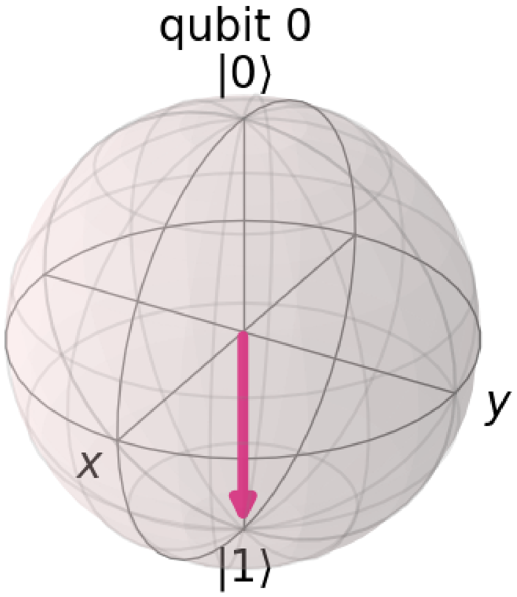
Communication System Lab1

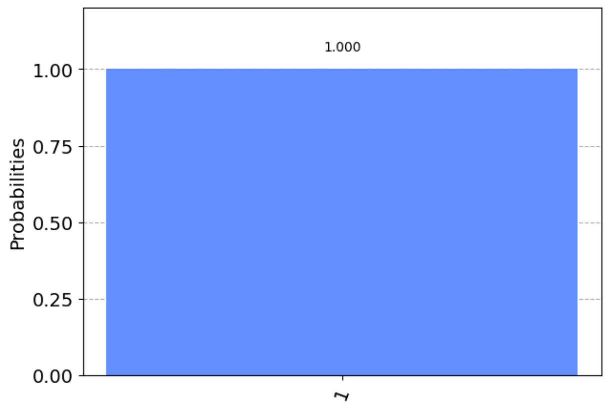
學號：B08502141　　　　名：石旻翰　　　　　系級：電機四

1 Problem1

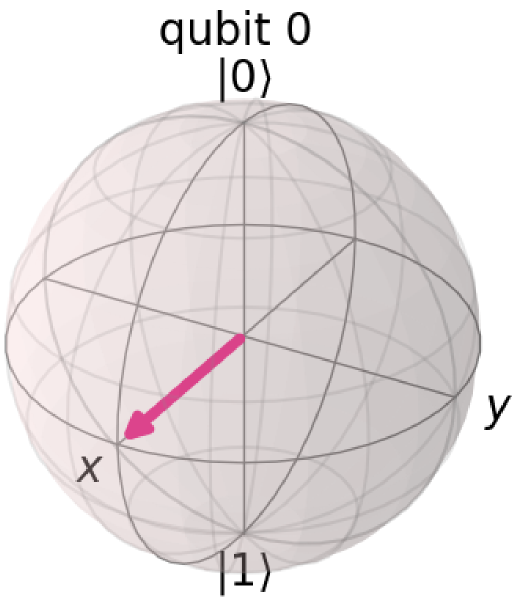
1.1 1-(e)

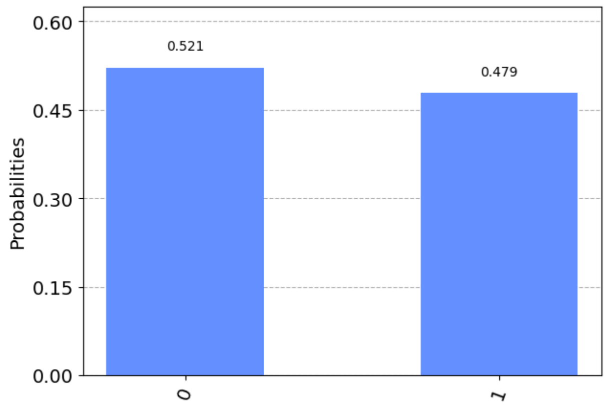
Statevector([0.+0.j, 1.+0.j],dims=(2,))



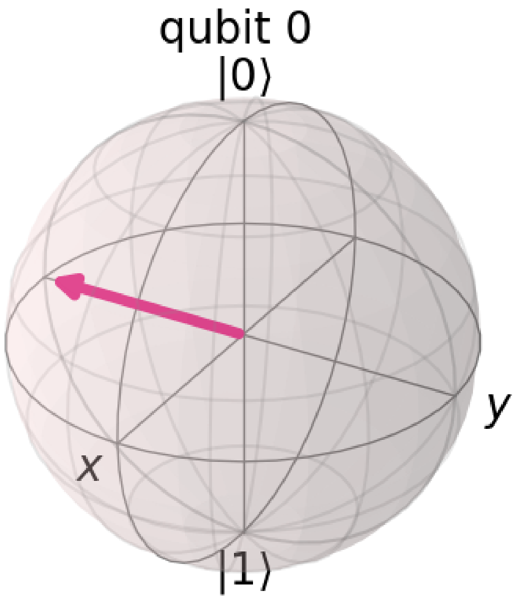


1.2 1-(f)



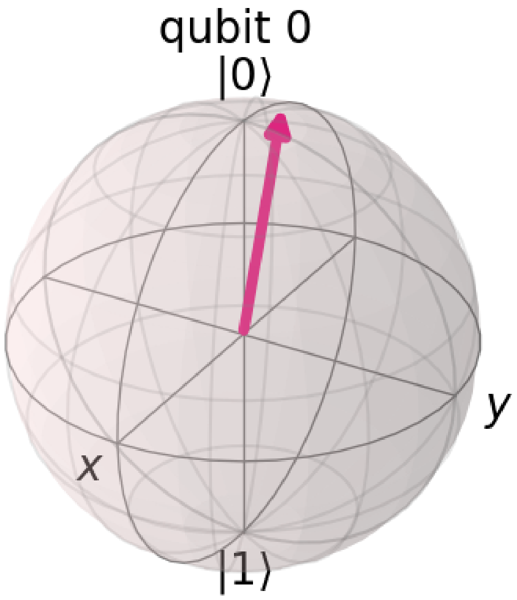


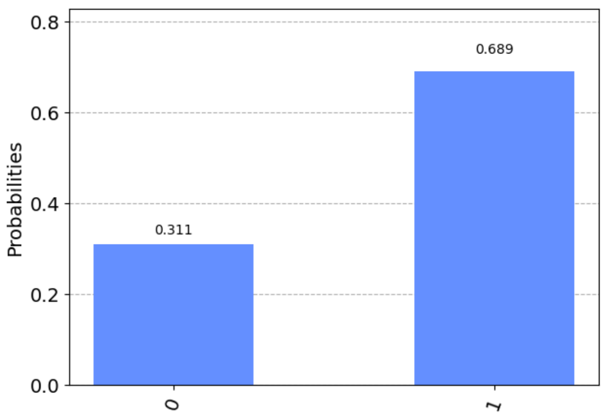
1.3 1-(g)



1.4 1-(h)

It is not the unique method, for it can be generated by several rotate gate to reach it.





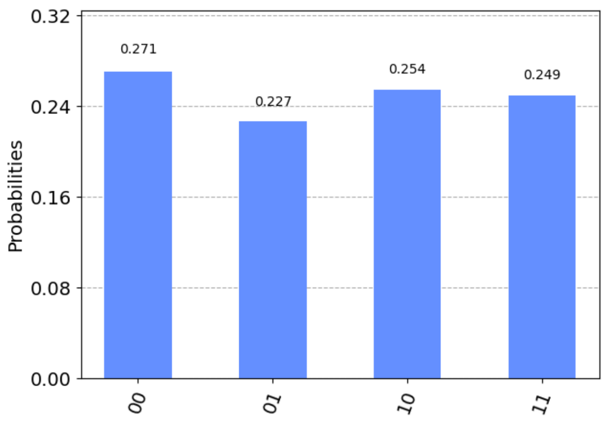
1.5 Bonus

$$\begin{bmatrix} cos(\frac{\theta}{2}) & -e^{i\lambda}sin(\frac{\theta}{2}) \\ e^{i\phi}sin(\frac{\theta}{2}) & e^{i(\lambda+\phi)}cos(\frac{\theta}{2}) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} cos(\frac{\theta}{2}) \\ e^{i\phi}sin(\frac{\theta}{2}) \end{bmatrix} = cos(\frac{\theta}{2})|0\rangle + e^{i\phi}sin(\frac{\theta}{2})|1\rangle \qquad [1.1]$$

Thus, by equation1.1, we can easily turn $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ to any point on the Bloch sphere, defined by the $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$

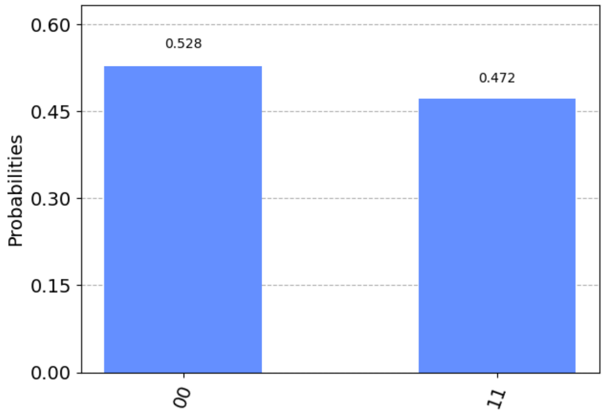
2 Problem2

2.1 2-(a)



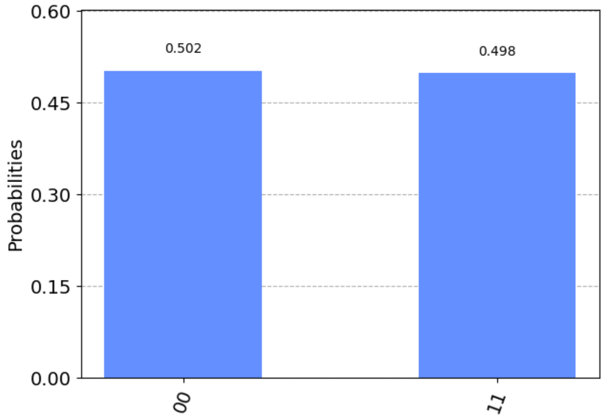
2.2 2-(b)

It should be 1, and by the graph below, the probability is 0.472.



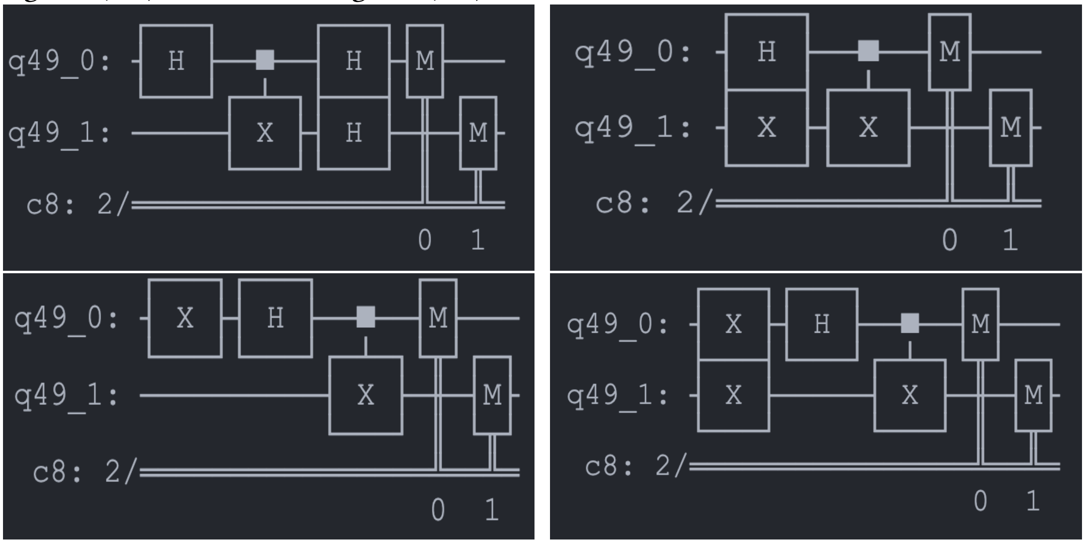
2.3 2-(c)

The result I get in this setting is in the following graph.



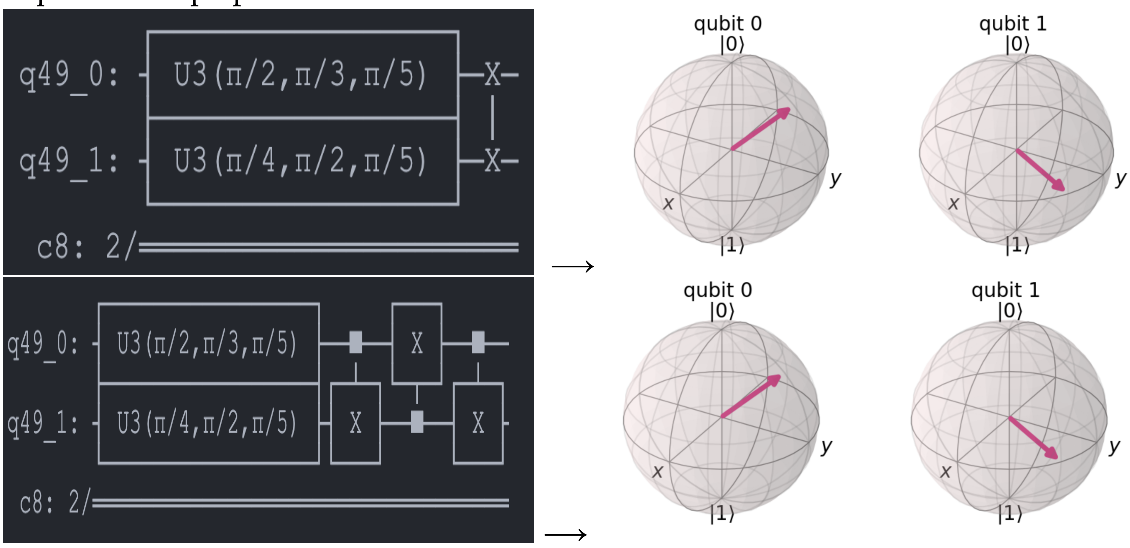
2.4 2-(d)

Start from the upper left, that circuit generates $|\Phi^+\rangle$, and the bottom left is $|\Phi^-\rangle$. The upper right is $|\Psi^+\rangle$, and bottom right is $|\Psi^-\rangle$.



2.5 2-(e)

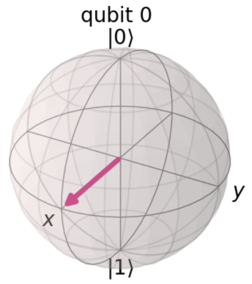
From the following graph, it can verify the Swap gate is equivalent to three C_x gates, for the 2-qubit state I prepared.

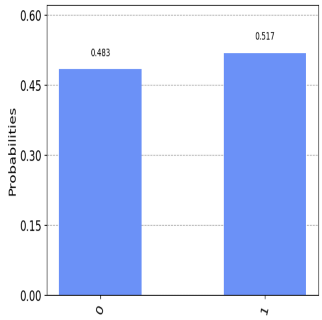


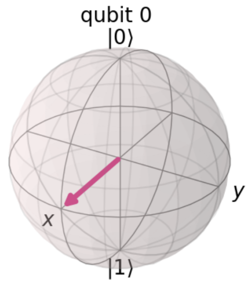
3 Problem3

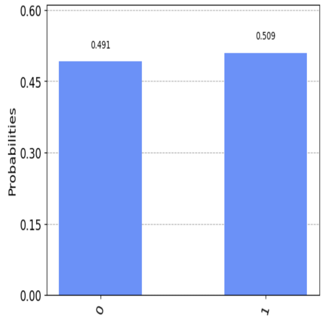
3.1 3-(a)

From the program below, we can verify the results of the two circuit is equivalent, and from the Bloch sphere plot and their measure results below, I cannot distinguish them from the other one.










```
circuit_1.h(q)
circuit_1.x(q)
circuit_1.draw()
```

```
q76: ─┤ H ├─┤ X ├─
c9: 1/════════════
```

```
circuit_2.h(q)
circuit_2.y(q)
circuit_2.z(q)
circuit_2.draw()
```

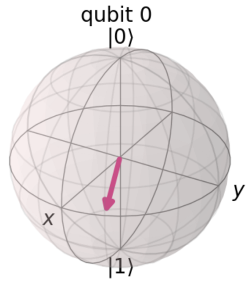
```
q76: ─┤ H ├─┤ Y ├─┤ Z ├─
c9: 1/══════════════════
```

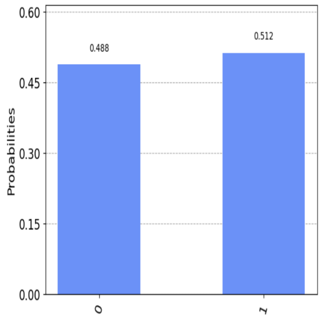
```
simulator = Aer.get_backend('statevector_simulator')
job = execute(circuit_1, simulator)
result = job.result()
statevector_1 = result.get_statevector()
simulator = Aer.get_backend('statevector_simulator')
job = execute(circuit_2, simulator)
result = job.result()
statevector_2 = result.get_statevector()
print(statevector_1.equiv(statevector_2))
```

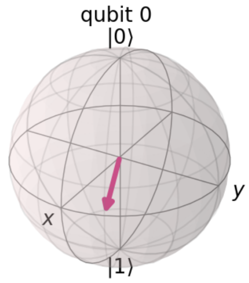
```
True
```

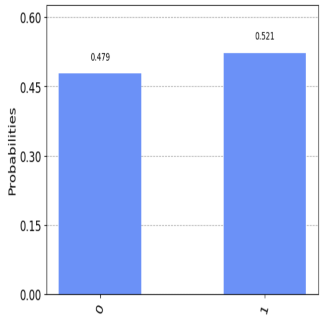
3.2 3-(b)

The two states I prepare $|\Psi_1\rangle$ and $|\Psi_2\rangle$ is $|\Psi_1\rangle = R_z(\frac{5\pi}{8})R_x(\frac{\pi}{2})\begin{bmatrix}1\\0\end{bmatrix}$, $|\Psi_2\rangle = R_z(\frac{\pi}{8})R_y(\frac{\pi}{2})\begin{bmatrix}1\\0\end{bmatrix}$, respectively, and they are equivalent, too. Also, I cannot distinguish them from the other one, too.



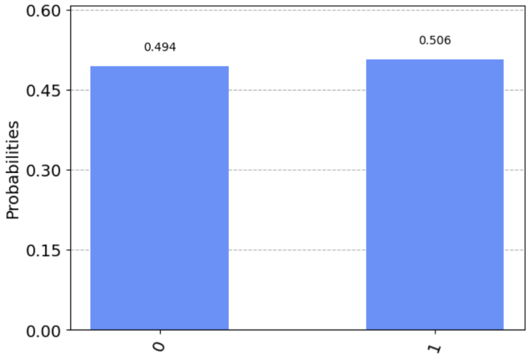


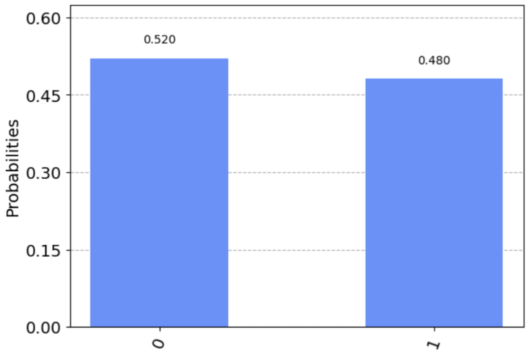




3.3 3-(c)

$|\Psi_1\rangle = \begin{bmatrix} 0.70710678 + 0.00000000e + 00j & -0.70710678 - 8.65956056e - 17j \end{bmatrix}$
$|\Psi_2\rangle = \begin{bmatrix} 0.70710678 + 0.j & 0.70710678 + 0.j \end{bmatrix}$
The measure result of $|\Psi_1\rangle$ is the left one, and the result of $|\Psi_2\rangle$ is the right one. And two state are not equivalent, for their imagine components are different.

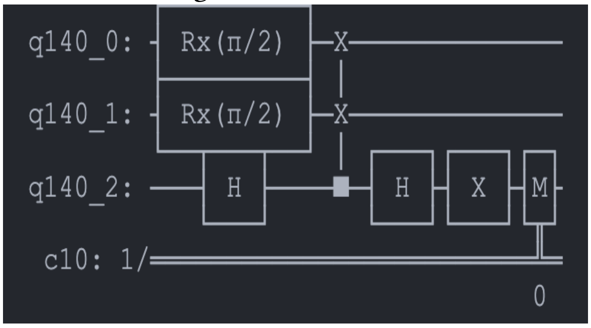


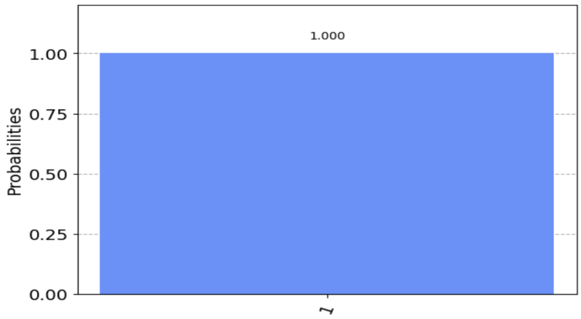


4 Problem4

4.1 The two states are equivalent.

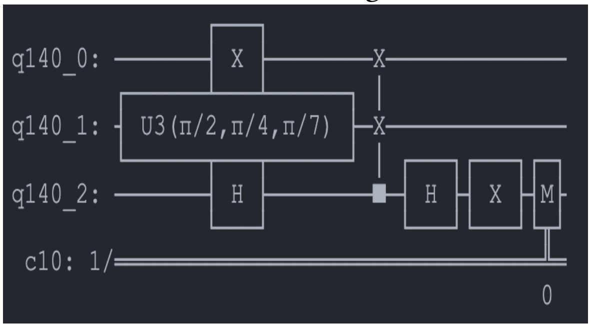
$|\Psi_1\rangle = |\Psi_2\rangle = R_x(\frac{\pi}{2})\begin{bmatrix}1\\0\end{bmatrix}$,the corresponding circuit is the left one, and the measure result of this circuit is the right one.

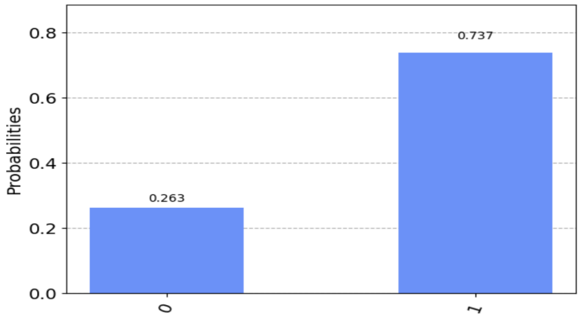




4.2 The two states are different.

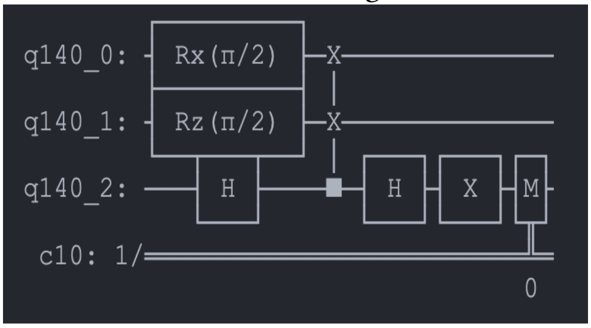
$|\Psi_1\rangle = X\begin{bmatrix}1\\0\end{bmatrix}$, $|\Psi_2\rangle = U_3(\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{7})\begin{bmatrix}1\\0\end{bmatrix}$, the corresponding circuit is the left one, and the measure result of this circuit is the right one.

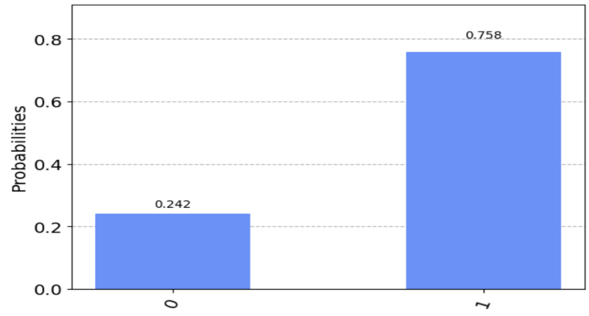




4.3 The two states are orthogonal.

$|\Psi_1\rangle = R_x(\frac{\pi}{2})\begin{bmatrix}1\\0\end{bmatrix}, |\Psi_2\rangle = R_z(\frac{\pi}{2})\begin{bmatrix}1\\0\end{bmatrix}$,, the corresponding circuit is the left one, and the measure result of this circuit is the right one.

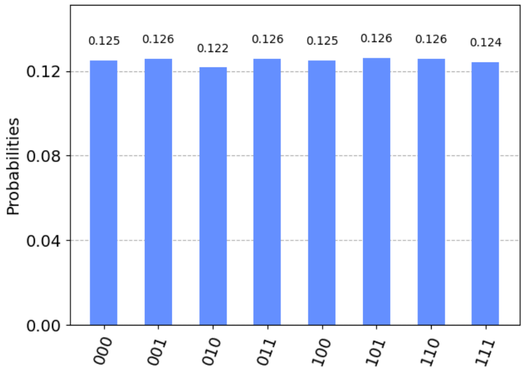




5 Problem5

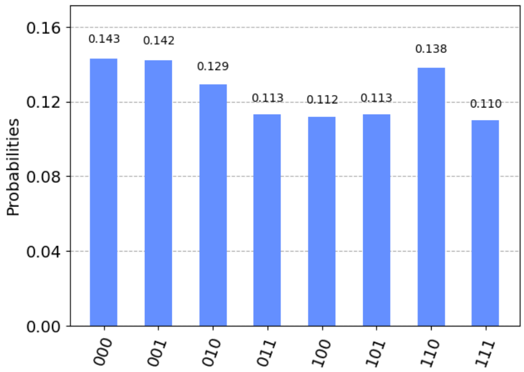
5.1 5-(a)

Sequence of number: {'110': 12886, '100': 12817, '101': 12920, '001': 12902, '010': 12463, '000': 12799, '111': 12725, '011': 12888}, and their probability distribution is in the following picture, as we can see in this picture, they have the similar probability(around 0.125), so it is random.



5.2 5-(b)

The result I get on IBMQ is in the following picture, and it is also random.



5.3 Bonus1

I use the program Question5-Bonus1 in my hw1.ipynb, and verify the result with ent[2], and the testing result is as the following picture. We can see the entropy of 3-bit generator is more than that of 2-bit generator, and they are really random.

```
~/Downloads ./ent -t ../Desktop/CSL/output_bonus_1_3bit.txt
0,File-bytes,Entropy,Chi-square,Mean,Monte-Carlo-Pi,Serial-Correlation
1,2048,2.498697,145467.000000,30.723633,4.000000,-0.993982
~/Downloads ./ent -t ../Desktop/CSL/output_bonus_1_2bit.txt
0,File-bytes,Entropy,Chi-square,Mean,Monte-Carlo-Pi,Serial-Correlation
1,2048,1.997822,161985.750000,29.775391,4.000000,-0.998465
```

5.4 Bonus2

From the picture in Bonus1, the entropy of 2-bit or 3-bit generator is 1.99 and 2.50, respectively. Thus, it can verify that qubit generator can generate more bits than bit generator.

References

[1] https://github.com/ozaner/qRNG/blob/master/qrng__init__.py

[2] https://www.fourmilab.ch/random/