 高於90分為A
    r='A'
else :
    if(a>=80):      # 介於 80與90分為B */
        r='B'
    else :
        if(a>=70):
            r='C'
        else :
            r='D'
print("the grade is %c" % r)
```

補充說明

1. 請留意 input 傳回字串型態，所以要用 int() 轉換資料型態。
2. 此題有人會寫成

```
if (90<a<100 )
```

但 Python 並沒有這種運算式，因為 $90 < a < 100$ 是數學語言，不是程式語言。

3. 有人會寫成

```
if (a<100 and a>=90) :b='A'
if (a<90 and a>=80) : b='B'
if (a<80 and b>=70) : b='C'
if (a<70 and b>=0) : b='D'
```

這樣雖然也可以，但是其執行效率非常差，因為不管分數為何，僅有一項會成立，如此會讓電腦不斷進行無效的判斷。而且，電腦的判斷時間遠大於執行計算的時間。

4. 也有人會這樣寫，但這是運算思維錯誤，而且結果也不對。

```
a=int(input("input a grade: "))
if(a>=60):
    r='D'
else:
    if(a>=70) :
        r='C'
    else:
        if(a>=80):
            r='B'
        else:
            r='A'
print("the grade is %c" % r)
```

自我練習

- 請寫一程式，判斷所輸入的數是正數、0 或負數。
- 某一貨品定價 100 元，若購買 10 件(含)以上打 9 折，若購買 30 ~ 99 件則打 8 折，若購買 100 件以上則打 7 件，試寫一程式可以輸入購買件數而得總價。測試資料如下：

輸入	輸出
5	500
10	900(100*10*0.9)
30	2400(100*30*0.8)
100	7000(100*100*0.7)

- 直線。直線標準式為 $ax+by+c=0$ ，請寫一程式，可以指派一直線係數 a,b,c 。其次，可再輸入任一點座標，並判斷所輸入點是否在直線上。例如，指派 a,b,c 分別是 1,2,-4，那方程式就是 $x+2y-4=0$ ，其次，輸入點若是 (2,1)，那就是在直線上，

若是 (1,2)，那就不在直線上。

4. 請寫一程式，可以輸入一個小寫字元，請判斷其是否為母音。說明，字元 a,e,i,o,u 稱為母音，其餘為子音。
5. 同上題，請寫一程式，可以自動產生一個小寫字元，並判斷是否為母音。
6. 請寫一個程式，可以產生一個 0 到 25 的亂數，且依以下分數顯示燈號
21 ~ 25：五個燈。
16 ~ 20：四個燈。
11 ~ 15：三個燈。
6 ~ 10：兩個燈。
1 ~ 5：一個燈。
0：零個燈。
7. 表決器。假設有一項評審工作，有 3 位評審，當其中兩人或以上同意，則表示過關，請設計此程式。
8. 請輸入一個整數 x ，並判斷此整數是否滿足 $1 < x \leq 6$ 。
9. 請自動產生一個 -10 到 10 的亂數 x ，並判斷此整數是否滿足 $x > 3$ or $x < -2$ 。
10. 請輸入 3 個線段長度，判斷這 3 個線段，是否可圍一個三角形。圍成三角形的條件是，任兩邊之和要大於第三邊？（提示：任兩邊之和要大於第三邊的數學語言是 $a+b > c$ and $a+c > b$ and $b+c > a$ ，請將以上數學語言轉為 Python。測試資料 3,4,5 可以；1,1,3 不可以）
11. 請輸入 4 個線段長度，判斷這 4 個線段，是否可圍一個凸的四邊形。例如，1,1,1,4 不行,1,2,3,4 可以。

12. 若有一數學式，同時判斷六個數字是否滿足 $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$ 請轉換為 Python 敘述。例如， $a_1=1, b_1=2, c_1=3$ 與 $a_2=2, b_2=4, c_2=6$ 就滿足； $a_1=1, b_1=2, c_1=3$ 與 $a_2=1, b_2=2, c_2=5$ 就不滿足。
13. 若有一數學式，同時判斷六個數字是否滿足 $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$ 又如何表示呢？

範例7c

請寫一個程式，可以判斷所輸入座標的所在象限。

執行結果

```
input x,y:2,-3
IV
```

程式列印

```
x,y=eval(input("input x,y:")) #數字請用逗號『,』隔開，例如輸入 2,-3
if x>0 :
    if y>0:
        b="I"
    else:
        b="IV"
else:
    if y>0:
        b="II"
    else:
        b="III"
print(b)
```

補充說明

1. 請留意 eval() 傳回數值型態。
2. 此題目有人會寫成

```
x,y=eval(input("input x,y:")) #數字請用逗號『,』隔開，例如 2,-3
if (x>0 and y>0) :
    b="I"
```

```
if (x<0 and y>0) :
    b="II";
if (x<0 and y<0) :
    b="III"
if (x>0 and y<0 ):
    b="IV"
print(b)
```

這樣雖然沒有錯，但是執行效率非常差，因為電腦要不斷的比較。

自我練習

1. 同上範例，但增加先判斷是否在原點或 x、y 軸上。測試資料如下：

輸入 (x,y)	輸出
(0,0)	原點
(0,3)	y 軸
(3,4)	I

2. 同上題，但程式一執行，要求先輸入密碼，密碼若為『abcd』，那才可執行本程式。
3. 同上題，但程式一執行，要求先輸入密碼，密碼若為『aa11』、『bb22』、『cc33』或『abcd』，那才可執行本程式。

if~elif~

C 等都有 switch case 解決多個分歧決策問題，Python 卻沒有，但使用 if~elif~ 替代。例如，範例 7b 的程式，以 if~elif~ 重做如下：

```
a=int(input("input a grade: "))
if(a>=90) :          # 高於90分爲A
    r='A'
elif(a>=80):        # 介於 80與90分爲B */
    r='B'
```

```

elif(a>=70):      # 介於 70與80分為C */
    r='C'
else :
    r='D'          # 不符合上述情況則為D */
print("the grade is %c" % r);

```

自我練習

1. 請寫一程式，將所輸入的 0、1、2...6，轉為 '星期日'、'星期一'...'星期六' 等字串。
2. 同範例 7b 自我練習第 6 題的燈號顯示，但請用 if~elif 重作。
3. 請寫一程式，可以輸入一個月份，然後得到該月的天數。本例假設 2 月是 28 天。
4. 請寫一程式，可以自動產生一個 0 到 127 的亂數，以字元方式輸出此亂數，並判斷是否為大寫字元、小寫字元、或數字。(提示：請線上查詢 ASCII 的編碼方式，如下圖)

ASCII Table															
Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

猜拳遊戲

範例7d

猜拳遊戲。請寫一個程式，可以由人和電腦猜拳，並評定勝負。

執行結果

```
input 1,2,3:2  
您出 石頭,電腦出 石頭,結果是 平手
```

演算法則

1. 這一任務就是寫人工智慧程式的入門了，寫程式前先想一下您和一個3歲小孩猜拳，那您們如何評判勝負？首先，我們先規定有三種拳，分別是『剪刀、石頭與布』，且『剪刀贏布，石頭贏剪刀、布贏石頭，兩者相同則平手等』。『人工智慧』就是要把這些規定以程式語言表示，並由以上規定來評判勝負。
2. 本例因為電腦有三種拳法，人也有三種拳法，所以共有9種情況，您要讓電腦也能思考，其實就是將以上所有可能發生的情況，以程式語言先規定好。
3. 資料的數位化。我們人類猜拳是直接用手勢表示『剪刀、石頭與布』，但是如何讓電腦瞭解您的拳法呢，裝一個攝影機當然可以，但是也還要影像處理，這不是我們這一單元任務，我們電腦只有鍵盤，所以可以先將以上『剪刀、石頭與布』以『1,2,3』表示，此即為資料的數位化。若使用攝影機，也是要影像處理，再將您的拳法以簡單的數字表示，也是要用『1,2,3』表示，因為用『1,2,3』，表示只要1byte就可以，若您用『剪刀、石頭與布』表示，當然也可以，但一個中文字就佔用2byte，且往後的處理也是比較複雜。

程式列印

```
import random
a=int(input("input 1,2,3:"))#記得要轉型
b=random.randint(1,3)
astr=''
bstr=''
r=''
if a==1:#people
    astr='剪刀'
    if b==1: #computer
        bstr='剪刀'
        r='平手'
    elif b==2:
        bstr='石頭'
        r='computer win'
    elif b==3:
        bstr='布'
        r='people win'
elif a==2:
    astr='石頭'
    if b==1:
        bstr='剪刀'
        r='people win'
    elif b==2:
        bstr='石頭'
        r='平手'
    elif b==3:
        bstr='布'
        r='computer win'
else:
    astr='布'
    if b==1:
        bstr='剪刀'
        r='computer win'
    elif b==2:
        bstr='石頭'
        r='people win'
    elif b==3:
        bstr='布'
        r='平手'
print("您出 %s,電腦出 %s,結果是 %s" % (astr,bstr,r))
```

補充說明

這一題輸入資料時，若忘了將資料轉為數值，如以下程式，那真的哭不完，因為沒有錯誤，但就是沒結果。

```
a=input("input 1,2,3:")#請留意a是字串型態
```

自我練習

1. 那三個人同時猜拳呢，由電腦產生兩個亂數，由您和電腦猜，且評判輸贏。(提示：兩個人共 9 種情況，用兩層決策，三個人共 27 種情況，可用 3 層決策)

一元二次方程式

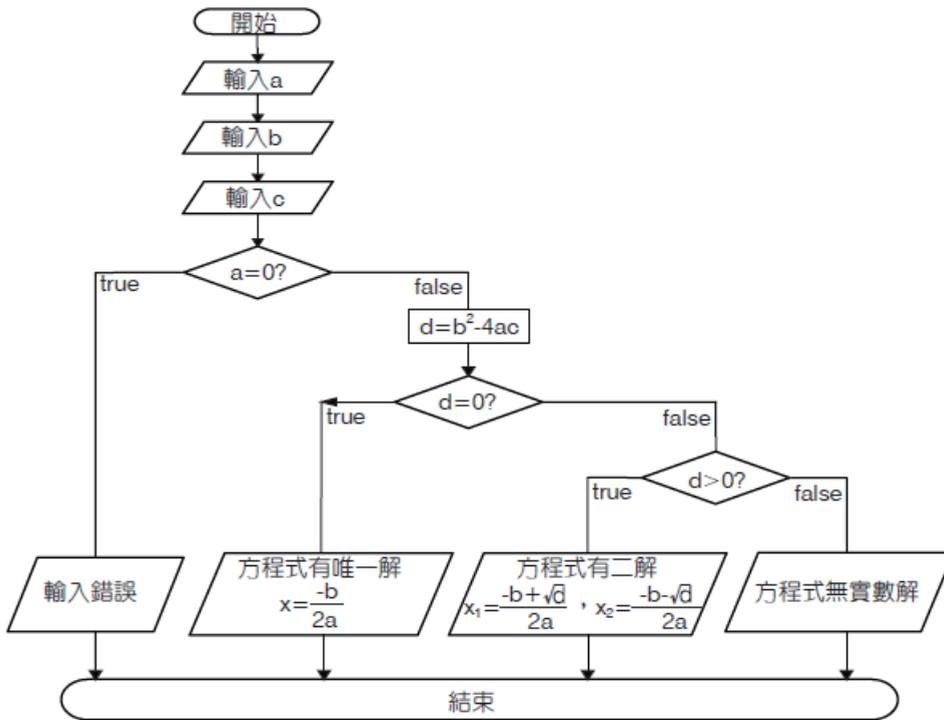
前面我們直接假設所輸入的一元二次方程式有解，因為不是隨便給三個係數，一個方程式就通通有解，本例則要加上判斷了，解一元二次方程式的演算法如下：

1. 設有一元二次方程式如下：

$$ax^2 + bx + c = 0$$

2. 若 $a=0$ 則輸出“輸入錯誤”。
3. 令 $d=b^2 - 4ac$ 。
4. 若 $d=0$ ，則方程式有唯一解 $x = \frac{-b}{2a}$ ；否則，若 $d>0$ ，則方程式有二解 $x_1 = \frac{-b + \sqrt{d}}{2a}$ ， $x_2 = \frac{-b - \sqrt{d}}{2a}$ ；否則，無實數解。

以上演算分析，以流程圖說明如下：



範例7e

請設計一個程式，可以解一元二次方程式 $ax^2 + bx + c = 0$ 。

執行結果

```
input a,b,c:1,2,1
only one answer,x1,x2= -1
```

程式列印

```
a1,b1,c1=eval(input("input a,b,c:"))
a=int(a1)
b=int(b1)
c=int(c1)
if(a==0):
    print("input error")# 若a=0，則列印錯誤訊息
```

```
else:
    d=b**2-4*a*c# 計算d值 */
    if(d==0):
        print("only one answer,x1,x2= %d" % (-b/(2*a)))
    elif d>0 :
        d=d**(1/2)
        x1=(-b+d)/(2*a)
        x2=(-b-d)/(2*a)
        print("two answer,x1= %f,x2=%f"%(x1,x2))
    else:
        print("no real answer")
```

自我練習

1. 解二元一次方程式。解二元一次方程式的演算法如下：

(1) 設二元一次方程式如下：

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

(2) 令 $d = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ (表示 $d = a_1b_2 - a_2b_1$)

(3) 假如 $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$ ，則方程式無限多解，且程式結束。

(代表兩重疊直線)

(4) 假如 $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$ ，則方程式無解，且程式結束。

(代表兩平行直線)

$$(5) \quad x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{d} = (c_1b_2 - c_2b_1) / d$$

$$(6) \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{d} = (a_1c_2 - a_2c_1) / d$$

請設計一個程式，可以解二元一次方程式。

範例7f**三角形判斷 (APCS105 第二梯次試題)**

1. 若已知三角形三邊長，判斷是否構成三角形、評定三角形種類與計算三角形面積的演算法如下：
 - (1) 輸入三角形的三邊長 a 、 b 、 c 。
 - (2) 任兩邊之和要大於第三邊，才能構成三角形。
 - (3) 若 c 是最長邊，假如 $a^2 + b^2 > c^2$ 則為銳角三角形；否則，假如 $a^2 + b^2 = c^2$ ，則為直角三角形；否則此三角形為鈍角三角形。
 - (4) 令 $d = \frac{1}{2}(a + b + c)$
 - (5) 三角形面積 = $\sqrt{d(d-a)(d-b)(d-c)}$
2. 請輸入三角形三邊長，首先判斷是否構成三角形、其次判別三角形的種類，最後計算其面積。

執行結果

```
input a,b,c:3,4,5
直角三角形
6.0
```

程式列印

任兩邊之和要大於第三邊，本例就將最大邊找出來，這樣就一石二鳥，可以方便判斷是否形成三角形，也能判斷三角形種類。

```
a1,b1,c1=eval(input("input a,b,c:"))
a=int(a1)
b=int(b1)
c=int(c1)
if a>b:
    a,b=b,a
if b>c:
    b,c=c,b
if a>b:
```

```
a,b=b,a
if (a+b>c) :
    c2=c**2
    b2=b**2
    a2=a**2
    if ((a2+b2)>c2):
        t="銳角三角形"
    elif ((a2+b2)==c2):
        t="直角三角形"
    else:
        t="鈍角三角形"
    d=(a+b+c)/2
    area=(d*(d-a)*(d-b)*(d-c))**(1/2)#留意括號的對稱
    print(t)
    print(area)
else:
    print("無法構成三角形")
```

補充說明

1. 『任兩邊之和要大於第三邊』如何以程式表示？

這句話同義於， $a+b>c$ and $b+c>a$ and $c+a>b$ 或將三邊長由小而大重新排列，最小的放入 a ，其次放入 b ，最大的放入 c 。最小的兩邊之和若小於等於第三邊，則此三邊未能構成三角形，程式提早離開。本例因為還要判斷三角形的種類，所以採用後者。

自我練習

1. 假設所得稅稅率累進法則如下：
 - (1) 淨所得 30 萬以下繳納 6%。
 - (2) 淨所得 30 ~ 80 萬之間，則前面 30 萬繳納 6%，超過 30 萬的部分繳納 13%。
 - (3) 淨所得在 80 ~ 200 萬之間繳納 21%。(前面 30 萬繳 6%，30 ~ 80 萬之間繳 13%)
 - (4) 淨所得超過 200 萬，超過的部分繳納 30%。

試寫一程式可以輸入淨所得，並計算應繳納稅額。測資如下：

淨所得	納稅額
20 萬	$20 \text{ 萬} * 6\% = 12000$
40 萬	$30 \text{ 萬} * 6\% + 10 \text{ 萬} * 13\% = 31000$
100 萬	$30 \text{ 萬} * 6\% + 50 \text{ 萬} * 13\% + 20 \text{ 萬} * 21\% = 125000$
220 萬	$30 \text{ 萬} * 6\% + 50 \text{ 萬} * 13\% + 120 \text{ 萬} * 21\% + 20 \text{ 萬} * 30\% = 395000$

2. 假設自來水費率如下：

A. 100 度以下，每度 3 元。

B. 100 ~ 300 度，超過 100 度的部分，每度 5 元。

C. 300 度以上，超過 300 度的部分，每度 6 元。

根據以上條件，寫一程式，可輸入用水度數，得到水費。例如，測資如下表：

度數	水費
50	$50 * 3 = 150$
200	$100 * 3 + 100 * 5 = 800$
400	$100 * 3 + 200 * 5 + 100 * 6 = 1900$

3. 假設計程車的計費方式如下：

1000 公尺以內 60 元。超過 1000 公尺時，

(1) 日間以每 500 公尺加收 6 元，不足 500 公尺時，以 500 公尺計算。

(2) 夜間以每 300 公尺加收 6 元，不足 300 公尺，以 300 公尺計算。

請根據以上條件，寫一程式完成車資的計算。本例，日間輸入 1，夜間輸入 0，再輸入一個里程數，並計算車資。測資如下表：

模式	里程	車資	模式	里程	車資
1	500	60	0	500	60
1	1800	$60 + 6 * 2 = 72$	0	1800	$60 + 8 * 3 = 84$

4. 寫一程式輸入 x 值，並印出其所對應的值，其函數對應如下：

$$y = f(x) = \begin{cases} x+3 & x > 3 \\ x^2 & 1 \leq x \leq 3 \\ \sqrt{x} & 0 < x < 1 \\ 0 & x \leq 0 \end{cases}$$

5. 請寫一程式，可以輸入三點座標，輸出其面積。

提示：三角形面積如下：

$$\frac{1}{2} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2} |(x_1y_2 + x_2y_3 + x_3y_1 - x_2y_1 - x_3y_2 - x_1y_3)|$$

例如，(0,0),(3,0),(0,4) 面積是 6。

6. 同上題，任意輸入 3 點，檢查是否共線，若未共線，輸出其面積。

提示：任 3 點面積若為 0，則共線。例如，(1,1),(2,2),(3,3) 共線；(1,1),(2,2),(2,3) 不共線。

7. 判斷任意點 D 是否在三角形 ABC 內或外。已知 ABC 三點座標，請寫一程式，可以輸入 D 點座標，並判斷任一點 D 是否在三角形 ABC 內或外。例如，若三角形三點是 (0,0),(3,0),(0,4)，D 若 (1,1) 則在裡面，D 若 (4,4) 則不在裡面。

提示：D 若在三角形內，則三角形 ABC 面積 = DAB + DBC + DAC 的面積。

8. 凸或凹四邊形。已知 ABCD 四點座標，請寫一程式，可以判斷四邊形 ABCD 是否為凸或凹四邊形。例如，(0,0),(3,0),(4,4),(0,4) 是凸四邊形，(0,0),(3,0),(1,1),(0,4) 是凹四邊形。

提示：若為凸四邊形，則任一點，均不得在任三點所圍成的三角形內。

9. 同上題，若為凸四邊形，並計算其面積。提示：任意凸多邊形，都可以將第 5 題公式推廣，並且代入。例如，(0,0),(3,0)(4,4),(0,4) 面積是 12。
10. 於二維平面中，請寫一程式，可以判斷兩直線平行、相交或重合，若平行計算其距離，相交計算其交點。例如， $a_1=3, b_1=4, c_1=5$ 與 $a_2=6, b_2=8, c_2=10$ 代表重合； $a_1=3, b_1=4, c_1=5$ 與 $a_2=3, b_2=4, c_2=10$ 代表平行，此時求其距離，此距離是 1；若 $a_1=3, b_1=4, c_1=7$ 與 $a_2=4, b_2=-3, c_2=1$ 代表相交，交點是 (1,1)。

(兩平行直線 $ax+by=c_1, ax+by=c_2$ 距離公式 = $\frac{|c_1 - c_2|}{\sqrt{a^2 + b^2}}$)

11. 假如有資料如下：

1,30,40,50

2,40,50,60

3,50,60,70

4,60,70,80

5,70,80,90

請寫程式讀入以上資料，計算每個人的平均、計算每個人不及格科數、計算直向每科平均、計算直向每科不及格人數。

(請不要用迴圈與 list)

12. 本書的特色就是『自我練習』多，因為程式設計不是一直『讀』，而是要能舉一反三，這樣才能開竅與頓悟。因為我曾經發現，有人一直上入門課，但就是寫不出任何程式，這都是沒有自己做題目的結果，也就是『一直聽沒有動手做』真的沒用。