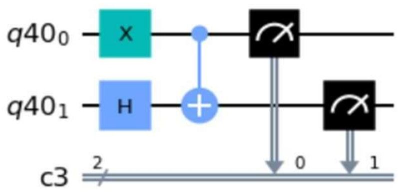
1) What is the output after executing the following circuit “1000” times?



a) {‘10’:490 , ‘01’:510}
b) {‘01’:490 , ‘10’:510}
c) {‘01’:490 , ‘11’:510}
d) {‘00’:490 , ‘11’:510}

2) Which of the statements given below will create a quantum circuit with ‘3’ quantum bits and ‘4’ classical bits?

a) qc= QuantumCircuit(4,3)
b) q= QuantumRegister(3)
   c= ClassicalRegister(4)
   qc= QuantumCircuit(q,c)
c) qc= QuantumCircuit(3,4)
d) qc= QuantumCircuit([3,4])

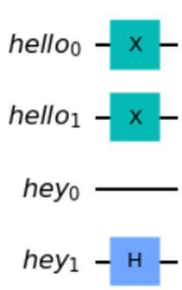
3) Given this code fragment, what is the probability that a measurement would result in |1> ?

```
qc = QuantumCircuit(1)
qc.x(0)
qc.ry(5*pi/8, 0)
```

a) 0.31
b) 0.25
c) 0.47
d) 0.69

4) Assuming the fragment below, which three code fragments would produce the circuit illustrated?

```
q = QuantumRegister(2,'hello')
c = QuantumRegister(2,'hey')
qc = QuantumCircuit(q,c)
```



a) qc.x([0,1])
   qc.h(3)
   qc.draw()

b) qc.h(q[0:2])
   qc.x(c[0])
   qc.draw()

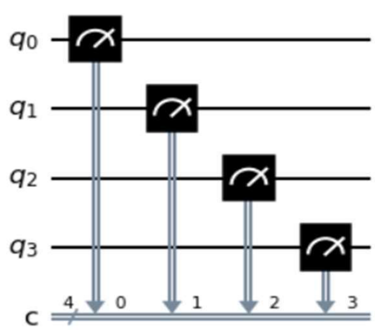
c) qc.h(q[0])
   qc.h(q[1])
   qc.x(c[0])
   qc.draw()

d) qc.h(q[-2:2])
   qc.x([3])
   qc.draw()

e) qc.h(q)
   qc.x(c)
   qc.draw()

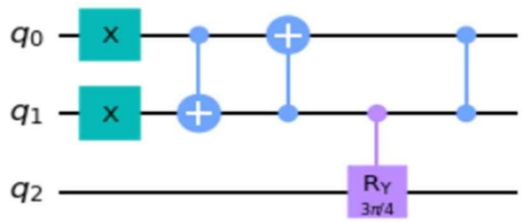
f) qc.x(q[0:2])
   qc.h(c[1:2])
   qc.draw('mpl')

5) Given an empty QuantumCircuit object, qc, with four qubits and three classical bits, which one of these code fragments would create this circuit?



a) qc.measure_all()
b) qc.measure(0,1,2,3,4)
c) qc.measure([0,0], [1,1], [2,2], [3,3],[4,4])
d) for i in range(4):
       qc.measure(i,i)
e) qc.measure([0,1,2,3], [0,1,2,3])

6) Select the code that perfectly matches the given figure below.



a)
```
qc = QuantumCircuit(3)
qc.x([0,1])
qc.cx(0,1)
qc.cx(1,0)
qc.ry(3*pi/4,1,2)
qc.swap(0,1)
```

b)
```
qc = QuantumCircuit(3)
qc.x(0,1)
qc.cx(0,1)
qc.cx(1,0)
qc.cry(3*pi/4,1,2)
qc.swap(0,1)
```

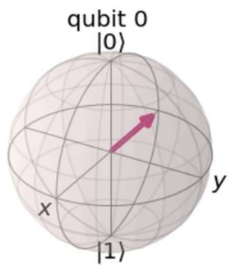
c)
```
qc = QuantumCircuit(3)
qc.x([0,1])
qc.cx(0,1)
qc.cx(1,0)
qc.cry(3*pi/4,1,2)
qc.cz(0,1)
```

d)
```
qc = QuantumCircuit(3)
qc.x([0,1])
qc.cx(0,1)
qc.cx(1,0)
qc.cry(3*pi/4,0,2)
qc.cz(0,1)
```

7) Given this code, which two inserted code fragments result in the state vector represented by this Bloch sphere?

```
qc= QuantumCircuit(1,1)
# Insert code fragment here
simulator = Aer.get_backend('statevector_simulator')
job = execute(qc, simulator)
result = job.result()
outputstate = result.get_statevector(qc)
```

plot_bloch_multivector(outputstate)



a) qc.ry(-pi/2,0)
b) qc.x(0)
   qc.h(0)
c) qc.h(0)
   qc.x(0)
d) qc.ry(pi/2,0)
e) qc.h(0)
   qc.rz(-pi/2,0)
f) qc.x(0)
   qc.rx(pi,0)
   qc.ry(-pi/2,0)

8) T-gate is a Qiskit phase gate with what value of the phase parameter?

a) п/2
b) п
c) п/4
d) п/8

9) Which two options would place a barrier across qubits 1 and 2 to the QuantumCircuit below?

```
q= QuantumRegister(3)
c= ClassicalRegister(3)
qc = QuantumCircuit(q,c)
```

a) qc.barrier_all()
b) qc.barrier([0,1])
c) qc.barrier()
d) qc.barrier(q[0:2])
e) qc.barrier(qc[0,1])

10) What code fragment codes the equivalent circuit if you remove the barrier in the following QuantumCircuit?

a) qc = QuantumCircuit(1,1)
   qc.h(0)
   qc.t(0)
   qc.h(0)
   qc.measure(0,0)
b) qc = QuantumCircuit(1,1)
   qc.h(0)
   qc.z(0)
   qc.h(0)
   qc.measure(0,0)
c) qc = QuantumCircuit(1,1)
   qc.h(0)
   qc.t(0)
   qc.tdg(0)
   qc.h(0)
   qc.measure(0,0)
d) qc = QuantumCircuit(1,1)
   qc.h(0)
   qc.sdg(0)
   qc.z(0)
   qc.x(0)

11) Given the following code, What is the depth of the circuit?

```
qc= QuantumCircuit(3)
qc.h(0)
qc.cx(0,1)
qc.h(0)
qc.h(1)
qc.cx(1,2)
```

a) 5
b) 3
c) 4
d) 6

12) Which one of the following codes can be used to add the two given circuits?

```
qc1= QuantumCircuit(1)
qc1.x(0)
```

```
qc1.h(0)

qc2= QuantumCircuit(2)
qc2.cx(0,1)
qc2.cx(1,0)
```

a) qc1+qc2
b) qc1.append(qc2)
c) qc_new = qc1.add(qc2)
d) None of the above

13) Which one of the following codes gives us the status of the job ?

a) job.monitor()
b) job.status()
c) job.status('check_my_job')
d) job.get_status()

14) Which one of the following codes is used to invert the given circuit?

```
qc = QuantumCircuit(2)
qc.x(1)
qc.z(0)
qc.h(0)
```

a) qc.get_inverse()
b) qc.inverse()
c) qc.invert()
d) We cannot invert circuits using qiskit

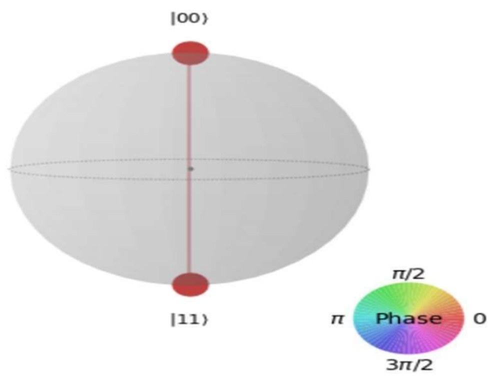
15) Which one of the following codes will result in an X gate?

a) HZH
b) ZYZ
c) HXH
d) ZXZ

16) Which one of the two following codes are used to check the version of qiskit?

a) qiskit._qiskit_version__
b) qiskit._version__
c) qiskit.__qiskit_version___
d) qiskit_version_table
e) qiskit_version_table_

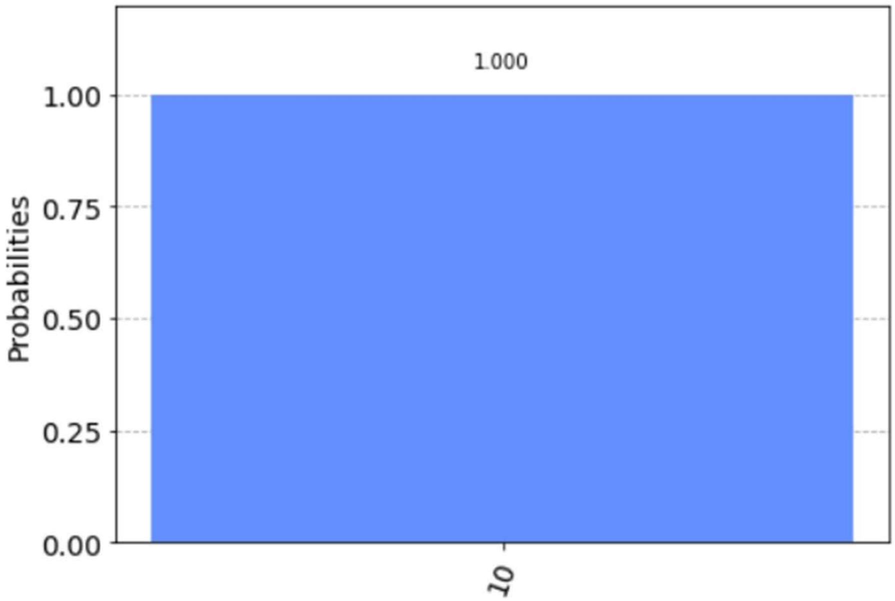
17) Given the qsphere output after executing the code, which one of the following code produces the given qsphere?



a) q = QuantumRegister(2)
   c = ClassicalRegister(2)
   qc = QuantumCircuit(q, c)
   qc.h(q[0])
   qc.cx(q[0], q[1])
b) q = QuantumRegister(2)
   c = ClassicalRegister(2)
   qc = QuantumCircuit(q, c)
   qc.x(q[0])
   qc.h([0])
c) q = QuantumRegister(2)
   c = ClassicalRegister(2)
   qc = QuantumCircuit(q, c)
   qc.h([0])
   qc.x([1])
d) q = QuantumRegister(2)
   c = ClassicalRegister(2)
   qc = QuantumCircuit(q, c)
   qc.h(q[1])
   qc.cx(q[0],q[1])

18) Select the most appropriate code, when executed gives the histogram data given below.

```
qc = QuantumCircuit(2,2)
qc.x(0)
qc.cx(0,1)
qc.cx(1,0)
qc.measure([0,1],[0,1])
```



a) backend = Aer.get_backend('statevector_simulator')
result = execute(qc, backend).result()
counts = result.get_counts()
plot_histogram(counts)

b) backend = Aer.get_backend('unitary_simulator')
result = execute(qc, backend).result()
counts = result.get_unitary()
plot_histogram(counts)

c) backend = Aer.get_backend('qasm_simulator')
result = execute(qc, backend).result()
counts = result.get_counts()
plot_histogram(counts)

d) backend = Aer.get_backend('pulser_simulator')
result = execute(qc, backend).result()
counts = result.get_statevector()
plot_histogram(counts)

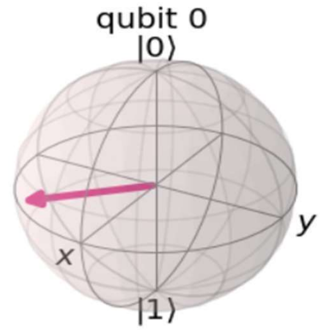
19) Which one of the following codes tells us the way qubits are arranged in the given quantum system?

provider = IBMQ.load_account()
backend = provider.get_backend('ibmq_belem')

a) plot_gate(backend)
b) plot_gate_map(backend)
c) plot_error_map(backend)
d) plot_gate_error(backend)

20) Arrange the given code fragments in particular order so that when the arranged code is executed it results in the given Bloch sphere.

```
qc = QuantumCircuit(1)
1) qc.x(0)
2) qc.sdg(0)
3) qc.h(0)
4) qc.z(0)
5) qc.t(0)
```



a) 1), 2), 5), 3), 4)
b) 1), 3), 2), 4), 5)
c) 3), 4), 2), 1), 5)
d) 4), 5), 1), 3), 2)
e) 1), 3), 5), 2), 4)
f) 4), 5), 2), 3), 1)

21) Which one of the following output results are correct when the given code is executed?

```
q = QuantumRegister(2)
c = ClassicalRegister(2)
qc = QuantumCircuit(q, c)
qc.h(q[0])
qc.x(q[1])
qc.h(q[1])

sim = Aer.get_backend('unitary_simulator')
job = execute(qc, sim)
unitary = job.result().get_unitary()
```

a)
```
[[ 0.70710678+0.j -0.70710678+0.j  0.        +0.j 0.        +0.j]
 [ 0.70710678+0.j 0.70710678+0.j  0.        +0.j 0.        +0.j]
 [ 0.        +0.j 0.        +0.j 0.70710678+0.j -0.70710678+0.j]
 [ 0.        +0.j 0.        +0.j  0.70710678+0.j  0.70710678+0.j]]
```

b)
```
[[ 0.70710678+0.j        0+0.j            0.        +0.j  0.        +0.j]
 [ 0.70710678+0.j  0.70710678+0.j  0.        +0.j  0.        +0.j]
 [ 0.        +0.j  0.        +0.j              0+0.j  -0.70710678+0.j]
 [ 0.        +0.j  0.        +0.j  0.70710678+0.j  0.70710678+0.j]]
```

c)
```
[[ 0+0.j -0.70710678+0.j                  0.        +0.j 0.        +0.j]
 [                0+0.j  0+0.j                  0.        +0.j 0.        +0.j]
 [ 0.        +0.j   0.        +0.j  0.70710678+0.j -0.70710678+0.j]
 [ 0.        +0.j  0.        +0.j  0.70710678+0.j  0.70710678+0.j]]
```

d)
```
[[ 0+0.j            0            +0.j        0.        +0.j 0.        +0.j]
 [0+0.j             0             +0.j       0.        +0.j 0.        +0.j]
 [ 0+0.j                  0.+0.j 0.70710678+0.j -0.70710678+0.j]
 [ 0.        +0.j  0.        +0.j  0.70710678+0.j                  0+0.j]]
```

e) None of these

22) Which one of the following output results are correct when the given code is executed?

```
qc = QuantumCircuit(2,2)
qc.h(0)
qc.z(1)
backend = Aer.get_backend('statevector_simulator')
job = execute(qc, backend)
statevector = job.result().get_statevector()
print(statevector)
```

a) [0.70710678+0.j 0+0.j -0.70710678 +0.j -0. +0.j]
b) [0.70710678+0.j 0+0.j -0. +0.j -0. +0.j]
c) [0.70710678+0.j 0.70710678+0.j -0. +0.j -0. +0.j]
d) [0+0.j 0.70710678+0.j -0. +0.j -0. +0.j]

23) Convert the given QASM circuit to the QISKIT circuit.

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q0[1];
creg c0[1];
h q0[0];
x q0[1];
measure q0[1] -> c0[1];
```

a)
```
q = QuantumRegister(1)
c = ClassicalRegister(1)
qc = QuantumCircuit(q, c)
qc.h(q[0])
qc.x(q[1])
```

b)
```
q = QuantumRegister(1)
c = ClassicalRegister(1)
qc = QuantumCircuit(q, c)
qc.h(q[1])
qc.x(q[0])
qc.measure([0,1],[0,1])
```

c)
```
q = QuantumRegister(1)
c = ClassicalRegister(1)
qc = QuantumCircuit(q, c)
qc.h(q[1])
qc.x(q[0])
qc.measure([1],[1])
```

d)
```
q = QuantumRegister(1)
c = ClassicalRegister(1)
```

```
qc = QuantumCircuit(q, c)
qc.h(q[0])
qc.x(q[1])
qc.measure([1],[1])
```

24) Which of the following given results matches the maximally entangled state or bell state?

a) {'10': 520, '01': 504}
b) {'10': 494, '11': 530}
c) {'00': 513, '11': 487}
d) {'01': 524, '10': 476}

25) Which of these code fragments would execute a circuit ?

```
from qiskit import QuantumCircuit, execute, BasicAer
backend = BasicAer.get_backend('qasm_simulator')
qc = QuantumCircuit(2)
# insert code here
```

a) execute(qc, backend, shots =1024, basic_gates=['u1', 'u2', 'u3', 'cx'], max_credits='100')
b) execute(qc, memory_slots_shape='square', backend)
c) execute(qc, backend, mode='custom')
d) execute(qc, backend, init_qubits= [0,1])

26) Which code snippet would execute a circuit given these parameters?

- Use the Unitary Simulator
- Measure the circuit 1000 times
- Use a coupling map that connects three qubits linearly
- Convert the output to latex

```
qc = QuantumCircuit(3)
# Insert code fragment here
result = job.result()
```

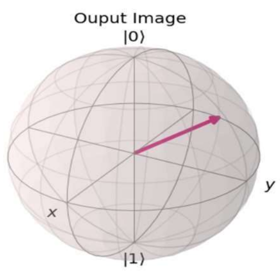
a)
```
backend = Aer.get_backend('unitary_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend, shot=1024, coupling_map=couple_map)
unitary = result.get_unitary()
unitary = array_to_latex(unitary)
```
b)
```
backend = Aer.get_backend('unitary_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend, loop=1000, coupling_map=couple_map)
unitary = result.get_unitary()
unitary = array_latex(unitary)
```

c)
```
backend = Aer.get_backend('unitary_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend, shots=1000, coupling_map=couple_map)
```

```
unitary = result.get_unitary()
unitary = array_to_latex(unitary)
```

d) backend = Aer.get_backend('ibmq_simulator')
```
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend, shot=1024, coupling_map=couple_map)
unitary = result.get_unitary()
unitary = array_latex(unitary)
```

27) Which one of the following codes given below will result in the image shown?



a) cords=[1,1,1]
```
plot_bloch_multivector(cords,'Output Image')
```
b) cords=[0,pi/7,0]
```
plot_bloch_vector(cords,cords_type='cartesian',name='Output  Image')
```
c) cords=[pi/2,pi/2,pi/2]
```
plot_bloch_vector(cords,cords_type='spherical', title='Output Image')
```
d) qc = QuantumCircuit(1)
```
qc.tdg(0)
qc.sdg(0)
```

28) Which simulators are available in Aer?

a) unitary_simulator
b) quantum_simulator
c) qasm_simulator
d) quantum_circuit_simulator
e) statevector_simulator
f) basic_qasm_simulator
g) ibmq_simulator
h) pulse_simulator

29) Which code fragment would yield an operator that represents a single-qubit Z gate?

a) op = Operator.Zop(0)
b) op = Operator([[0,1]])
c) qc = QuantumCircuit(1)
```
qc.z(0)
op = Operator(qc)
```
d) op = Operator([[1,0,0,1]])

30) Which one of the following codes will calculate the process fidelity of the given code below?

```
op_a = Operator(XGate())
op_b = np.exp(1j * 0.5) * Operator(XGate())
```

a) process_fidelity(op_a, op_b)
b) fidelity.get_process(op_a,op_b)
c) process_fidelity_cal(np.array(op_a),np.array(op_b))
d) fidelity_process(np.array(op_a),np.array(op_b))

31) Which one of the following codes are used to calculate the tensor product of A and B?

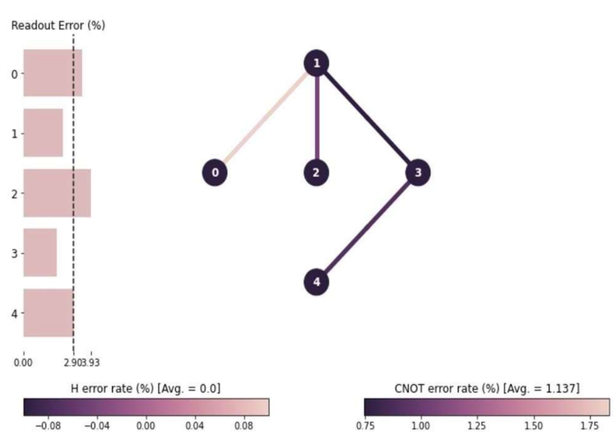
a) A = Operator(Pauli(label='X'))
B = Operator(Pauli(label='Z'))
A.tensor(B)
b) A = Operator(Pauli(label='X'))
B = Operator(Pauli(label='Z'))
A.expand(B)
c) A = Operator(Pauli(label='X'))
B = Operator(Pauli(label='Z'))
A.compose(B)
d) A = Operator(Pauli(label='X'))
B = Operator(Pauli(label='Z'))
B.tensor(A)

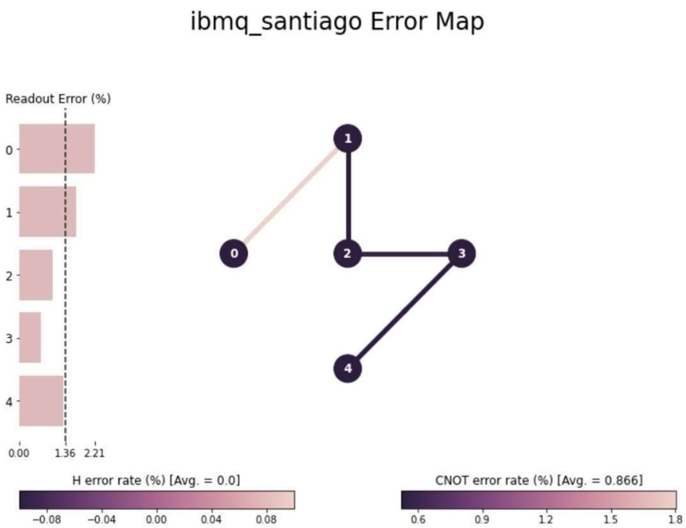
32) Which of the given codes below are true or result true when executed.

a) Operator(Pauli(label='X')) == Operator(XGate())
b) Operator(XGate()) == np.exp(1j * 0.5) * Operator(XGate())
c) Operator(Pauli(name='X')) == Operator(XGate())
d) Operator(XGate()) == np.exp(1j * 0) * Operator(XGate())

33) Which of the given options are correct?

ibmq_belem Error Map





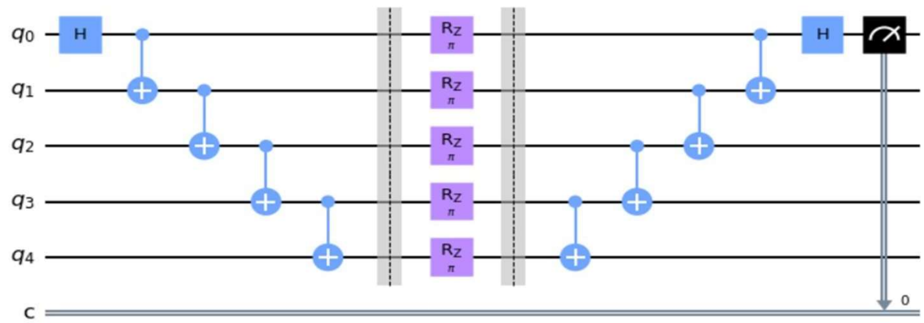
a) H error rate of ibmq_belam is equal to the H error rate of ibmq_santiago
b) CNOT error rate of ibmq_santiago is greater than the CNOT error rate of ibmq_belam
c) CNOT error rate of ibmq_santiago is less than the CNOT error rate of ibmq_belam
d) Readout error rate of 3rd qubit in ibmq_belam is greater than the readout error of 3rd qubit in ibmq_santiago
e) Readout error rate of 3rd qubit in ibmq_belam is less than the readout error of 3rd qubit in ibmq_santiago
f) The average readout error of ibmq_santiago is greater than the average readout error of ibmq_belem
g) The average readout error of ibmq_santiago is less than the average readout error of ibmq_belem

34) Which of the following codes, when executed gives the density matrix?

```
q  = QuantumRegister(1)
c = ClassicalRegister(1)
qc = QuantumCircuit(q, c)
qc.h(q[0])
# Insert your code here
```

a) DensityMatrix = Density_Matrix.get_instruction(qc)
b) Density_Matrix = DensityMatrix.get_instructions(qc)
c) Density_Matrix = DensityMatrix.from_instruction(qc)
d) DensityMatrix = DensityMatrix.from_instructions(qc)

35) Arrange the given code fragments in particular order so that when the arranged code is executed it results in the given figure below.



1) qc = QuantumCircuit(5, 1)
2) qc.barrier()
3) qc.h(0)
4) n = 5
5) for i in range(n-1):
6) for i in reversed(range(n-1)):
7) qc.cx(i, i+1)
8) qc.rz(pi, range(5))
9) qc.measure(0, 0)

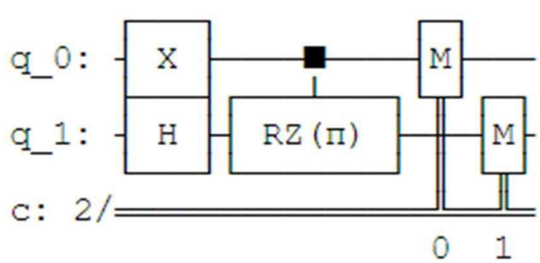
a) 1),7),8),5),2),7),4),3),2),3),6),9)
b) 4),1),3),5),7),2),8),2),6),7),3),9)
c) 3),6),7),2),5),7),8),2),3),1),4),9)
d) 4),1),3),6),7),3),8),2),5),7),2),9)

36) Which line of code would assign a unitary simulator object to the variable backend?

a) backend = BasicAer.UnitarySimulatorPy()
b) backend = BasicAer.get_backend('unitary_simulator')
c) backend = BasicAer.get_back('unitary_simulator')
d) backend = BasicAer.StatevectorSimulatorPy().get_backend()

37) When executed, which one of the following codes produces the given image?

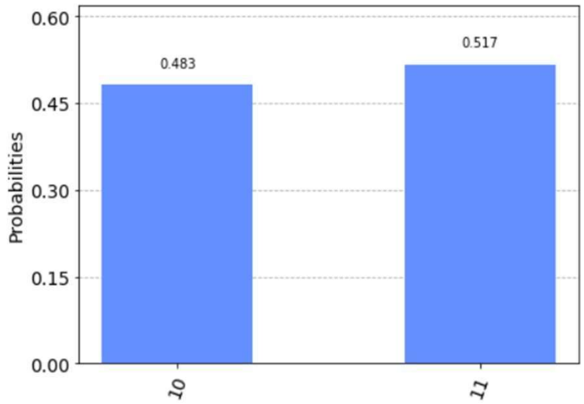
```
qc = QuantumCircuit(2,2)
qc.x(0)
qc.h(1)
qc.crz(pi,0,1)
qc.measure([0,1],[0,1])
#Insert your code here
```



a) qc.draw('mpl')

b) qc.draw('text')
c) qc.draw('latex')
d) qc.plot('text')

38) When executed, which one of the following codes produces the histogram image given below?




```
qc = QuantumCircuit(2,2)
#Insert your code here
qc.measure([0,1],[0,1])
#Insert your code here
counts = result.get_counts()
plot_histogram(counts)
```

a) qc.x(1)
qc.cx(1,0)
qc.h(1)
backend = Aer.get_backend('qasm_simulator')
result = execute(qc,backend, shots=1024).result()

b) qc.x(0)
qc.cx(1,0)
qc.h(1)
backend = Aer.get_backend('statevector_simulator')
result = execute(qc,backend, shots=1000).result()

c) qc.x(0)
qc.cx(0,1)
qc.h(1)
backend = Aer.get_backend('qasm_simulator')
result = execute(qc,backend, shots=1024).result()

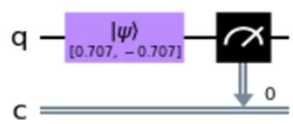
d) qc.x(1)
qc.cx(0,1)
qc.h(0)
backend = Aer.get_backend('qasm_simulator')
result = execute(qc,backend, shots=1000).result()

39) Which of the following options belongs to aer backend?

a) AerSimulator('aer_simulator_statevector')
b) UnitarySimulator('get_unitarysimulator')
c) AerSimulator('aer_simulator_superop')
d) AerSimulator('aer_simulator_density_matrix')
e) AerSimulator('aer.get_simulator_statevector')
f) AerSimulator('aer_simulator_matrix')

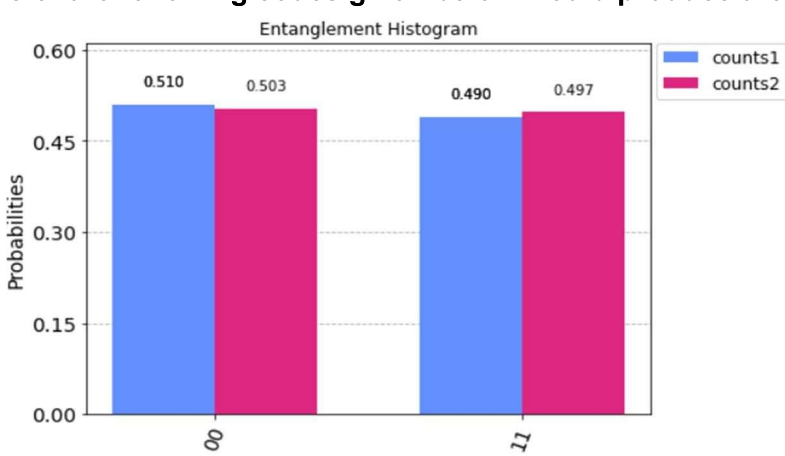
40) Which one of the following codes would initialize the given quantum circuit?

```
initial_state = [1/sqrt(2), -1/sqrt(2)]
arb_st = QuantumCircuit(1,1)
#Insert your code here
arb_st.measure(0,0)
arb_st.draw('mpl')
```



a) arb_st.initialize(initial_state, 0)
b) arb_st.get_initialize(inital_state,1)
c) arb_st.initialize(get_initial_state, 0)
d) arb_st.initialize(initial_state, 1)

41) Which one of the following codes given below would produce the given image?



a) plot_histogram([counts1, counts2], name="Entanglement Histogram")
b) plot_histogram([counts1, counts2],legend=legend, name="Entanglement Histogram")
i) plot_histogram([counts1, counts2],legend=legend, title="Entanglement Histogram")
c) plot_histogram([counts1, counts2],legend=True, name="Entanglement Histogram")

42) Which one of the following codes will create a random circuit?

```
from qiskit.circuit.random import random_circuit
#Insert your code here
```

a) circ = random_circuit(2, 2, reset=reset, measure=True)

b) circ = random_circuit(2, 2,conditional=True, measurement=measure)
c) circ = random_circuit(2, 2, measure=False)
d) circ = random_circuit(2, 2, max_operands=True)

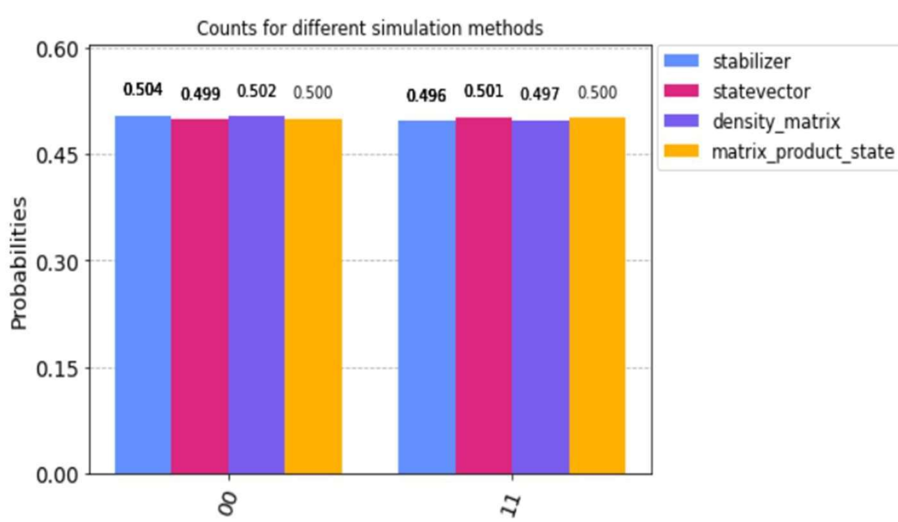
43) Which of the following codes given below will take the bloch vector to |1> state?

a) qc.x(0)
   qc.ry(pi,0)
b) qc.h(0)
   qc.rz(pi,0)
   qc.ry(pi/2,0)
c) qc.h(0)
   qc.rx(pi,0)
   qc.ry(-pi/2,0)
d) qc.x(0)
   qc.h(0)
   qc.ry(-pi/2,0)

44) Choose the correct backend from the given options.

a) ‘qasm_simulator’
b) ‘qasm.simulator’
c) ‘qasmsimulator
d) ‘QasmSimulator’

45) According to the image given below, choose the backend that has produced the most stable bell state?



a) aer_simulator_stabilizer
b) aer_simulator_density_matrix
c) aer_simulator_matrix_product_state
d) aer_simulator_statevector

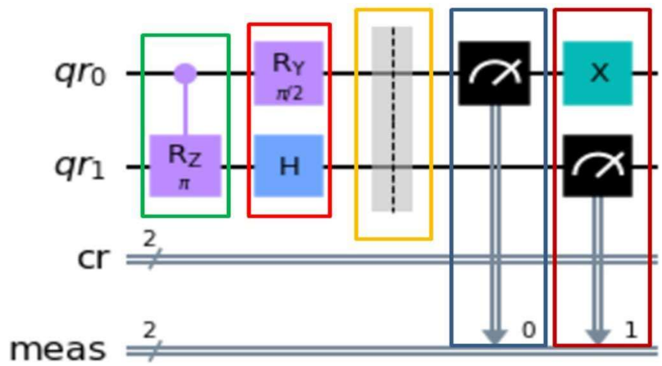
46) Which of the following codes is used to decompose the given circuit?

```
qc = QuantumCircuit(3)
qc.x(0)
qc.cx(0,1)
qc.ccx(0,1,2)
#Insert your code here
```

a) qc.decompose()
b) qc.decomposed()
c) qc.get_decomposed()
d) qc.get_decompose(qc)

47) Given the image after executing the code given below. Choose the option that doesn't fit or match the code.

```
q = QuantumRegister(2, 'qr')
c = ClassicalRegister(2, 'cr')
qc = QuantumCircuit(q,c)
qc.crz(pi,0,1)
qc.ry(pi/2,0)
qc.h(1)
qc.measure_all()
qc.x(0)
```



a) Yellow box(Barrier)
b) Green box(Rz gate)
c) Red box(Ry gate and Hadamard gate)
d) Blue box(Measurement)
e) Maroon box(X gate and Measurement)
f) None of the above

48) Which of the following gates given below are multi qubit gates?

a) cry(pi,0,1)
b) rz(pi/4)
c) ccx(0,1,2)
d) cx(0,1)
e) All of the above

49) Given the unitary matrix, which one of the following codes given below builds a circuit with unitary gate applied to the first 2 qubits?

```
U = [[1,0,0,0],
     [0,0,0,1],
     [0,1,0,0],
     [0,0,1,0]]
```

a) qc = QuantumCircuit(3)
   qc.unitary(U, [1,2])

b) qc = QuantumCircuit(3)
   qc.get_unitary(U, [0,2])
c) qc = QuantumCircuit(2)
   qc.get_unitary(U, [1,2])
d) qc = QuantumCircuit(3)
   qc.unitary(U, [0,1])

50) There are two qubits where an X gate is applied to qubit 0 and a H gate to qubit 1. What will be the output if they are measured?

a) '00'
b) '10' & '01'
c) '01' & '11'
d) '11' & '00'

Answers

1	C	26	C
2	B C	27	C
3	A	28	A C E H
4	A F	29	C
5	D E	30	A
6	C	31	A
7	A B F	32	A D
8	C	33	A C D F
9	B D	34	C
10	B	35	B
11	C	36	B
12	D	37	B
13	B	38	D
14	B	39	A C D
15	A	40	A
16	C D	41	C
17	A	42	C
18	C	43	D
19	B	44	A
20	B C E	45	C
21	E	46	A
22	C	47	F
23	D	48	A C D
24	C	49	D
25	A	50	C