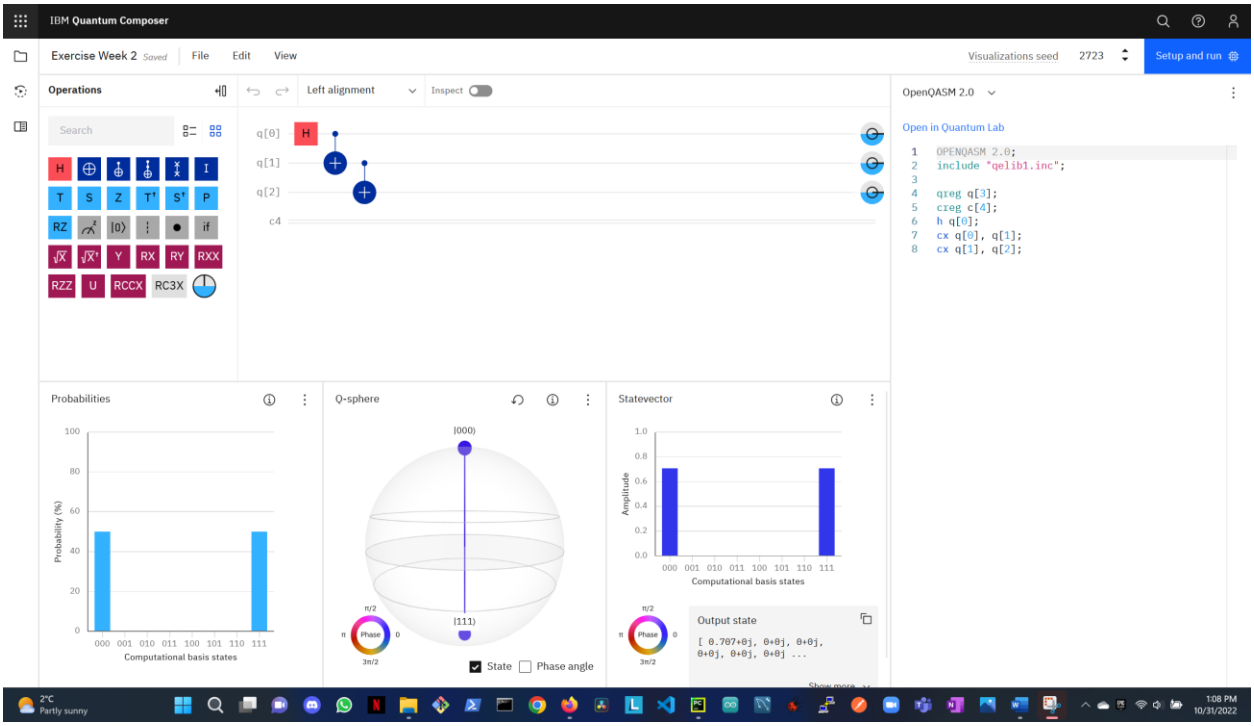
Haritha Weerathunga

Exercise 1. Entanglement

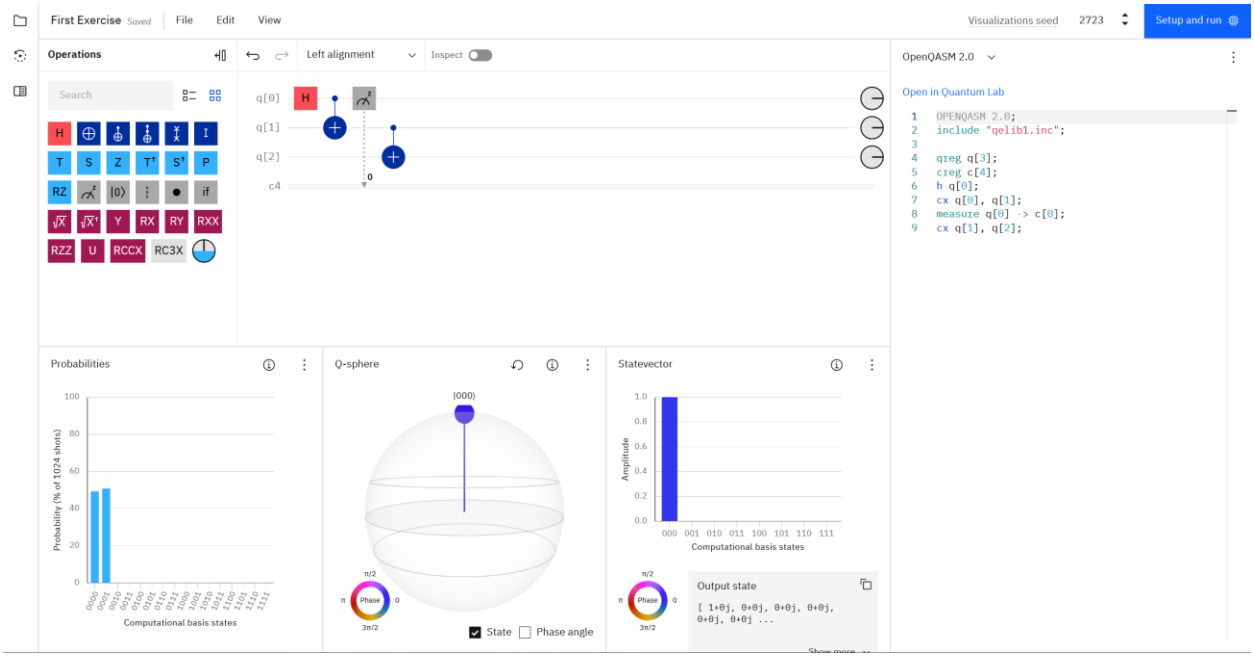


Code

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[3];
creg c[4];
h q[0];
cx q[0], q[1];
cx q[1], q[2];
```

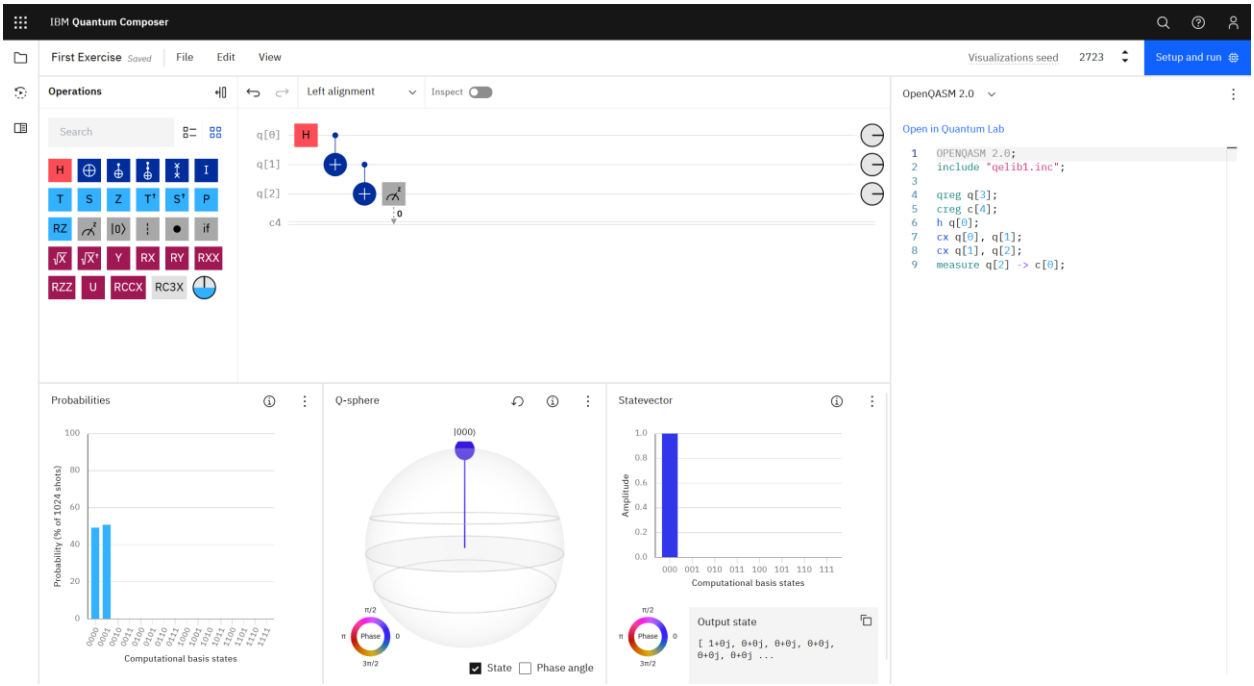
How does that change the states of the other qubits? Why does this happen? What happens if you measure some other qubit instead? Try also with different Visualization seeds (top right corner of the composer), for
example with values 600-605. What kind of effect do different seeds have on the resulting statevector?

After adding measurement

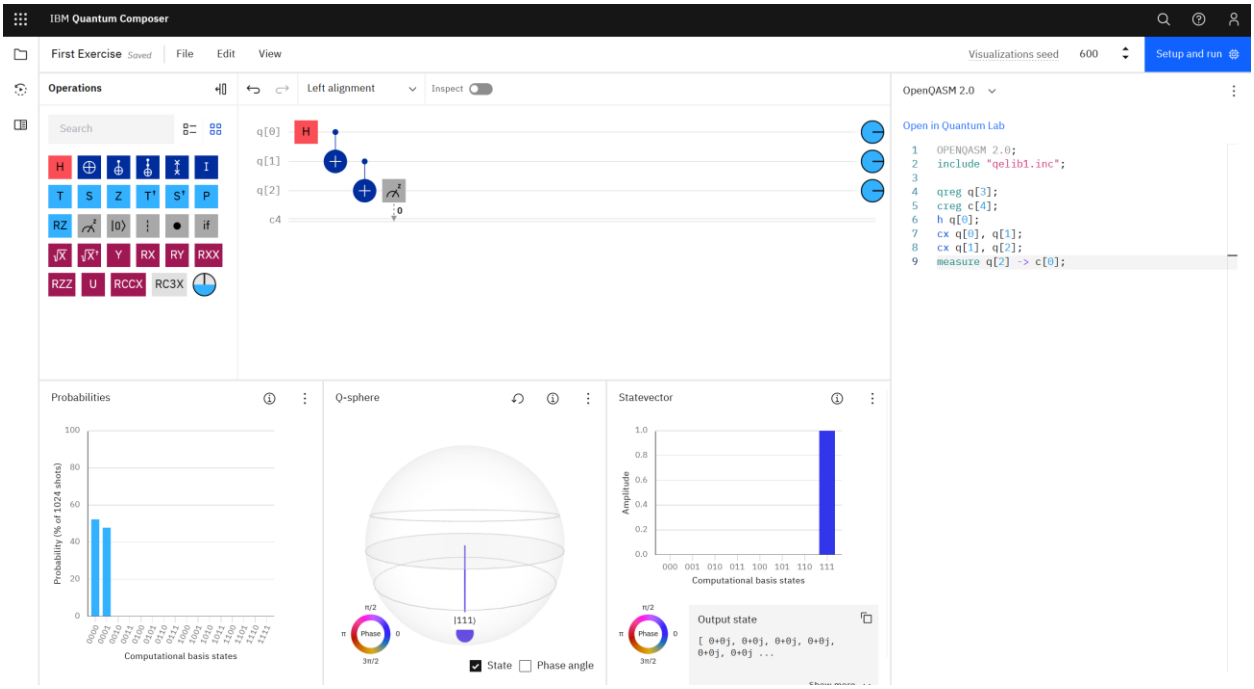


- This happens because when you always measure a qubit it always falls to one of the states zero or one. That is the nature of qubits. Quantum state is destroyed when you measure the qubit

Measuring another qubit



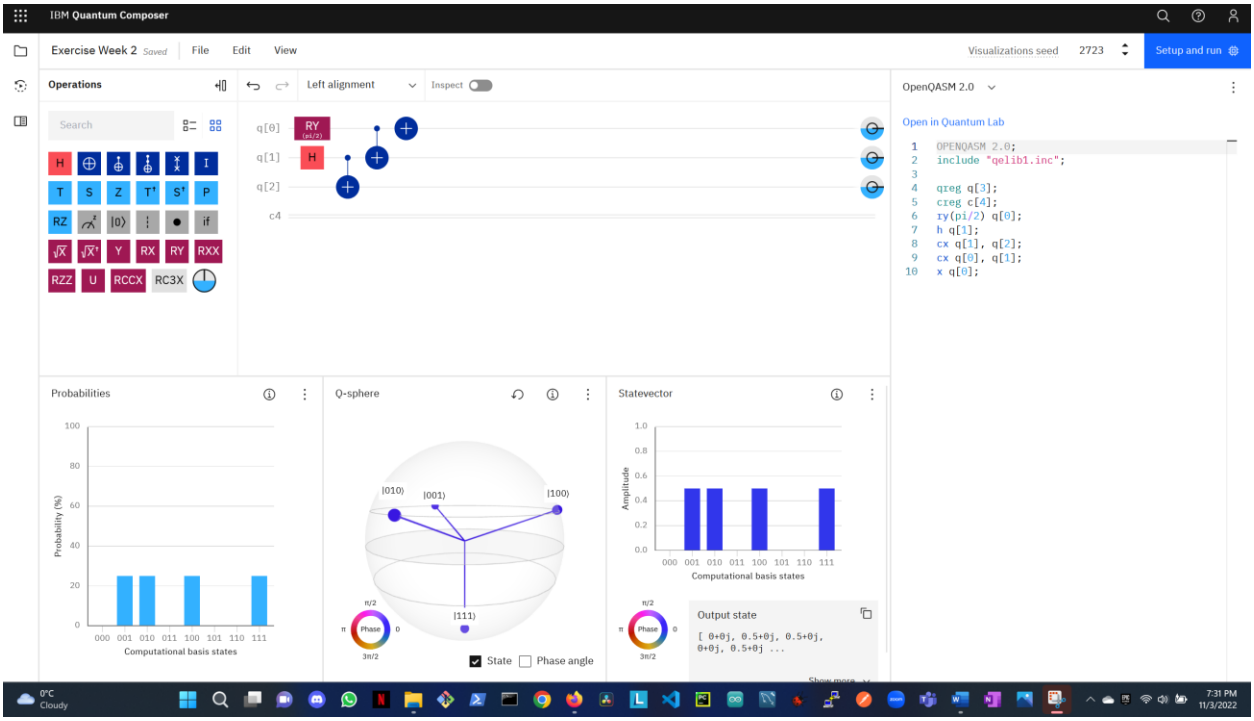
Changing the seed value



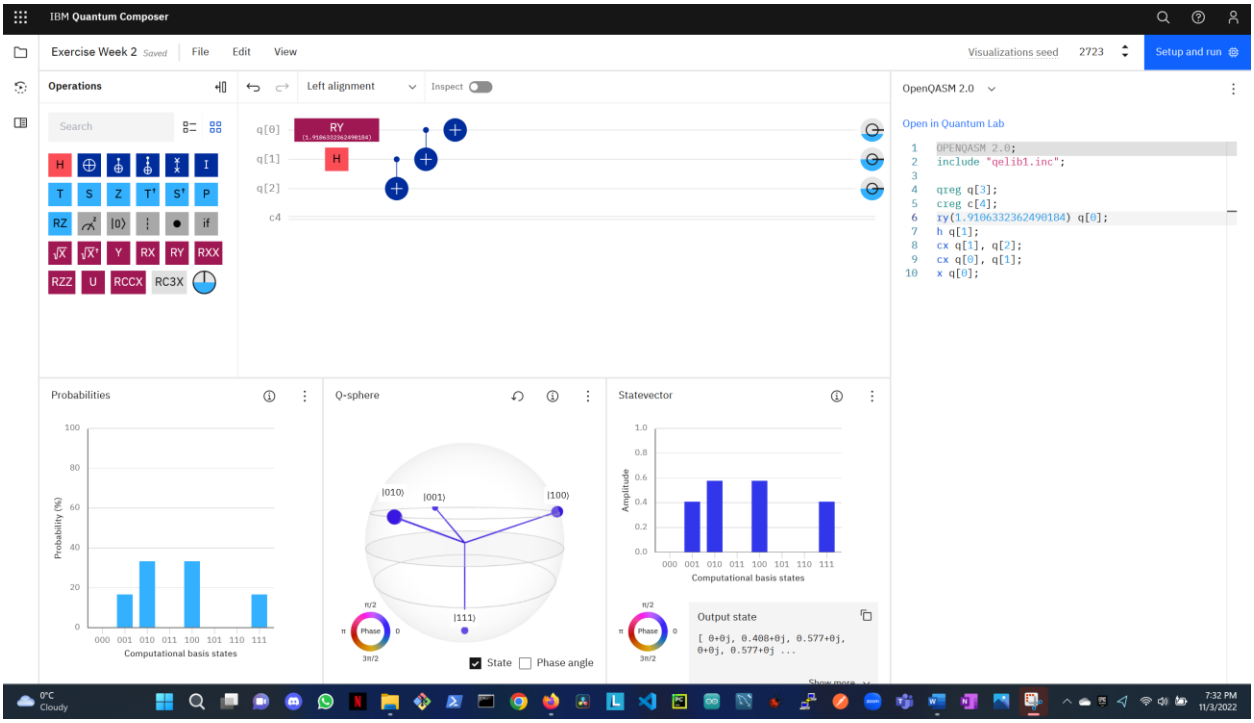
- Changing the seed value totally changes the phase of the q sphere and the state values of the state vector

Exercise 2. W-state

Before changing the pi value

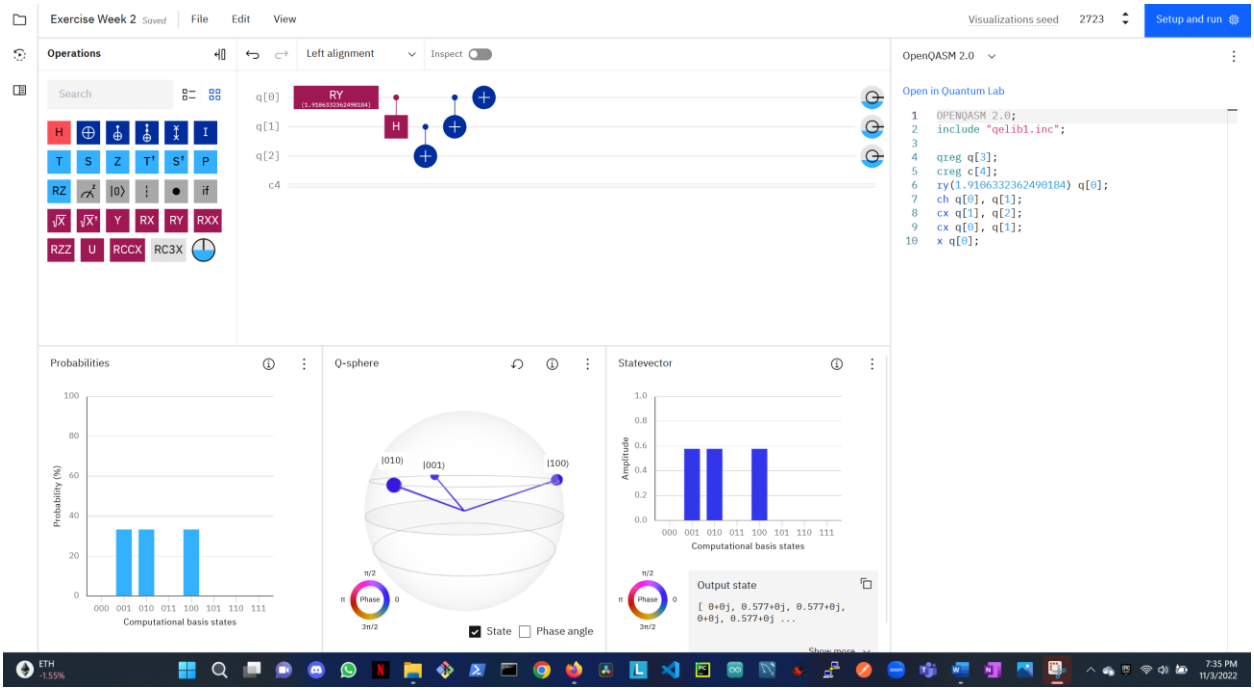


After changing pi value from code level



- How does it change the state vector ?
    - Here we can see that state vector shows the amplitude changes in 010 and 100 which is near to 0.6. 001 and 111 stays the same

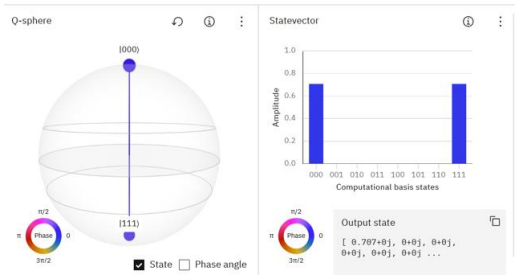
Adding the control gate modifier to the Hadamard gate



- QASM Code

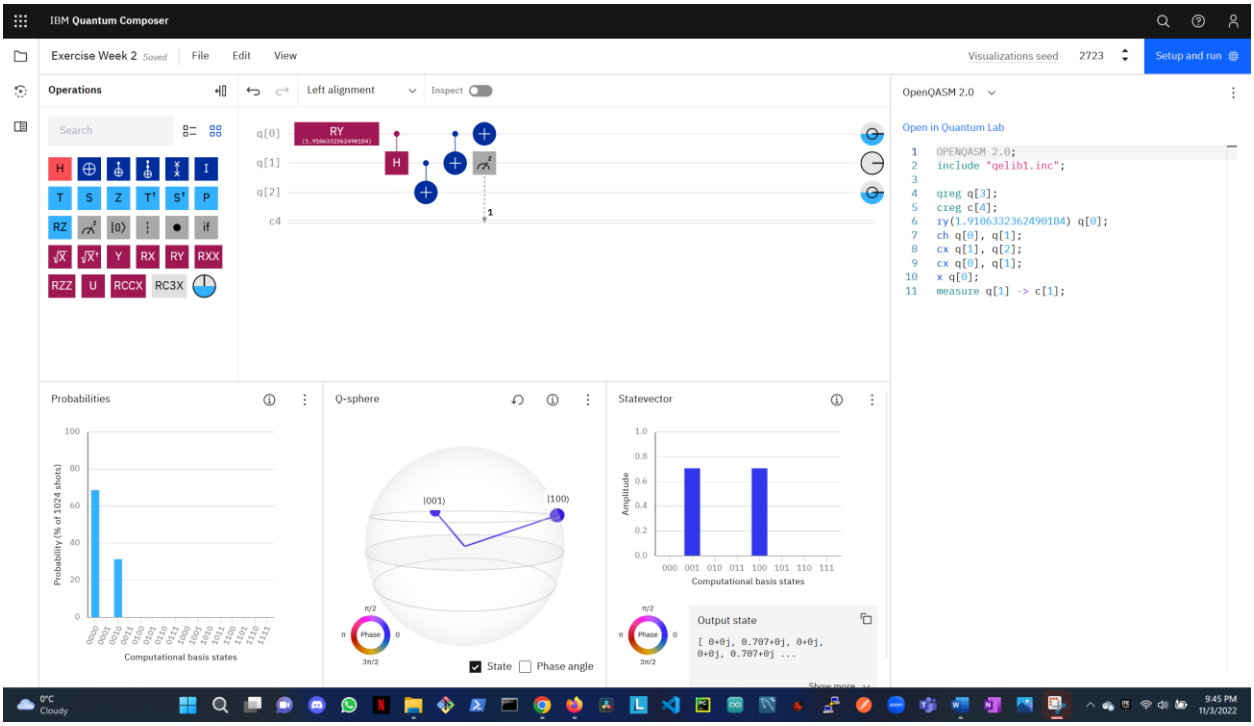
```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[3];
creg c[4];
ry(1.9106332362490184) q[0];
ch q[0], q[1];
cx q[1], q[2];
cx q[0], q[1];
x q[0];
```

  -
- How does the quantum states differ from exercise 1?
  - In the exercise one we see the following quantum state



    - Where we have 000 and 111 states
  - In the latest snapshot we see that we have a phase shifted qubits on 001,010,100

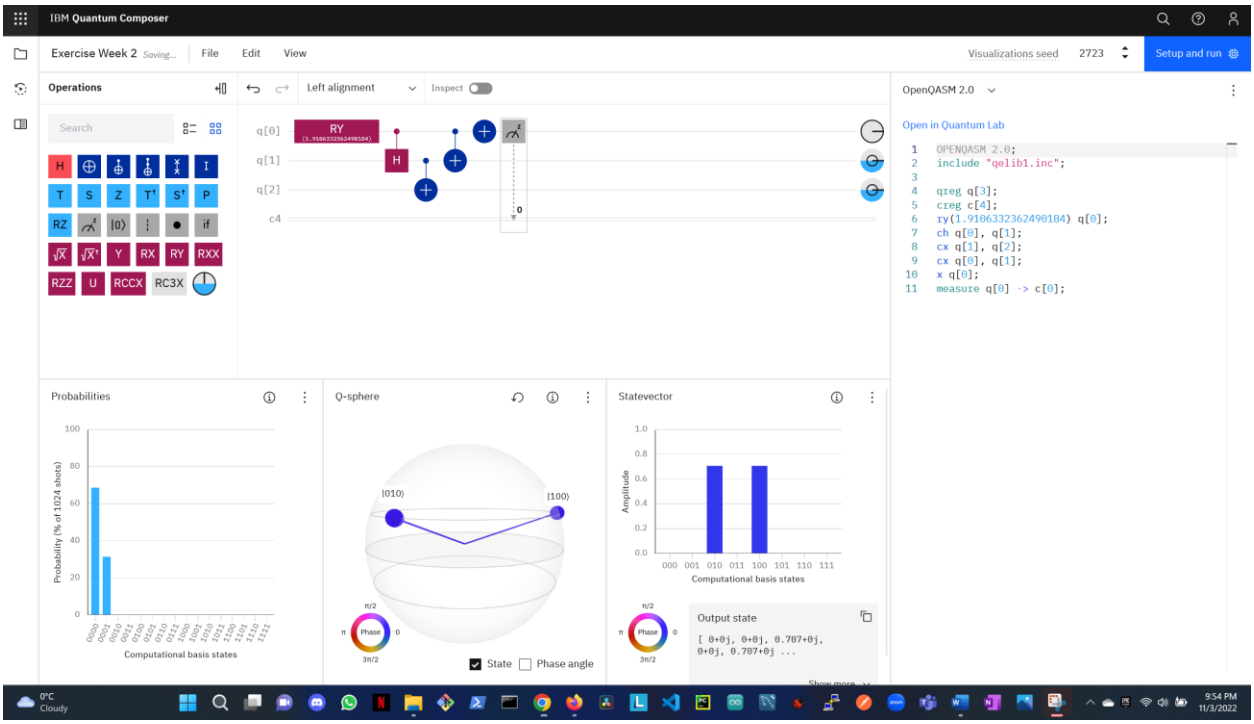
Add a measurement gate to a qubit



What kind of effect does it have on the results?

- This brings the measured qubit value to a constant state. In this case the quantum state of the q[1] is destroyed and the 010 state is no more

Delete the measurement and put a new one on another qubit. How does it change the results?



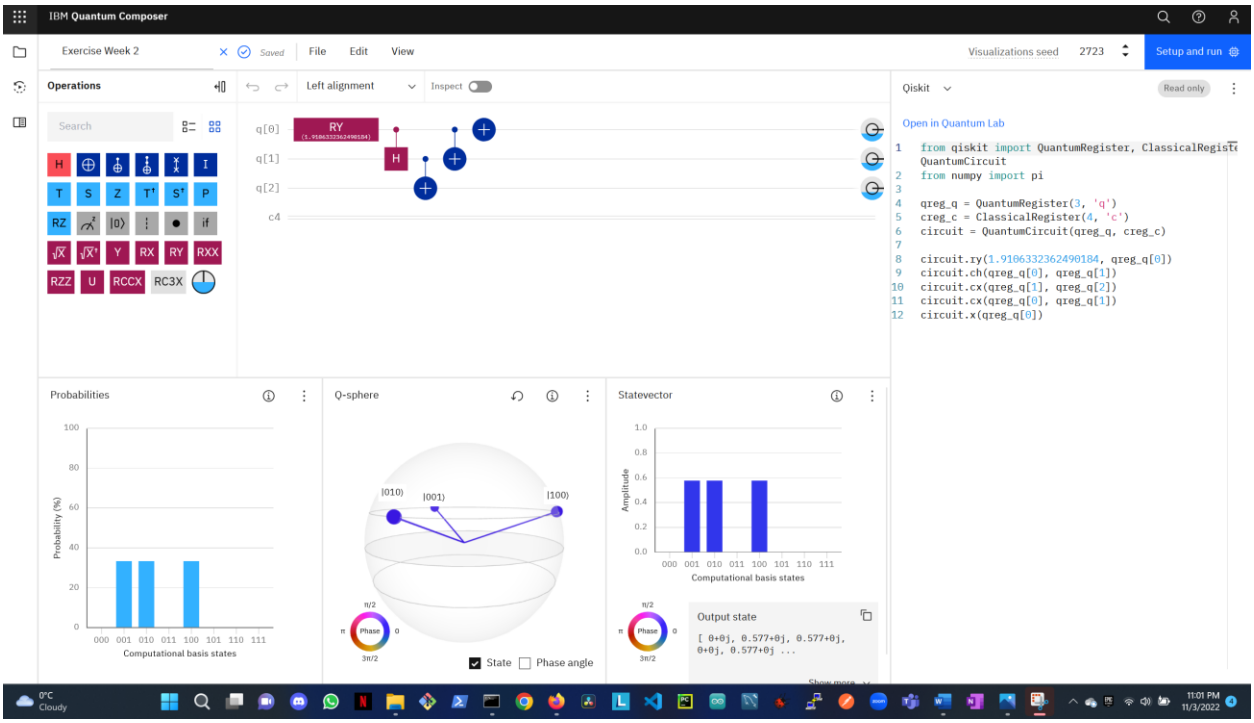
- The only change is the state change of a different qubit

How does this change from the situation in the exercise 1 when measuring a single qubit? Any idea why that happens?

- In the exercise 1 after the measurement though we add another one again did not change anything. But here it changed. The reason can be this W state is a entangled quantum state.

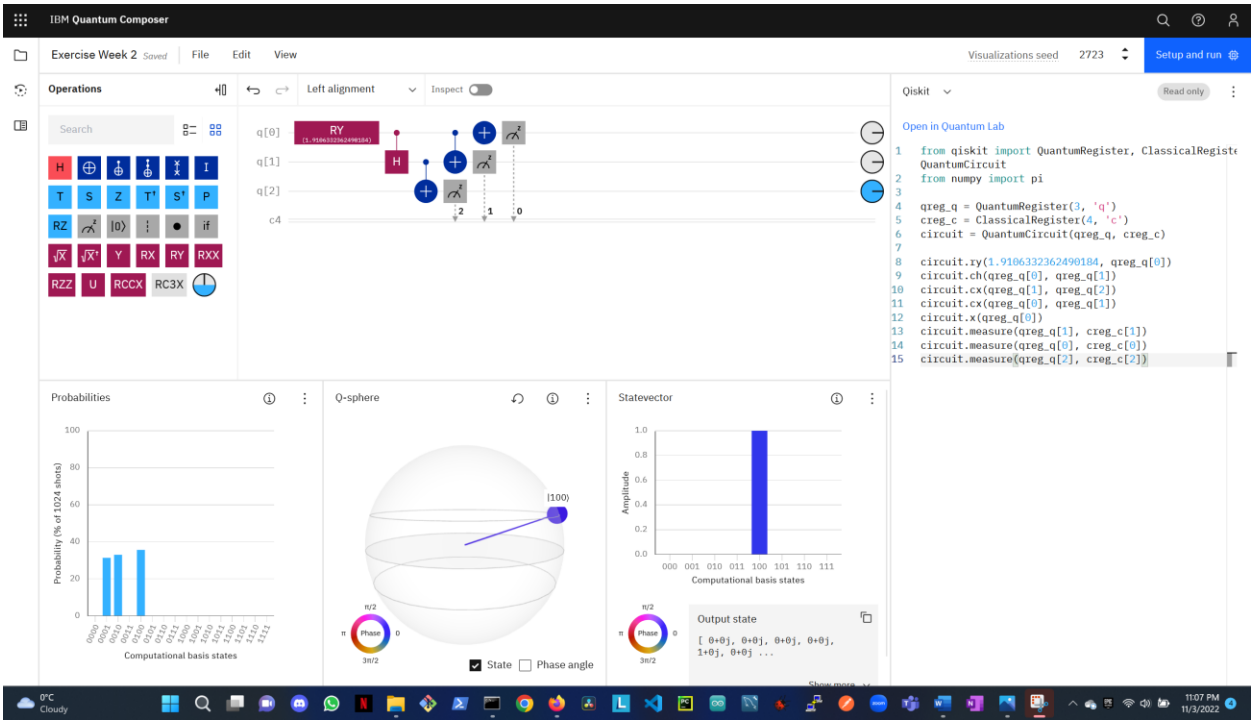
Qiskit

What do you think about the changes in the code? Which one do you prefer? You can change the view back and forth to compare



- This code seems wordy but it gives out more explanation. Personally I prefer python. So I think this is a nice touch

Before running the circuit make sure that you are measuring the state of each of the qubits

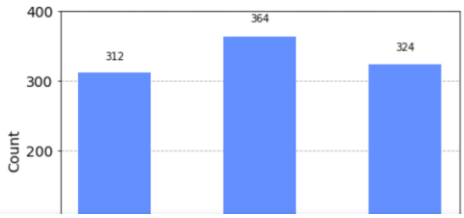


```
from ibm_quantum_widgets import CircuitComposer
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(4, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.ry(1.9106332362490184, qreg_q[0])
circuit.ch(qreg_q[0], qreg_q[1])
circuit.cx(qreg_q[1], qreg_q[2])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.x(qreg_q[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[2], creg_c[2])



from qiskit import Aer, transpile
backend = Aer.get_backend('statevector_simulator')
result = backend.run(transpile(circuit, backend), shots=1000).result()
from qiskit.visualization import plot_histogram
counts = result.get_counts()
plot_histogram(counts)
```

[1]:
```
from ibm_quantum_widgets import CircuitComposer
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(4, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.ry(1.9106332362490184, qreg_q[0])
circuit.ch(qreg_q[0], qreg_q[1])
circuit.cx(qreg_q[1], qreg_q[2])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.x(qreg_q[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[2], creg_c[2])


from qiskit import Aer, transpile
backend = Aer.get_backend('statevector_simulator')
result = backend.run(transpile(circuit, backend), shots=1000).result()
from qiskit.visualization import plot_histogram
counts = result.get_counts()
plot_histogram(counts)
```

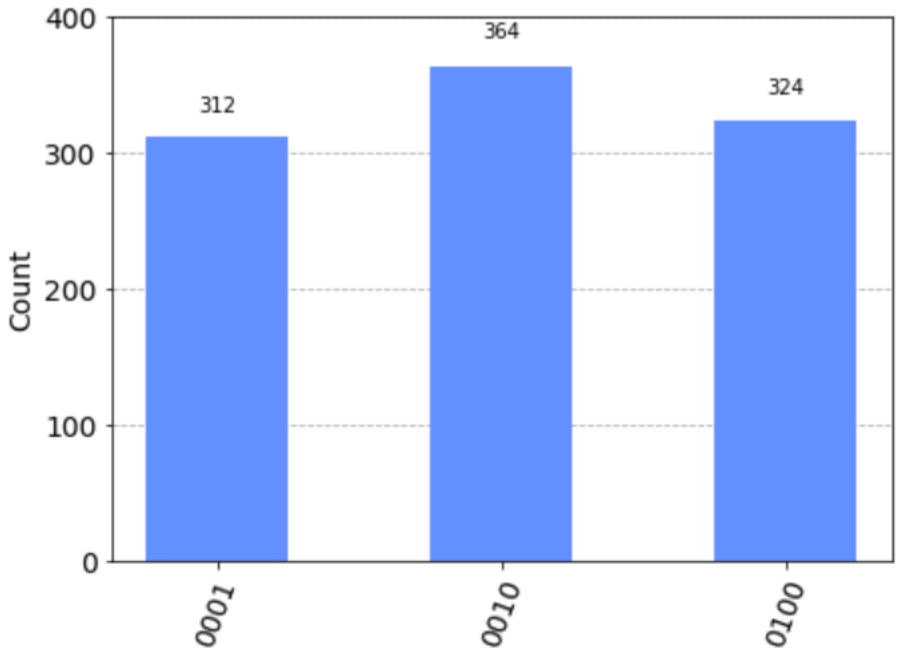
```
<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messagestream.MessageStream size changed, may indicate binary incompatibility. Expected 56 from C header, got 64 from PyObject
Matplotlib is building the font cache; this may take a moment.
```

[1]:



<frozen importlib._bootstrap>:219: RuntimeWarning: scipy._lib.messagestream.MC header, got 64 from PyObject
Matplotlib is building the font cache; this may take a moment.

[1]:



I think these results come because we are measuring the output.