



# 數學解題程式設計

# 目錄

1 動機或期待

2 學習與收穫

3 心得與省思

4 課程活動

5 個人作品

——→ 參數式

——→ 阿基米德螺線

——→ 隨機漫步

——→ 蕨葉圖形製作

### ➤ 選課動機或期待

因為以往從沒做過程式設計，再加上高中後看過了些關於寫程式的影片，**想嘗試看看**，所以選擇了這門課程。

### ➤ 學習與收穫

課程中，藉由老師發給我們的講義，循序漸進地學習，從認識基本的程式語言開始，如: #作為註解用，並不會執行、for為循環語句、break為中止循環語句、print為輸出等，先按照講義上的指令打，**漸漸熟悉後，開始自己改編或撰寫程式**。

其中我印象最深刻的單元是阿基米德螺線的單元，藉由繞圓圈的圈數以及座標位置的改變，能夠設計出各種不同的圖案，很有趣，像是利用**阿基米德螺線**設計出棒棒糖的圖形，雖然剛開始做出來的棒棒糖和棒子的圖案沒有合起來，但在試著改變座標位置後，終於完成圖型。課程中不但訓練了我的**耐心**以及**邏輯思考的能力**，在檢查所寫的程式時也**更加地細心，讓自己能夠靜下來完成事情**。

### ➤ 心得與省思

課程中的數學題目雖然平常寫來覺得不難，但打成程式語言卻很不一樣，光少了一個空格就可能有錯誤，所以每個格式都需仔細檢查與編排。

因為我**喜歡動手做事情**，在課程中也學到許多，像是幾何圖形、序列問題、循環語句、函數、機率...等的程式，雖然會有遇到困難的時候，**但每當自己一步步努力地完成程式時，真的很有成就感**，未來還是會練習程式設計，訓練自己。

## ➤ 課程活動

### 約瑟夫環狀排列問題 (Josephus Problem)

P65

```
N=10 #初始全部的人數
M=N-2 #自殺的人數
d=3 #從1報數到d的值

A=[i for i in range(1,N+1,1)]
B=[]
print ("初始排列:",A) #檢查用,之後可加#

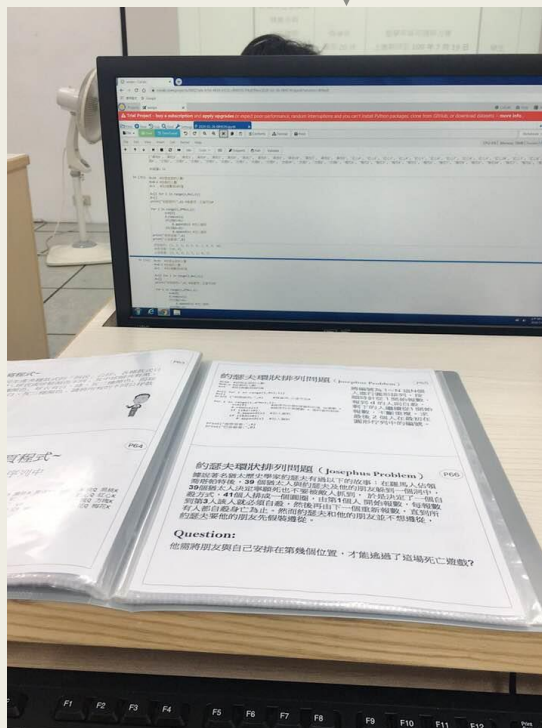
for i in range(1,d*M+1,1):
    x=A[0] #將序列中第1個資料的值 給變數 x
    A.remove(x) #將序列中有變數 x 值的資料移除
    if (i%d!=0):
        A.append(x) #加入資料
    if (i%d==0):
        B.append(x) #加入資料

print("尚存活者:",A)
print("已自殺者:",B)
```

將編號為 1~N 這N個人進行圓形排列，按順時針從 1 開始報數，報到 d 的人則自殺，剩下的人繼續從1 開始報數，不斷重複。求最後 2 個人在最初在圓形佇列中的編號。

講義

### 課程過程



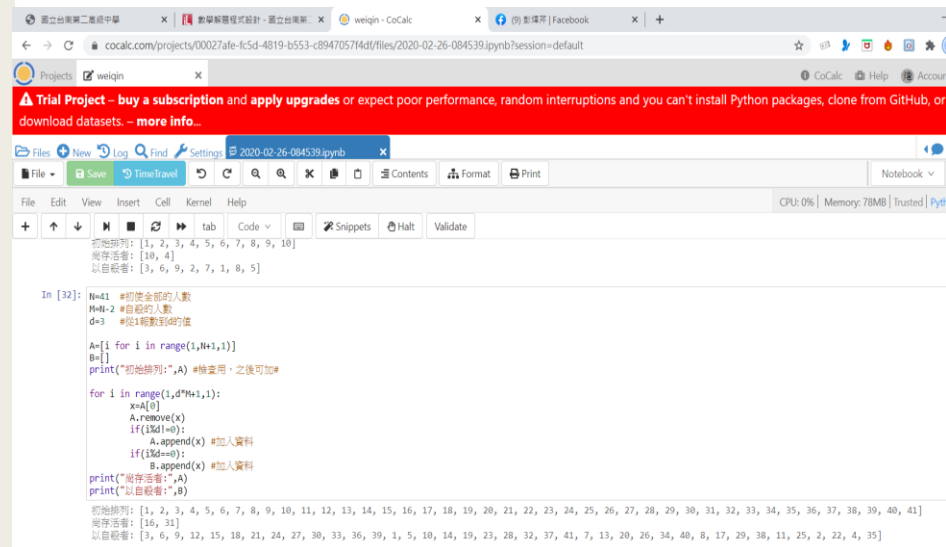
### 約瑟夫環狀排列問題 (Josephus Problem)

P66

據說著名猶太歷史學家約瑟夫有過以下的故事：在羅馬人佔領喬塔帕特後，**39** 個猶太人與約瑟夫及他的朋友躲到一個洞中，**39**個猶太人決定寧願死也不要被敵人抓到，於是決定了一個自殺方式，**41**個人排成一個圓圈，由第**1**個人 開始報數，每報數到第**3**人該人就必須自殺，然後再由下一個重新報數，直到所有人都自殺身亡為止。然而約瑟夫和他的朋友並不想遵從，約瑟夫要他的朋友先假裝遵從。

### Question:

他需將朋友與自己安排在第幾個位置，才能逃過了這場死亡遊戲？



實作

## ➤ 個人作品

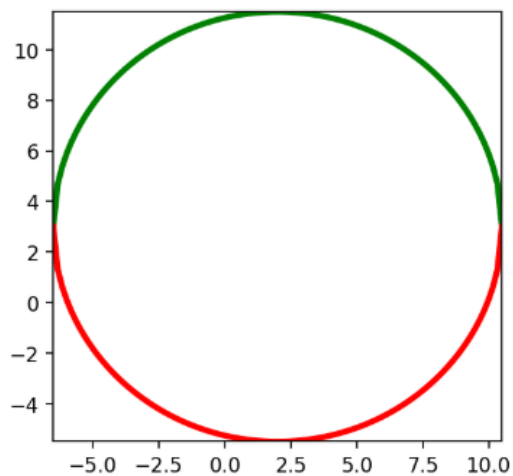
### → 參數式

```
In [24]: import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(4,4),dpi=80)

(h,k)=(2,3) #圓心座標
r=8         #半徑長

x=np.linspace(h-r,h+r,101)
y=k+(r**2-(x-h)**2)**0.5
plt.plot(x,y,color="green",linewidth=3)
x=np.linspace(h-r,h+r,101)
y=k-(r**2-(x-h)**2)**0.5
plt.plot(x,y,color="red",linewidth=3)
plt.xlim(h-r-0.5,h+r+0.5)
plt.ylim(k-r-0.5,k+r+0.5)

plt.show()
```



```
In [71]: import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(4,4),dpi=80)

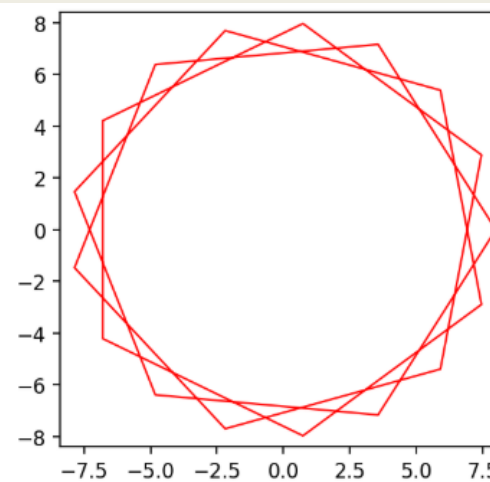
pi=3.14159 #圓周率
r=8         #半徑長
N=3         #繞幾圈
M=17        #表示幾個角
a=0         #起始的偏移角度

t=np.linspace(a+0,a+360*N,M+1) #參數
t=t*pi/180 #參數，單位轉換
x=r*np.cos(t) #第一座標
y=r*np.sin(t) #第二座標

plt.plot(x,y,color="red",linewidth=1)

plt.xlim(-1.05*r,1.05*r)
plt.ylim(-1.05*r,1.05*r)

plt.show()
```



## → 阿基米德螺線

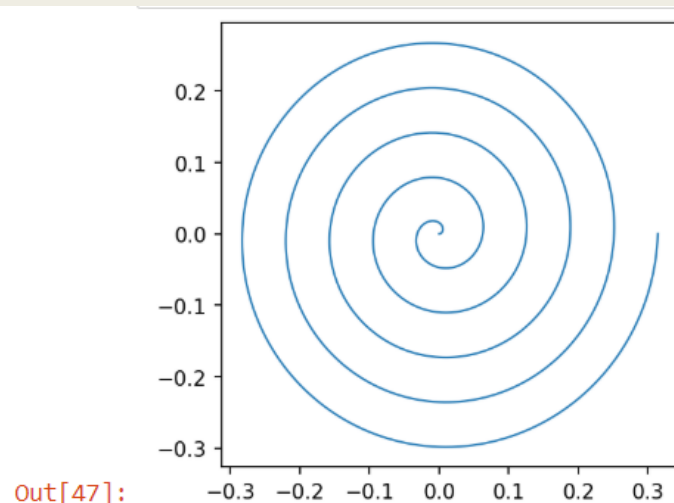
```
In [47]: import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(4,4),dpi=80)

pi=3.14159          #圓周率
N=5                 #繞幾圈

t=np.linspace(0,360*N,1001)  #參數
t=t*pi/180          #參數，單位轉換
r=0.01*t
x=r*np.cos(t)        #第一座標
y=r*np.sin(t)        #第二座標

plt.plot(x,y,linewidth=1)

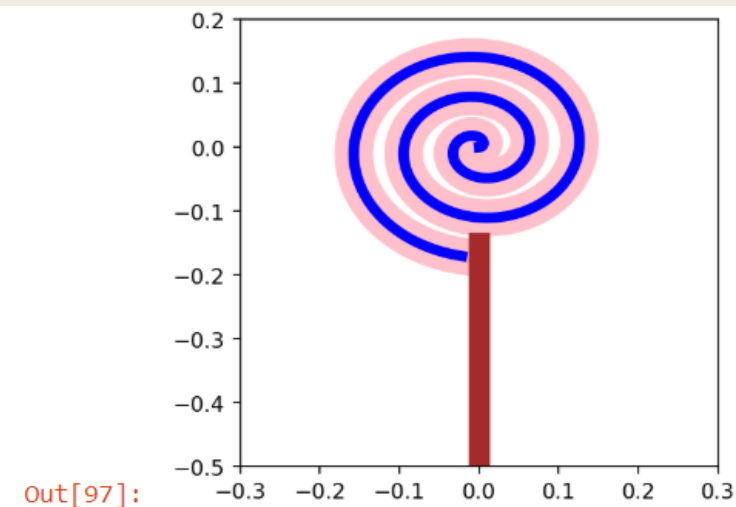
plt.show()
```



```
In [97]: import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(4,4),dpi=80)
pi=3.14159          #圓周率
N=2.73              #繞幾圈

t=np.linspace(0,360*N,1001)  #參數
t=t*pi/180          #參數，單位轉換
r=0.01*t
x=r*np.cos(t)        #第一座標
y=r*np.sin(t)        #第二座標
plt.plot(x,y,color="pink",linewidth=18)

t=np.linspace(0,360*N,1001)  #參數
t=t*pi/180          #參數，單位轉換
r=0.01*t
x=r*np.cos(t)        #第一座標
y=r*np.sin(t)        #第二座標
plt.plot(x,y,color="blue",linewidth=5)
X=[0,0]
Y=[-0.15,-0.6]
plt.plot(X,Y,color="brown",linewidth=10)
plt.xlim(-0.3,0.3)
plt.ylim(-0.5,0.2)
plt.show()
```

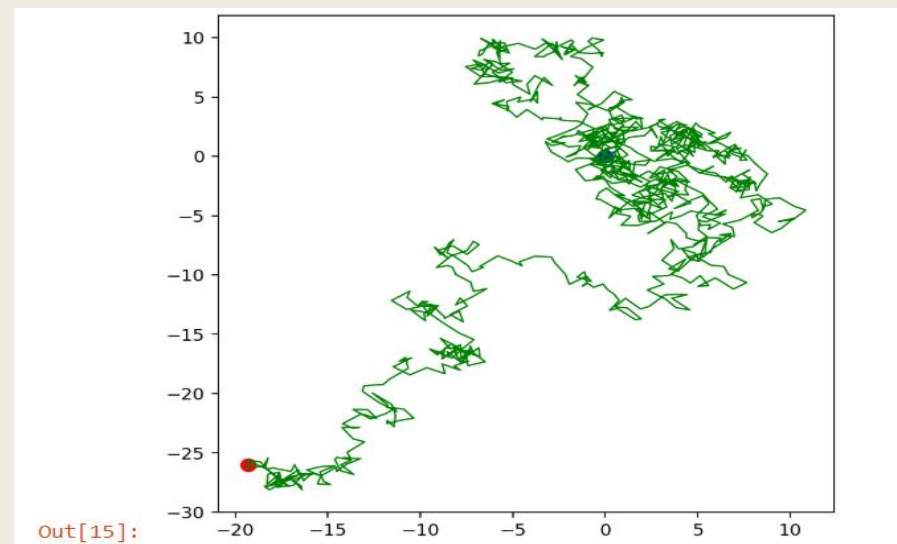


## → 隨機漫步

```
In [15]: import matplotlib.pyplot as plt
import random
plt.figure(figsize=(6,6),dpi=80)
N=1000 #隨機次數
X=[0]
Y=[0]

for i in range(1,N+1,1):
    u=X[i-1]+3*(random.random()-0.5) #輸出-1~1之間的隨機整數
    v=Y[i-1]+3*(random.random()-0.5) #輸出-1~1之間的隨機整數
    X.append(u)
    Y.append(v)
plt.scatter(X[0],Y[0],color="blue",linewidth=3) #起點
plt.plot(X,Y,color="green",linewidth=1) #路徑
plt.scatter(X[N],Y[N],color="red",linewidth=3) #終點

plt.show()
```



概念:

利用機率性質來進行隨機漫步，  
設定1代表向右或向上，0代表不動，-1代表向左或向下，  
以這種方式使用電腦亂數指令就可以模擬隨機漫步。

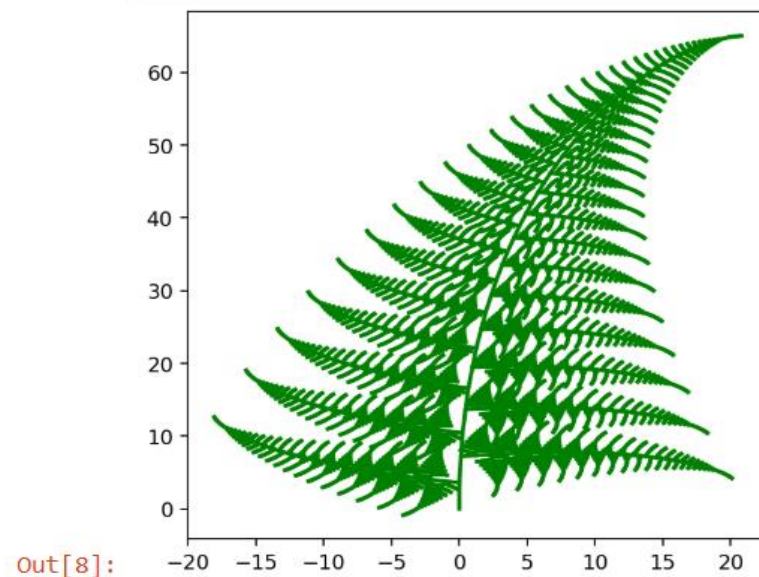
## → 蕨葉圖形製作

```
In [8]: import numpy as np
import matplotlib.pyplot as plt

plt.figure(figsize=(5,5),dpi=80)
pi=3.14159

def F(x0,y0,r,a):
    if r < 1 :return
    x1=x0+0.04*r*np.cos(a);y1=y0+0.04*r*np.sin(a)
    x2=x1+0.04*r*np.cos(a-pi/90);y2=y1+0.04*r*np.sin(a-pi/90)
    X=[x0,x1,x2]
    Y=[y0,y1,y2]
    plt.plot(X,Y,color="green",linewidth=2)
    F(x1,y1,0.3*r,a+pi*4/9)
    F(x2,y2,0.3*r,a-pi*41/90)
    F(x2,y2,0.9*r,a-pi/90)

F(0,0,90,pi/2)
```



概念:

利用函數與座標系統製作蕨葉圖形，  
設計圖形之函數值並加以製作。