

第一題

1. 繪製下圖（線條顏色、符號與數量都可以由程式輕易變更）

注意事項：

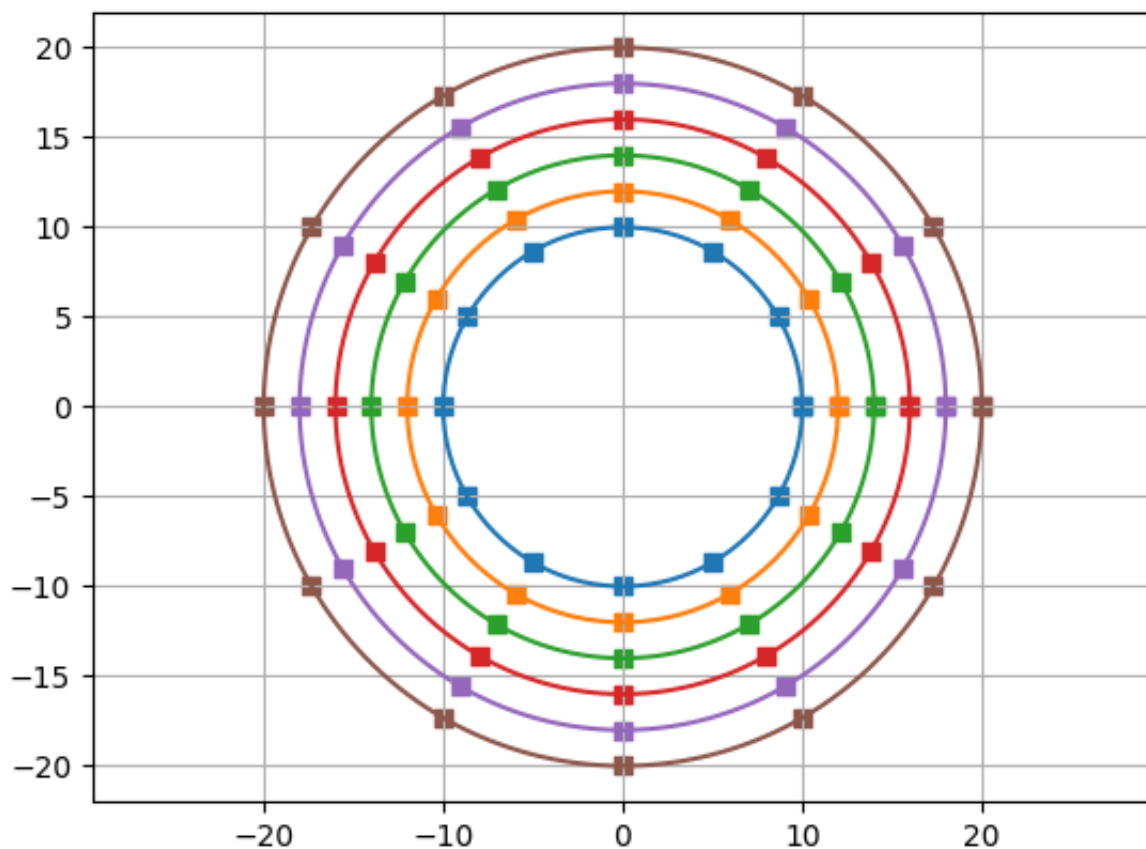
- 將數量的決定放在 code 第一條，譬如， $n = 6$
- 將符號的決定放在 code 第二條，譬如，`marker = 's'`
- 不論畫同心圓或圓上方塊符號，可以採用迴圈方式或非迴圈的矩陣計算方式。最好兩者都試試看（從自己最有把握的方法先做），才知道 python 的長處（可以先不論顏色）。
- 上述方式可以迅速變更設定，看到結果。

Getting started

In []: # Q1

```
from turtle import color
import numpy as np
import matplotlib.pyplot as plt
from parsing import col
colors = ['r', 'g', 'b']
xmin, xmax = -20, 20
r = np.linspace(10, 20, 6)
theta = np.linspace(0, 2*np.pi, 200)
fig = plt.figure()
axes = fig.add_subplot(111)
axes.axis('equal')
for i in r:
    x = i * np.cos(theta)
    y = i * np.sin(theta)
    plt.plot(x, y)
theta1 = np.linspace(0, 2*np.pi, 13)
for i in r:
    x = i * np.cos(theta1)
    y = i * np.sin(theta1)
    plt.scatter(x, y, marker='s')
plt.xlim(-20, 20), plt.grid(True)
```

Out[]:((-20.0, 20.0), None)



2. 繪製下圖（線條顏色與數量都可以由程式輕易決定）

注意事項：

- 將方框數量的決定放在 code 第一條，譬如， $n = 8$ ，改變 n 值，便能看到結果的改變，譬如， $n = 128$ 得到右圖。

Getting started

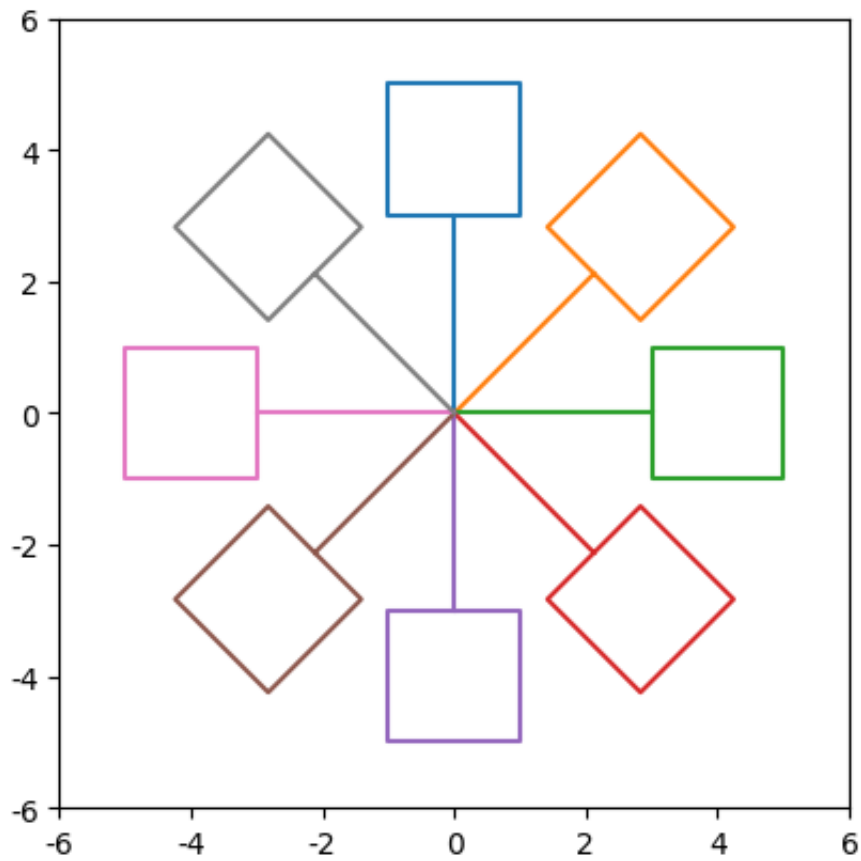
Getting started

In []:# Q2

```
import numpy as np
import matplotlib.pyplot as plt
figure, ax = plt.subplots()
ax.set_xticks(np.array([-6, -4, -2, 0, 2, 4, 6]))
ax.set_xticklabels(['-6', '-4', '-2', '0', '2', '4', '6'], fontsize=10)
ax.set_yticks(np.array([-6, -4, -2, 0, 2, 4, 6]))
ax.set_yticklabels(['-6', '-4', '-2', '0', '2', '4', '6'], fontsize=10)
ax.set_xlim([-6, 6])
ax.set_ylim([-6, 6])
ax.grid(False)
ax.set_aspect(1)
x = np.array([0, 0, -1, -1, 1, 1, 0])
y = np.array([0, 3, 3, 5, 5, 3, 3])
m = np.array([x, y])
```

$n = 8$

```
for i in range(n):
    tm = np.array([[np.cos(2*i*np.pi/n), np.sin(2*i*np.pi/n)],
                   [(-1)*np.sin(2*i*np.pi/n), np.cos(2*i*np.pi/n)]]).dot(m)
    plt.plot(tm[0,:], tm[1,:])
```

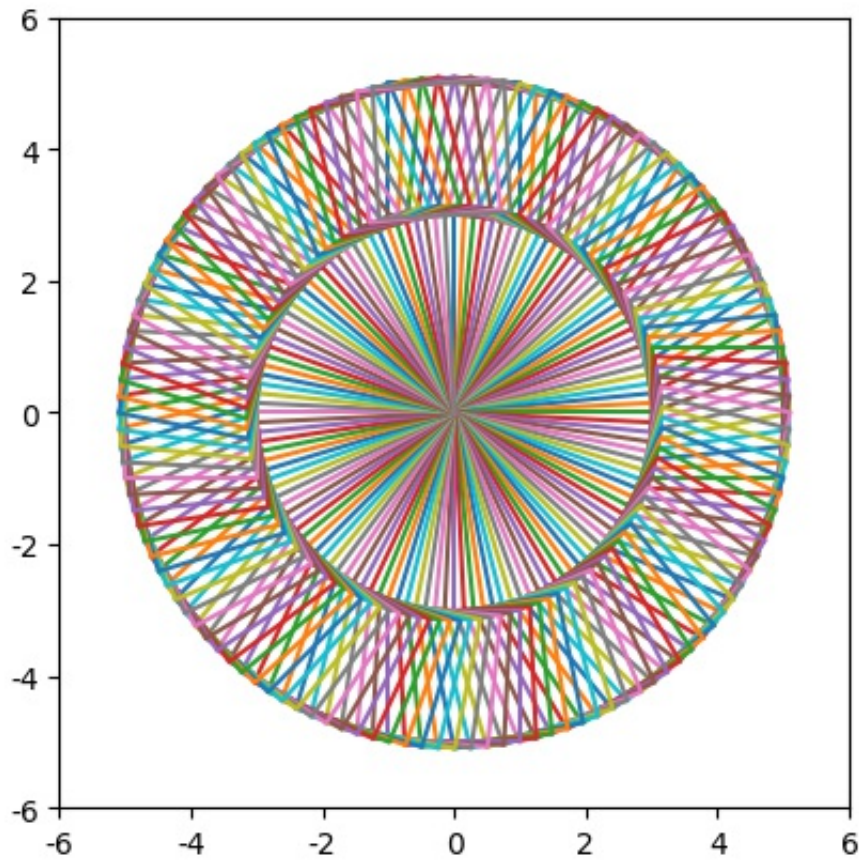


```

In [ ]:figure, ax = plt.subplots()
        ax.set_xticks(np.array([-6, -4, -2, 0, 2, 4, 6]))
        ax.set_xticklabels(['-6', '-4', '-2', '0', '2', '4', '6'], fontsize=10)
        ax.set_yticks(np.array([-6, -4, -2, 0, 2, 4, 6]))
        ax.set_yticklabels(['-6', '-4', '-2', '0', '2', '4', '6'], fontsize=10)
        ax.set_xlim([-6, 6])
        ax.set_ylim([-6, 6])
        ax.grid(False)
        ax.set_aspect(1)
        x = np.array([0, 0,-1,-1, 1, 1, 0])
        y = np.array([0, 3, 3, 5, 5, 3, 3])
        m = np.array([x,y])

        n = 128
        for i in range(n):
            tm = np.array([[np.cos(2*i*np.pi/n),np.sin(2*i*np.pi/n)],
                           [(-1) *np.sin(2* i * np.pi/n),np.cos(2*i*np.pi/n)]]).dot(m)
            plt.plot(tm[0,:],tm[1,:])

```



第三題

計算如下右圖的卡方右尾面積與自由度對照表，並輸出到 EXCEL 檔，檔名為：Chi2Table.xlsx，含欄與列的名稱。

```
In [ ]:#Q3
from scipy.stats import chi2
import pandas as pd
import numpy as np
import math

#sol1
# F = np.array([0.995, 0.99, 0.975, 0.95, 0.9, 0.1, 0.05, 0.025, 0.01,0.005])
# for i in F:
#     for j in range(1,16):
#         x = chi2.ppf(1-i, j)
#         print(x)

#sol2
df=np.linspace(1, 15, 15)
F=np.array([[0.995], [0.99], [0.975], [0.95], [0.9], [0.1], [0.05], [0.025], [0.01], [0.005]])
x = chi2.ppf(1-F, df)
A = pd.DataFrame(x)
B=A.T
B.columns=["0.995", "0.99", "0.975", "0.95", "0.9", "0.1", "0.05", "0.025", "0.01", "0.005"]
B.index=[np.linspace(1, 15, 15)]
B

file_name = 'Chi2Table.xlsx'
B.to_excel(file_name)
```