

中學生 Python 程式設計教材 特色

各位教育先進大家好

又到選書季，目前『資訊科技』教科書，不論是國中與高中，此課綱內容包金包銀包銅也包鐵，但是，考量城鄉差距，各校學生程式參差不齊，這麼豐富的内容，可能您再怎麼努力，學生也不一定領請，而且讀大學時，教授通通很優秀，也沒有老師將整本教材教完，所以我是認為選擇部分課綱，好好發揮，那學生收穫可能更大。考量程式設計才是資訊科技重點，教科書的程式設計篇也僅能點到為止，而且教科書因為審定時程，有些語言則過時，所以我建議中學生以『Python 為程式設計入門教材，最為適合，理由如下。選到適合自己的好書可以讓您在電腦教室住套房、輕鬆吹冷氣，健健康康、開開心心，廣栽桃李到六五。選到別人認為最好的書，每天可能拚到死，學生還是雞飛狗跳，那就校長室咖啡喝不完。其次，本書不定期舉辦教材研習，研習單位請搜尋『泉勝』，也歡迎各校邀約分享，凡邀約者可送教材或 Arduino 教具。

特色 1、人人都能寫出程式。

特色 2：充分的解說。

特色 3：與範例相近的自我練習。

特色 4：使用數學闡述『程式指令的緣由』。

特色 5：使用程式闡述數學。

特色 6、以 Python 完成電腦輔助測驗，輕輕鬆鬆就可寫出英文聽寫程式，由電腦念英文，學生寫出結果，電腦評分。

特色 7、寫出人工智慧入門範例，猜拳、ABGame、十點半自動補牌。

特色 8 含有 APCS 入門考題。

特色 9、以程式繪出數學函數圖形，教您寫程式學數學。

特色 10、含有推甄課程學習成果範例，讓您輕鬆面對升學巨變。

以上特色舉例說明如下：

特色 1、人人都能寫出程式。例如，以下程式可解一元二次方程式。每次只要改變係數，就可解方程式。

```
a=2;b=-7;c=3#變數先指派，可簡化問題。
```

```
d=(b*b-4*a*c)**(1/2)
```

```
x1=(-b+d)/(2*a)
```

```
x2=(-b-d)/(2*a)
```

```
print(x1)#3.0
```

```
print(x2)#0.5
```

因為 Python 是根據使用者習慣而設計，但是以前其他語言都要先學習一大堆規定。

特色 2：充分的解說。例如，以上程式的演算解說如下：

國中解一元二次方程式是先用十字交叉乘法，十字交叉乘法需要一點判斷，可減少計算，但是本範例使用公式法，公式法雖計算較多，但完全不用判斷，最適合由電腦完成。這種強調多計算少判斷，就是電腦與人類不同的運算思維，人類計算能力較差，所以希望使用一些技巧來簡化計算，但是電腦則是計算能力強，判斷能力差，所以強調用計算來簡化程式。解一元二次方程式的公式法，演算步驟如下：

(1) 設有一元二次方程式如下：

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

(2)資料的抽象化，本例指派 a,b,c 三個整數（本例假設整係數方程式），我們就約定此 a,b,c 代表 $ax^2 + bx + c = 0$ 。

(3)令 $d = \sqrt{b^2 - 4ac}$ 。提示：此為數學語言，Python 是 $d=(b*b-4*a*c)**(1/2)$ ，括號、乘號均不能省。)

(4) 則其二解分別為 $x_1 = \frac{-b+d}{2a}$ ， $x_2 = \frac{-b-d}{2a}$ 。（提示：此為數學語言，

Python 是 $x1=(-b+d)/(2*a)$ ，括號、乘號均不能省。)

(5)例如， $2x^2-7x+3=0$ 。其解為 $x_1=3.0, x_2=0.5$ 。

特色 3：與範例相近的自我練習。程式設計不是老師一直講講講，而是要讓學生舉一反三，所以本書有充分的自我練習，老師用書一定可以拿到習題解答。例如，寫一個程式，可以輸入一個二元一次方程式，並求其解（本例假設所輸入的方程式恰有一解）。（其次，每個老師一天可能要連續上 5、6 節，而且現在老師無法 50 歲就退，不能生病了，每個人幾乎都要教到當阿公當阿嫲，適度的放下，老師才能延長使用年限，學生也更能精進。）

運算思維

解二元一次方程式，國中會先教代入消去法與加減消去法，這都需要一點判斷，但可以簡化計算，本題使用公式法，可完全不用判斷，直接計算而得，這樣最適合計算機了，此也是電腦的運算思維。解二元一次方程式的克拉馬公式演算過程如下：

(1). 設二元一次方程式如下：

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

(2)輸入 a1,b1,c1,a2,b2,c2 六個整數。（本例假設整係數方程式）

$$(3) \text{ 令 } d = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1 \text{。 (此稱為行列式表示法)}$$

$$(4) \text{ 則其解分別是 } x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{d} = (c_1 b_2 - c_2 b_1) / d \quad (\text{x 的地方用 } c_1, c_2 \text{ 代替})$$

$$y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{d} = (a_1 c_2 - a_2 c_1) / d \quad (\text{y 的地方用 } c_1, c_2 \text{ 代替})$$

(5) 例如， $3x+y=5$

$$x-2y=-3$$

則其解為 $x=1 \ y=2$

特色 4：使用數學闡述『程式指令的緣由』。學生都會問為什麼有 if、for、while、串列。我的體會分享如下：

if

人類常要面對決策，電腦就用 if 來解決這些人工智慧問題。例如，以下程式可協助使用者將考試分數加上等第。

```
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)
```

迴圈 for 與 while

人類的生活其實充滿週期重複性，例如，1 天 24 小時，1 週 7 天等。電腦當然就有 for 與 while 迴圈。for 則用於程式設計階段就知道要做幾次，例如，以下程式，可解決乘法運算子。(備註：電腦內部只有加法與比較)

```
a=5
```

```
b=8
```

```
s=0
```

```
for b in range(1,b+1):  
    s=s+a  
print(s)#40
```

請繼續鍵入以下程式，並觀察執行結果。井號『#』右邊是註解，僅給人看，先不要打。

```
a=8#被除數  
b=3#除數  
q=0#商  
while(a>=b):#只要(被除數>除數) 就執行迴圈  
    a=a-b # (被除數)-(除數)  
    q=q+1 #商每次遞增 1  
print(q)# 商數  
print(a)#餘數
```

以上是除法運算，前面乘法運算於程式設計階段，就已經知道運算次數，所以用 for，但是除法運算就是不曉得要減幾次，只能一面減，一面判斷，所以用『while』，關於『while』的用法，請看 4_3 節。

串列

處理少量的資料，可以為每一筆資料設定一個變數，但若資料數量很多，例

如，若有資料如下：

```
3,8,4,7,2,9
```

要求其和、極大、極小呢？是不是指派6 個變數儲存如下

```
a1=3;a2=8;a3=4;a4=7;a5=2;a6=9
```

這樣當然也可以，但是程式會非常冗長，為了改善此一問題，就要使用本節的

串列型態了。因為串列型態，可以使用『索引』，再配合for 與while 迴圈而簡

化程式，此即為本章重點。例如，以上資料就可宣告串列如下：

```
a=[3,8,4,7,2,9]
```

a 是串列變數名稱，變數名稱也應該遵循2_1 節變數命名規則。那一下程式，

就可以一一輸出、或一一改變其值。

```
for i in range(0,6):
    print(a[i])
for i in range(0,6):
    a[i]=a[i]*2
```

特色 5：使用程式闡述數學。前面我們先用簡單數學闡述程式，等到程式進步了，現在則要用程式闡述進階抽象的數學（其實程式與數學互為表裡，兩者想輔相成）。例如，以下程式使用循序猜值法解開根號。

循序猜值法

所謂循序猜值法，就是將所有可能的解一一循序代入，又稱為暴力猜值法。例如，您要求任一數的開平方，因為開平方的結果一定在 0 與此數之間，那我們就從 0 開始，每次遞增 1、0.1、0.01 或 0.001...，至於是要遞增多少，那就依您要的精密度了。例如，要求整數解，那就每次遞增 1；要求到小數 1 位，那就遞增 0.1；要求到小數兩位，就遞增 0.01 等等等，請看以下範例說明。

範例 6_1b 請用循序猜值法求任意數的平方根。

演算法

1、人類求開根號的方法，是使用二項式定理。 $(a+b)^2=a^2+2ab+b^2$ ，但是電腦就不用如此麻煩，電腦可用循序猜值與二分猜值法。循序猜值法程式如下：

1、以下程式每次遞增 1。

```
a=9
for i in range(9):
    if i*i>=a:
        print(i)#3
        break
```

2、以下程式每次遞增 0.1。因為 range 的遞增值只能整數，不能是實數 0.1，所以我把他先放大 10 倍，求結果時再除以 10。

```
a=9
s=10#放大倍數
for i in range(a*s):
    if (i/s)*(i/s)>=a:
        print(i/s)#3.0
        break
```

3、同理，以下程式每次遞增 0.01。

```
a=9
s=100#放大倍數
for i in range(a*s):
    if (i*i/(s**2))>=(a):
        print(i/s)
```

break

二分猜值法程式如下：

1. 本例以求9 的平方根為例。
2. 求解9 的平方根，其解必在0 到9 之間，所以設定下界為0，上界為9。首先，先猜4.5，如下圖步驟1，但4.5 的平方為20.25，大於9，表示猜的太大，那就縮小範圍，將上界調為4.5，持續在0 與4.5 之間猜值，如下圖步驟2。第二次就猜2.25，但是2.25 的平方為5.025，小於9，表示猜的太小，那就調整下界為2.25，且持續在2.25 與4.5 之間猜，如下圖步驟3，那要猜到何時呢？答案是設定一個精密度，例如，您要小數一位，那就是下界與上界之間的距離大於0.1（若是要小數兩位，那就是下界與上界之間的距離大於0.01），通通要繼續猜；當離開迴圈時，此時下界或上界的值，就都是所要求的答案了。

猜值步驟	猜值內容		
1	x1 0	x 4.5 (太大)	x2 9
2	x1 0	x 2.25 (太小)	x2 4.5
3		x1 x 3.37 (太大)	x2

- (1) 設求解的正數為 y ，則其平方根必在 $x_1=0$ 與 $x_2=y$ 之間。
- (2) 首先猜 x_1+x_2 之和的一半 x 。
- (3) 若所猜 x 的平方小於 y ，表示猜的太小，並縮小猜值範圍為 $[x, x_2]$ 。
- (4) 若所猜的 x 的平方大於 y ，表示猜的太大，並縮小猜值範圍為 $[x_1, x]$ 。
- (5) 重覆步驟(2)、(3)、(4)，只要 $[x_1, x_2]$ 的範圍大於所要求的精密度（小數兩位0.01 或小數三位0.001），則要繼續猜；當離開迴圈時，此時的 x_1 或 x_2 即

為平方根。

程式列印

```
y=9.0
x1=0.0
x2=y
n=1
while (abs(x1-x2)>0.01):#距離>0.01 繼續猜
    print("%d:x1=%2.2f,x2=%2.2f" %(n,x1,x2))
    x=(x1+x2)/2
    t=x*x
    if t<y :
        x1=x
    else:
        x2=x
    n=n+1
print(x)#2.99
```

自我練習

- 1、請以二分猜值法，求解兩數相除的結果。
- 2、請以二分猜值法，求解一正數的立方根。
- 3、請寫一程式，由電腦產生一個 1 到 100 的亂數，由使用者用二分猜值法猜，電腦應逐次回答太大、太小、或猜中，且回應幾次猜中。

特色 6、以 Python 完成電腦輔助測驗

電腦輔助測驗

坊間有不少電腦輔助測驗系統，這些系統有填充題、有選擇題等，都會自動評判對錯、都會計時、都可以反覆練習，達到熟能生巧的效果，本書將會在各章介紹一些範例，希望您自己也可以設計出適合自己的電腦輔助學習與測驗系統。

範例 4_4a 心算測驗。心算測驗不用準備題庫，只要產生亂數就可以，程式如下：

執行結果

g408

程式列印

```
import random
import time#可以使用系統時間函式
r=0
w=0
```

```

t1=int(time.time())#取系統時間,單位是秒
for i in range(10):
    a=random.randint(1,9)
    b=random.randint(1,9)
    c=a+b
    d=int(input('%d :%2d+%2d=% (i+1,a,b)'))
    if d==c :
        r=r+1
    else:
        w=w+1
t2=int(time.time())
t=t2-t1
print('The right answer is:%d'% r)
print('The wrong answer is:%d'% w)
print('total time=%d second'% t)

```

範例 6_3e 英文打字練習.請寫一個程式，可以出現 10 題英文單字，讓使用者可鍵入單字，並評判正確與錯誤題數、使用時間。

程式列印

1、將單字用串列儲存如下：

```

import time #使用 time.time()
a=['look','at','one','two','tree','and','day','book','from','go']
n=10
right=0
wrong=0
t1=int(time.time())#取得系統時間的秒數
for i in range(n):
    print(a[i],':')
    b=input()
    if b==a[i]:
        right=right+1
    else:
        wrong=wrong+1
t2=int(time.time())
t=t2-t1
print('right=%d'% right)
print('wrong=%d'% wrong)
print('total time=%d second'% t)

```

自我練習

- 1、 請將 `a=['look','at','one','two','tree','and','day','book','from','go']` 改為 `a=['look at one two tree and day book from go']`，並寫程式完成以上範例功能。
- 2、 請將自己課本單字輸入本題。
- 3、 那如何測驗整句英文。
- 4、 如何測驗中文打字。

範例6_3f 如何寫填充題程式。英文單字需要不斷練習才會進步，本單元將寫一個單字記憶程式，他會不斷出現中文，您可輸入英文，電腦會評判是否正確，並統計正確與錯誤題數。

執行結果

G609

程式列印

#將題目用串列儲存，待學習檔案後，可將這些資料放在檔案

```
import time
a=[["bus","n","公車"],
    ["delighted", "adj", "高興的"],
    ["gear", "n", "服裝"],
    ["behave", "v", "表現"],
    ["attract", "v", "吸引"],
    ["criticism", "n", "批評"],
    ["concerned", "adj", "擔心的"],
    ["There are four people in my family.", " ", "我們家有四個人"],
    ["good", "adj", "好的"],
    ["apple", "n", "蘋果"],
    ["look", "v", "看見"],
    ["book","n","書"],
    ["television","n","電視"],
    ["cellphone","n","手機"]];
n=10
right=0
wrong=0
t1=int(time.time())#取得系統時間的秒數
for i in range(n):
    print(a[i][2],a[i][1],':')
    b=input()
    if b==a[i][0]:
        right=right+1
    else:
```

```

wrong=wrong+1
t2=int(time.time())
t=t2-t1
print('right=%d'% right)
print('wrong=%d'% wrong)
print('total time=%d second'% t)

```

自我練習

1、請將自己的學科，寫成填充題。

範例6_3g 選擇題測驗。請寫一程式，可以讓使用者使用選擇題測驗。

執行結果

G610

程式列印

#將題目用串列儲存，待學習檔案後，可將這些資料放在檔案

#第三個是答案

```
import time
```

```

a=[["book","n","2","學校","書","老師","校長"],
    ["delighted","adj","1","高興的","悲傷的","生氣的","憤怒的"],
    ["gear","n","3","背包","褲子","服裝","鞋子"],
    ["behave","v","2","生氣","表現","難過","看到"],
    ["attract","v","4","打架","斥責","排斥","吸引"],
    ["criticism","n","1","批評","讚美","說明","美人"],
    ["concerned","adj","1","擔心的","高興的","傷心的","生氣的"]];

```

```
n=7
```

```
c=0
```

```
d=0
```

```
t1=int(time.time())
```

```
for i in range(n):
```

```
    print("%d . %s,%s : " % (i+1,a[i][0],a[i][1]),end=")#題編號、題目
```

```
    for j in range(1,4+1):
```

```
        print("(%d) %s : " % (j,a[i][j+2]),end=")#輸出四個選項
```

```
    b=input()
```

```
    if (b==a[i][2]):
```

```
        c=c+1
```

```
        print("Right")
```

```
    else:
```

```
        print("Wrong")
```

```
        d=d+1
```

```
t2=int(time.time())
```

```
t=t2-t1
print("The Right number is :%d"%c)
print("The wrong number is :%d"%d)
print('total time=%d second'% t)
```

範例 9f 英文默寫。英文要能進步，就是要強迫自己能將看過的句子覆誦一次，您想想看，中文別人講一次，我們通常可以覆誦一次，其實英文也要這樣自我要求。以下是我寫的程式，只要將一篇文章先輸入或複製貼上，程式就會逐一出現每一個句子、每個句子的第一個單字、並將該句剩餘的單字以底線『_』標示，讓使用者填空輸入，輸入完電腦還能自動批改。

演算法

- 1、將整篇文章複製，以字串儲存。
- 2、將文章以點(.)分解為若干句子。例如，3 個句字有 3 個點，那分解後有 4 句，最後一句是空白，所以本例程式撰寫就減 1。
- 1、本例將每一個句子的第 1 個單字輸出，且每一個單字以底線『_』標示，這樣使用者比較好作答，電腦也比較方便批改。
- 2、逐一由使用者輸入整個句子。
- 3、本例假設使用者所輸入單字數與題目一樣。例如，題目提示 5 個單字，使用者就輸入 5 個單字。
- 4、逐一按順序比對兩個句子的單字，並統計正確單字的個數，也就是第 1 個單字與第 1 個單字比對，第 2 個單字與第 2 個單字比對。
- 5、若全部單字都對，輸出『全對』，否則輸出正確單字的個數。
- 7、輸出正確的句子。

輸出結果

g06

程式列印

```
a="Action speak louder than words.Wasting time is robbing oneself.Never say die."
b=a.split('.')#整篇文章分解
#print(len(b))
#print(b)#由輸出發現，b 的長度是 4，所以以下就減 1
for i in range(len(b)-1):
    c=b[i]+'.'
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    print(d[0],end=' ')
    for j in range(1,dlen) :
        print('____',end=' ')
```

```

#print('.')
e=input("請輸入句子%d : "%i)
f=e.split(' ')
#print(len(f))
s=0
#計算答對幾個單字
for j in range(dlen):
    if (d[j]==f[j]):
        s=s+1
if s==dlen :
    print("全對")
else:
    print("錯了%d 單字" %(dlen-s))
print("正確句子是%s"%c)

```

自我練習

1、本例假設使用者輸入的單字數與提示的單字數一致，若使用者輸入太長或太短都會造成執行錯誤，請問您如何修改程式補救。

範例 9g 英文聽寫。

我們於第一章曾經強調，Python 之所以強，就是有源源不絕的好用套件不斷發表，上網搜尋『Python 文字轉語音』，找到套件如下

(<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/368251/>) :

```

import win32com.client

speaker = win32com.client.Dispatch("SAPI.SpVoice")

a="Hello, it works!"

speaker.Speak(a)

a="洪國勝，您好"

speaker.Speak(a)

```

程式語言的進步真是一日千里，以上程式真的可以發音，也就是視障者，可以將任何中英文文章複製，就可以自動播放。或是，您也可將英文課本複製下來，貼到變數 a，那電腦就會用語音輸出，讓您不斷練習聽力。以下是將文章，先放在記事本，以 a0.txt 存檔（請與 Python 程式放在同一資料夾），如下圖，然後寫程式幫您播放。

```
a07
```

撰寫以下程式，就可讀檔，且整篇發音。

```

import win32com.client

speaker = win32com.client.Dispatch("SAPI.SpVoice")

f='a0.txt'

open(f,'r',encoding='utf-8') as f1:

    data=f1.read()

```

```
f1.close()
print(data)
speaker.Speak(data)
```

其次，根據以上程式，我將以上程式改寫如下，就可完成英文聽寫程式，希望可以幫助大家學習英文。

輸出結果

將英文課文複製，然後貼到以下變數 a，電腦就會逐一唸出每一句，且電腦提示每一句的長度，使用者就可開始輸入聽到的單字，然後電腦還會檢查這些單字是否正確。

g05

程式列印

```
import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
speaker.Speak("Hello, 各位同學大家好, 現在開始聽寫測驗")
#將課文複製，然後貼來以下變數 a, 電腦就會逐一唸出每一句。
a="Action speak louder than words.Wasting time is robbing oneself.Never say die."
b=a.split('.')#整篇文章分解
#print(len(b))
#print(b)#由輸出發現，b 的長度是 4，所以以下就減 1
for i in range(len(b)-1):
    c=b[i]+'.'
    speaker.Speak(c)#您要念幾遍，就寫幾次
    speaker.Speak(c)
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    #print(d[0],end=' ')
    for j in range(dlen):
        print('____',end=' ')
    #print('.')
    e=input("請輸入句子%d: "%i)
    f=e.split(' ')
    #print(len(f))
    s=0
    #計算答對幾個單字
    for j in range(dlen):
        if (d[j]==f[j]):
            s=s+1
    if s==dlen :
```

```
print("全對")
else:
    print("錯了%d 個單字" %(dlen-s))
print("正確句子是%s"%c)
```

特色 7、寫出人工智慧入門範例

猜拳遊戲

範例 3_3a 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

G304

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 3byte，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))

```

範例 9c 猜數字

有一種遊戲稱為幾 A 幾 B 的猜數字遊戲，兩個人對玩，互相猜對方預先寫下的四位數（四位數中的阿拉伯數字不可重覆），若所猜的數字與對方位置相同者為 A，數字相同，位置不對，則稱為 B。例如對方預寫的數字為 6713，若猜 6731 則應回應 2A2B，若猜 7652 則應回應 0A2B。試寫一程式，電腦自動產生四位數的亂數，再讓使用者猜的一種遊戲程式，電腦應逐一回應使用者已猜的狀況。其次，四位數亂數，數字不能重複，例如，1001、5566、1022 等都不行。

```

import random
c=[0,1,2,3,4,5,6,7,8,9]
#將以上數字打亂
for i in range(len(c)):
    d=random.randint(0,9)
    c[i],c[d]=c[d],c[i]
#取前四個數字,此即為亂數
e=str(c[0])+str(c[1])+str(c[2])+str(c[3])
print(e)
right=False
a=0
b=0
while (not right):
    f=input('輸入四位數：')
    for i in range(4):
        if f[i]==e[i]:
            a=a+1
        for j in range(4):
            if f[j]==e[i]:

```

```

        b=b+1
    b=b-a
    print('%d a %d b'%(a,b))
    if a==4:
        right=True
    a=0
    b=0
print("BinGo" )

```

※範例 9d 人工智慧與刪除法。

先將所有可能的答案通通列出來，再逐步將所有不可能的答案一一刪掉，最後剩下的值即為所求，此稱為刪除法。例如，同上題，剛剛是人腦猜，人腦有人腦的運算思維，但是電腦有電腦的運算思維，既然是電腦程式設計，那就要利用電腦的強項，才能事半功倍，以下示範電腦用刪除法來猜，使用者必須逐一回應電腦已猜的狀況。電腦猜數字若使用以上刪除法，則演算法如下：

- (1) 使用陣列列出 0000 至 9999 的四位數。
- (2) 逐一刪除阿拉伯數字重複者，例如 1001 或 3343 等。
- (3) 於剩下的可能數字中，挑最小的當作猜值，本例為 0123。(使用者應回應幾 A 幾 B，本例假設使用者的數字是 4237，所以回應 0A2B)
- (4) 於剩下的可能值中，使用臆測值，本例為 0123 逐一比對，將結果是 0A2B 者，蒐集起來，也就是不是 0A2B 者全部刪除。(為什麼呢？因為答案 4237 也會是 0A2B，將會被保留，此即為刪除法，逐一將不可能的答案，逐一刪除。)
- (5) 重複(3)、(4)兩個步驟，直到剩下最後一個，就是答案了，或使用者回應 4A0B 為止。

執行結果

下圖就是我預設 4237，電腦逐一猜值的過程。本例電腦 5 次就猜出結果，真的很強。其次，若您中間有任一過程回答錯了，例如，1045 應該 1B，您回答為 2B，那串列就會刪光，所以還要補足這一缺憾，避免程式當掉，這請讀者自行練習。

g03

程式列印

```

a=[]
b=[]
for i in range(10):
    for j in range(10):
        for k in range(10):
            for l in range(10):
                a.append(str(i)+str(j)+str(k)+str(l))

```

```

print(len(a))
for i in range(len(a)):
    print(a[i])
#逐一蒐集數字不重複者
for i in range(len(a)):
    flag=True
    c=a[i]
    for j in range(3):
        for k in range(j+1,4):
            if c[j]==c[k]:
                flag=False
                break
    if flag:#沒有重複才蒐集
        b.append(c)
for i in range(len(b)):
    print(b[i])
a=b[:]#串列的複製
b=[]#重新清為空串列
print(len(b))
right=False
while (not right):
    d=a[0]
    print ('Computer guess:%s' %d)
    a1,b1=eval(input('input na , nb:'))
    if a1==4 :
        right=True
        print("Bingo")
        break
#將相同 a 相同 b 的蒐集起來
for i in range(1,len(a)):
    a2=0
    b2=0
    f=a[i]
    for j in range(4):
        if f[j]==d[j]:
            a2=a2+1
    for k in range(4):
        if f[j]==d[k]:

```

```

        b2=b2+1
    b2=b2-a2
    if a1==a2 and b1==b2 :
        b.append(f)
    a=b[:]
    b=[]
12_6 人工智慧與十點半遊戲的製作

```

人工智慧與十點半遊戲的製作

專題學生：洪國勝

指導老師：洪國勝

高雄市立新莊高中

壹、摘要

分別使用 Python 的主控台與視窗環境製作具有人工智慧的撲克牌『十點半』遊戲。

貳、研究動機

高一資訊科技的程式設計，我們學了演算法，下學期的多元選修，我們學到了 Python 輸出入、決策、迴圈、陣列等程式設計工具，過年的時候，跟著家人玩撲克牌的『十點半』遊戲，我發現這一遊戲很有趣，也很適合將其電腦化，其中莊家補牌規則也很適合推廣人工智慧，所以激發我使用 Python 設計一個遊戲，讓人可以和電腦玩。

參、專題製作流程

一、十點半遊戲說明

1、十點半是使用撲克牌的紙牌遊戲，玩家和莊家各發一張牌，撲克牌每張牌的點數如下表：

牌面	1	2	3	4	5	6	7	8	9	10	J	Q	K
點數	1	2	3	4	5	6	7	8	9	10	0.5	0.5	0.5

例如，1C、1D、1H、1C 都是一點，J、Q、K 等穿西裝的牌都是半點。

1、 玩家先補牌，最多可補 4 張，補牌過程若有以下情況，遊戲直接結束（莊家不用補），直接計算結果。

2_1、玩家點數和超過『十點半』，玩家爆了，沒人賭資。

2_2、玩家點數和剛好『十點半』，玩家得到 2 倍賭金。

2_3、玩家順利補了 4 張，點數和沒有超過『十點半』，稱為玩家『五小』，玩家得到 2 倍賭金。

3、 玩家沒死，那莊家翻開自己的牌，考慮是否補牌，也是至多可以補 4 張，補的過程可直接翻開玩家的牌，比較大小，決定輸贏。（因為玩家『十點半』得到


```

p2=0#莊家點數和
m=100 #賭資
m2=""#牌型
m1=0#賠率
#牌面設定，使用 b 串列儲存此 52 張牌
b=[i for i in range(52)]
#洗牌
for i in range(52):
    r=random.randint(0,51)
    b[r],b[i]=b[i],b[r]
#輸出玩家第一張牌
p1=p[b[0]]#點數
print('玩家第一張牌是：',end='')
print (w[b[0]],p1)
j=1;#此牌是莊家的
over=False#結束否旗號
add=True#補牌否旗號
#詢問玩家是否補牌
c=input('補牌否？')
if c=='n':
    add=False
while add:
    j=j+1
    p1=p1+p[b[j]]#玩家點數和
    print ('第%d 張是%s,點數和累積是%3.1f'%(j,w[b[j]],p1))
    if p1>10.5 :
        m1=1
        m2='玩家爆了'
        over=True #遊戲結束了
        add=False
    elif p1==10.5:
        m1=-2
        m2='玩家十點半'
        over=True#遊戲結束了
        add=False
    elif j==5:
        m1=-2
        m2='玩家五小'

```

```

        over=True#遊戲結束了
        add=False
    else:
        c=input('補牌否? ')
        if c=='n':
            add=False
if not over:    #莊家開始補牌
    p2=p2+p[b[1]]
    print('莊家第一張牌是 %s,點數是%3.1f' %(w[b[1]],p2))
    p3=p1-p[b[0]]#玩家補牌的點數和，不含蓋著那一張
    k=1#莊家持牌張數
    m1=1
    while ( k<=4 and p2<10.5 and ((p2<=p3) or (p2<6))):
        j=j+1
        k=k+1
        p2=p2+p[b[j]]
        print("k:%d,%s,點數累積是：%3.1f" %(k,w[b[j]],p2))
    #評判結果
    if(p2>10.5):
        m1=-1
        m2='莊家爆'
    elif k==5:
        m1=1
        m2='莊家五小'
    elif p1>p2:
        m1=-1
        m2='玩家贏'
    else:
        m1=1
        m2='莊家贏'
#計算結果
print()
m=m-m1*10
print('結果是:%s'%m2)
print('賠率是:%d'%m1)
print('賭資剩:%d'%m)

```

8、執行結果如下

特色 8 含有簡易 APCS 考題

完全奇數 (APCS107/10 試題)

13311,13199 稱為完全奇數，請寫一程式，可以輸入一個整數N，並求出比此數大的完全奇數M 與比此數小的完全奇數P，並求其差的最小值，也就是 $| N - M |$ 與 $| N - P |$ 的最小值。例如，輸入N=13256，則M=13311；P=13199，所以輸出55。

程式列印

```
def judge(n):#定義一個函式，第八章會再詳細介紹
    while n>0:
        t=n %10
        if t %2 ==0:
            return False
        n=n//10
    return True
#主程式由此開始
n=13256
#往上找完全奇數
m=n
odd=judge(m)#使用 m 呼叫函式
while not odd:
    m=m+1
    odd=judge(m)
print('m=%d'%m)
p=n
odd=judge(p)
while not odd:
    p=p-1
    odd=judge(p)
print('p=%d'%p)
print(min(abs(n-m),abs(n-p))) #55
```

程式說明

judge 是一個函式，這一函式可以將傳來的數字從各位數逐一分解，我們採用一直除以 10 的餘數，就可以將數字一一分解，並解查所有數字是否有偶數，若找

直線

我們人類繪製直線是使用直尺，但電腦並沒有直尺，那電腦如何繪製直線呢？

答案就是要先建立直線方程式，然後再根據直線方程式一點一點密集繪製，這

些密集的點，看起來就是直線。例如，設有直線方程式如下：

$$y=ax+b$$

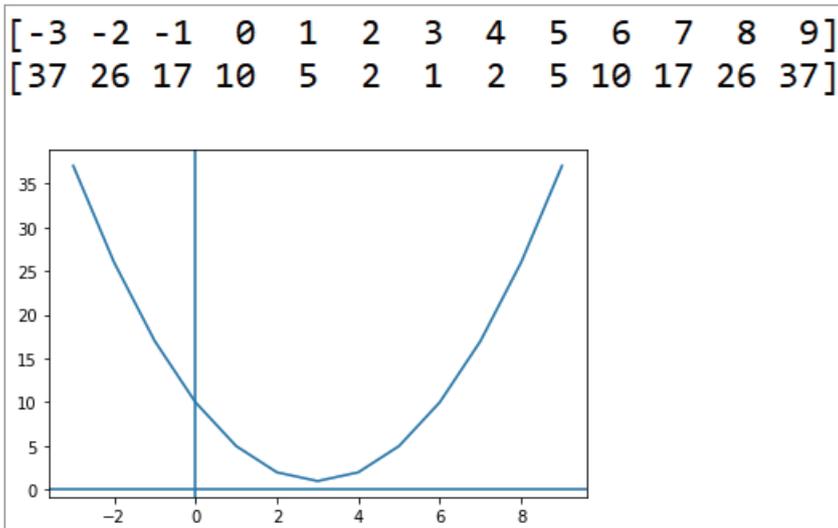
以下程式可繪出直線。

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-5, 5, 1) #從-5到5，間隔是1，不含結束點5，
且間隔可以實數
print(x)
y = 2*x + 1
print(y)
plt.plot(x,y)
plt.axhline(y=0)#x軸
plt.axvline(x=0)#y軸
plt.show()
```

以下程式可繪出二次曲線

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-3, 10, 1) #數字是我多次修正的結果，這樣可以表現圖形
最有變化的部分
print(x)
y = x *x -6*x +10
print(y)
plt.plot(x,y)
plt.axhline(y=0)#x軸
plt.axvline(x=0)#y軸
plt.show()
```

以上程式執行結果如下



特色 10 含有推甄課程學習成果範例

7_3 課程學習成果

111 學年度學測個人申請入學需要提供『課程學習成果』，108 課綱強調『素養導向』，什麼是素養導向？簡單的說就是打破各科疆界，把知識應用在實際的現實生活中，也就是學生要能將各科的學習融會貫通，統整在一起。例如，本書就很強調跨領域學習，所以書中不斷幫大家解決數學、英文等問題，課程學習成果當然是把大家的作業與報告加深或加廣，以下是我提供的『課程學習成果』參考範例。

開根號計算之研究
 專題學生：洪國勝
 指導老師：洪國勝
 高雄市立新莊高中

壹、摘要

分別使用循序法、二分猜值法、遞迴法、與二項式定理，配合 Python 程式設計工具，解出開根號運算。

貳、研究動機

於高一資訊科技的演算法我們學到了『循序法、二分猜值法、遞迴法』、於國中數學我們學到『二項式』定理，於多元選修我學到『Python』程式設計，由於對資訊科學非常有興趣，所以我將開根號的四個演算法，以 Python 程式設計實現，希望作為申請入學課程學習成果的依據。

參、專題原理

一、Python 語法

算術運算子(Arithmetic operators)

下表是 Python 的算術運算子列表：

運算子	定義	優先順序	結合律
-----	----	------	-----

**	次方	1	由左至右
+/-	正負號	2	由右至左
*	乘法運算	3	由左至右
/	除法，得實數商	3	由左至右
//	除法，得整數商	3	由左至右
%	求餘數(Modulus)	3	由左至右
+/-	加法/減法運算	4	由左至右
=	指派	14	由右至左

例如，

```
a=5;b=2
print(a**b)
```

結果是 25。

決策運算

Python 的決策指令是『if~else~』，其語法如下：

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

例如，以下程式可評判成績是否及格。

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

迴圈運算

Python 迴圈有兩種，分別是 for 與 while，for 用於程式設計階段已知執行次數，例如，以下程式可進行乘法運算。

```
a=5
b=8
s=0
for b in range(1,b+1):
    s=s+a
print(s)#40
```

while 用於設計階段不知迴圈要重複幾次。例如，除法運算是，只要被除數大於除數，就要重複執行

```
被除數=被除數-除數
商=商+1
```

所以程式如下：

```

a=8#被除數
b=3#除數
q=0#商
while(a>=b):#只要(被除數>除數) 就執行迴圈
    a=a-b    # (被除數)-(除數)
    q=q+1    #商每次遞增 1
print(q)# 商數
print(a)#餘數

```

串列

前面的 **a,b,c** 適用少量的資料處理，串列則適用於大數據分析的資料結構。例如，若有資料 5 筆如下：

```
3,5,6,8,1
```

若使用 5 個變數儲存，那計算總和、極大與極小等問題，程式會很冗長，若使用串列儲存如下：

```
a=[3,5,6,8,1]
```

這樣就一口氣宣告與指派 **a[0]**、**a[1]**、**a[2]**、**a[3]**、**a[4]**等 5 筆資料。其次，串列可配合迴圈，那就可大批處理資料。例如，以下迴圈就可求其和。

```

a=[3,5,6,8,1]
s=0
for i in range(len(a)):
    s=s+a[i]
print(s)

```

二、循序法

循序法是由小而大，逐一將所有可能的答案通通代入。例如，以下程式就可解出開根號整數解。

```

a=9
for i in range(9):
    if i*i>=a:
        print(i)#3
        break

```

以下程式，可解出開根號實數解。

```

a=9
s=10#放大倍數
for i in range(a*s):
    if (i/s)*(i/s)>=a:
        print(i/s)#3.0
        break

```

三、二分猜值法

前面循序法是一個一個猜，每猜一個，猜值範圍僅減少一個；二分猜值法則是每次猜其一半，這樣每次就可將猜值範圍減少一半。使用二分猜值法解平方根的程式如下：

```
y=9.0
x1=0.0
x2=y
n=1
while (abs(x1-x2)>0.01):#距離>0.01,繼續猜
    print("%d:x1=%2.2f,x2=%2.2f" %(n,x1,x2))
    x=(x1+x2)/2#每次都猜中間
    t=x*x#計算結果
    if t<y :#猜的太小
        x1=x #調整下界
    else:
        x2=x#調整上界
    n=n+1
print(x)#2.99
```

四、遞迴

前面的循序法、二分猜值法都是正向思考，遞迴則是一種反向思考，強調的是，先解決一個，或一部份，而剩下的竟然和原方法相同，這樣是否可繼續使用原方法呢，答案是肯定的，這就是電腦所提供的『遞迴』工具，這樣就可以輕鬆解決問題。以下是以『遞迴』法解開根號的程式。

```
def f(x,x1,x2):
```

```
    if abs(x1-x2)<0.01:
```

```
        return(x1)
```

```
    else:
```

```
        y=(x1+x2)/2
```

```
        t=y*y
```

```
        if t<x:#猜的太小(此時，只是改變猜值範圍而已，方法還是一樣，所以繼續呼叫原方法)
```

```
            return(f(x,y,x2))
```

```

else:# 猜的太大
    return(f(x,x1,y))
a=9
x=a
x1=0#下界
x2=a#上界
print(f(x,x1,x2))#2.997

```

五、二項式定理

二項和的二項式定理如下：

$$(10a+b)^2=100a^2+20ab+b^2=100 a^2+b(20a+b)$$

以 138384 為例，開根號運算過程如下：

1. 由以上 $100 a^2+b(20a+b)$ ，表示我們由左到右，每兩個一組，先找a，再找b。
2. 找出a。從9,8,7到1 的平方依序找出小於等於最左邊那一組的數字(本例是13)，因為 $81(9*9),64(8*8),49,36,25,16$ 都太大，所以找到3，並扣掉3 的平方9，剩下4，請看下圖運算步驟1。

3. 使用迴圈，從第二組數字開始逐一找 $b_1b_2b_3\dots$ 。

(1) 計算每一次的餘數。餘數 $d = \text{前面餘數} \times 100 + \text{這一組數字}$ ，本例d是483。

(2) 用迴圈找b。b 從9,8,7到1，找 $b(20a + b)$ 小於餘數d。例如，本例a是3，d是483

$$b=9,9*(20*3+9)=621 \text{ 太大，不行}$$

$$b=9,8(20*3+8)=544 \text{ 太大，不行}$$

$$b=7,7(20*3+7)=469 \text{ 已經小於483，所以可以，請看下圖步驟2。}$$

(3) 餘數d 扣掉 $b(20a + b)$ 。 $d = d - b(20a + b)$ 。

5. 輸出 $ab_1b_2b_3\dots$, 即為所求。

		3	7	2
1	法則 3	1	3	8
		9	3	8
		4	8	3
2	$3 \times 20 + 7 = 67$ $67 \times 7 = 469$	4	6	9
3	$37 \times 20 + 2 = 742$ $742 \times 2 = 1484$		1	4
			8	4
			1	4
			8	4
4				0

6、以下是 Python 程式列印

```

aa=138384
bb=[0]
i=0
#每兩個數字一組
while aa>0:
    i=i+1
    bb.append(aa %100)
    aa=aa//100
print(bb)
n=i
print(n)
#找出 a
d=bb[n]
a=10
f=100
while f>d:
    a=a-1
    f=a*a
d=d-f
print('1. a=%d,d=%d'%(a,d) )
n=n-1
print('n=%d'% n)
#找 b
for i in range(n,1-1,-1):

```

```
d=d*100+bb[i]
b=10
print('2. a=%d,d=%d'%(a,d) )
f=10000
while f>d:
    b=b-1
    f=b*(20*a+b)
    d=d-f #d 是每次扣剩的
    a=a*10+b
print('平方根是： %d'% a)
print('d=%d'% d)
```

肆、心得

程式設計是一個學習數學的好工具，自己把所有演算法，用程式設計工具實現，真的可以更加瞭解數學，感謝洪老師的指導，讓我使用程式設計工具解決很多數學問題，更加強我往資訊科學研究的動力。

六、參考資料

1、 Python 程式設計入門教材，洪國勝，泉勝 108 年出版。

七、延伸專題

請完成開立方的計算。