

姓名:劉韶颺

系級:地科學士一

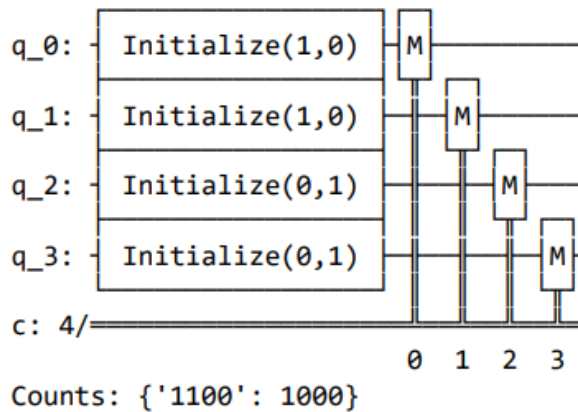
學號: 110605006

- ✓ 請標記所寫題號以及截圖執行結果(執行結果長條圖 or 印出計次數 or 布洛赫球面圖)，截圖後請附上適當文字敘述輔助說明
- ✓ 如果所選題目為手寫題，請將過程清楚寫下並拍照放上
- ✓ 每章任選一題，1.5 , 2.5 為加分題，可自行選擇要不要填寫

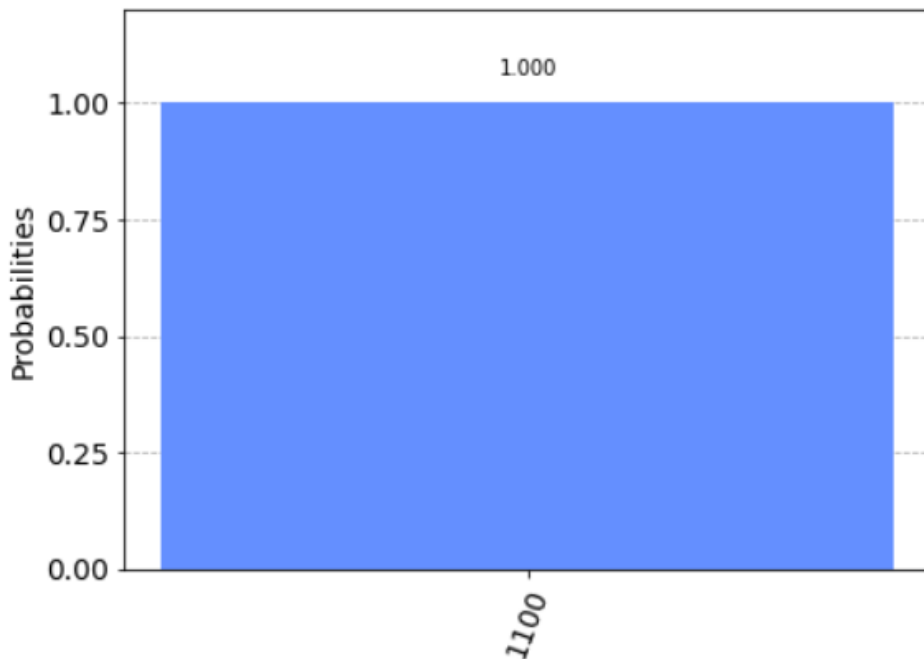
範例：（老師練習 #Program 2.3）

（注意！截圖時請一併印出量子線路及機率狀態圖）

內容：



Out[5]:



建構一個具有 4 個量子位元的量子線路，並使用量子位元的狀態向量來設定這 4 個量子位元不同初始值(狀態)，最後針對這 4 個量子位元進行測量之後儲存於 4 個古典的位元中。然後我們將這個量子線路透過量子電腦模擬器執行 1000 次，並繪製出這 1000 次的模擬結果，來看出不同量子位元測量的值為 0 或是 1 的機率。

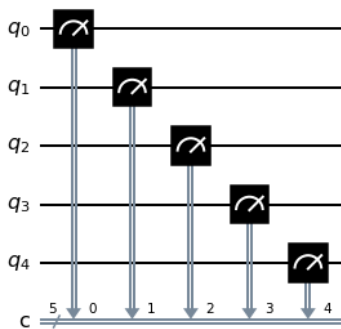
## 第一章：1-1

內容：

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(5, 'q')
creg_c = ClassicalRegister(5, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[2], creg_c[2])
circuit.measure(qreg_q[3], creg_c[3])
circuit.measure(qreg_q[4], creg_c[4])
circuit.draw(output='mpl')
```



說明：此程式是申請完IBM帳號後使用創作者界面，直接用拉圖片式的方式製作而程的量子程式。

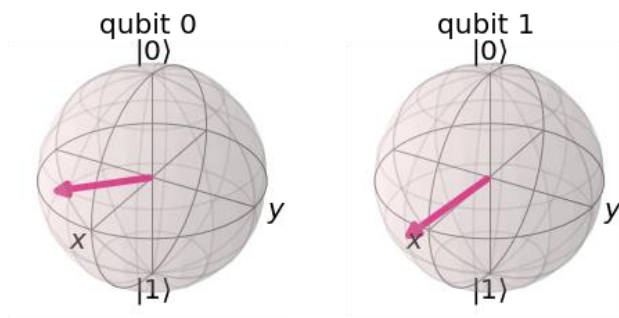
首先import量子程式的Library:qiskit, 用QuantumRegister建立一個量子位元、ClassicalRegister建立古典位元、QuantumCircuit建立量子電路，再來用measure來將量子位元測量後放到古典位元內。

## 第二章：2-3

內容：

```
from qiskit import *
import math
qc = QuantumCircuit(2,2)
qc.initialize([1/3+(2/3)*(1j),math.sqrt(3)/3+(1/3)*(1j)],0)
qc.initialize([1/5-(2/5)*(1j),-2/5-(4/5)*(1j)],1)

qc.draw()
from qiskit.quantum_info import Statevector
state=Statevector.from_instruction(qc)
state.draw('bloch')
```



說明：

首先import量子程式的Library:qiskit, 用QuantumRegister建立一個量子位元、QuantumCircuit建立量子電路, 用initialize設定一量子位元的初始狀態。接著import statevector來使用可以描繪布洛赫球面的函數。

### 第三章：3-2

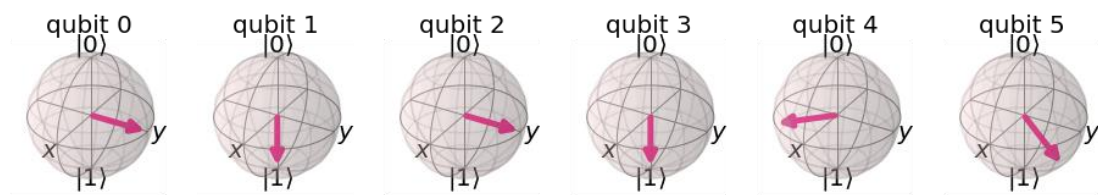
内容：

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(6, 'q')
creg_c = ClassicalRegister(6, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.rx(pi/2, qreg_q[0])
circuit.p(pi/4, qreg_q[1])
circuit.rx(pi/2, qreg_q[2])
circuit.p(pi/4, qreg_q[3])
circuit.rx(pi/2, qreg_q[4])
circuit.p(pi/2, qreg_q[5])
circuit.x(qreg_q[0])
circuit.y(qreg_q[1])
circuit.z(qreg_q[2])
circuit.s(qreg_q[3])
circuit.t(qreg_q[4])
circuit.h(qreg_q[5])
circuit.u(pi, 0, 0, qreg_q[3])
circuit.u(pi, pi/2, 0, qreg_q[4])
circuit.u(pi, pi/2, pi/4, qreg_q[5])
circuit.h(qreg_q[5])
circuit.draw()

#from qiskit.quantum_info import Statevector
#state=Statevector.from_instruction(circuit)
#state.draw('bloch')
```



說明：

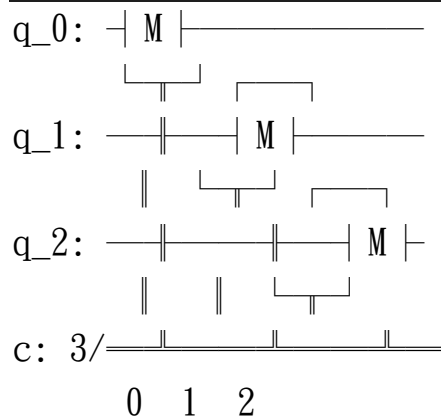
首先import量子程式的Library:qiskit, 用QuantumRegister建立一個量子位元、ClassicalRegister建立古典位元、QuantumCircuit建立量子電路，再來用.rx, .x, .h, p, u, t, s, z等等函數來設計題目所需的量子電路。

(選填) 加分題 : 1-5

內容 :

#1-5

```
from qiskit import QuantumCircuit, execute
from qiskit.providers.aer import AerSimulator
sim = AerSimulator()
qc = QuantumCircuit(3, 3)
qc.measure([0], [0])
qc.measure([1], [1])
qc.measure([2], [2])
print(qc)
job=execute(qc, backend=sim, shots=1000)
result = job.result()
counts = result.get_counts(qc)
print("Total counts for qubit states are:",counts)
```



Total counts for qubit states are: {'000': 1000}

說明 :

第1行為程式編號註解。

第 2 行使用 import 敘述引入 qiskit 套件中的 QuantumCircuit 類別、transpile 函數以及

`execute` 函數。

第3行使用`import`敘述引入`qiskit.providers.aer`中的`AerSimulator`類別。

第4行使用`AerSimulator()`建構IBM QASM量子電腦模擬器物件，儲存於`sim`變數中。

第5行使用`QuantumCircuit(3,3)`建構一個包含3個量子位元及3個古典位元的量子線路物件，儲存於`qc`變數中。

第6行使用`QuantumCircuit`類別的`measure`方法在量子線路中加入測量單元  
測量索引值為0的量子位元，並將測量結果儲存於索引值為0的古典位元。

第7行使用`print(qc)`呼叫`print`函數以文字模式顯示量子線路。

第8行呼叫`execute`函數建立一個工作，儲存於`job`變數中，其中傳入參數`qc`表示要執行的量子線路，

`backend=sim`設定在後端使用`sim`物件所指定的量子電腦模擬器，`shots=1000`設定在後端量子電腦模擬器上

執行量子線路`qc`共1000次，而每次執行都測量量子位元並將測量結果儲存於古典位元中。

第9行使用`job`物件的`result`方法取得`job`物件的執行相關資訊，儲存於物件變數`result`中。

執行相關資訊除了執

行環境之外，也包括執行結果，也就是量子線路在量子電腦模擬器上的執行結果。

第10行使用`result`物件的`get_counts(qc)`方法取出有關量子線路量測結果的計數  
(`counts`)，並以字典(`dict`)型別  
儲存於變數`counts`中。

第11行使用`print`函數顯示"Total counts for qubit states are :"`字串`及字典型別變數  
`counts`的值，在這個程式中  
`counts` 變數的值為`{'0': 1000}`，也就是測量結果為'0'的計數結果為 1000 次。