Spring 2024 Mar 27

Sam Bieberich

What are we doing now?

Dear Mr. Hunt,

I hope this message finds you well. My name is Sam Bieberich, and I serve as the President of QuantA&M, the quantum computing club at Texas A&M University. I trust you remember our club, as you spoke to our members and IEEE last semester, with Akshat serving as your main point of contact.

I am reaching out to inquire about the 2024 IBMQ Developer Exam for certification. A small group of our members, including myself, is diligently preparing for the exam as part of our club curriculum this year, and we are planning to take it during the summer. However, we are facing challenges in locating information about when and how the exam will be offered.

Additionally, I was wondering if you could provide insights into any scholarships available for undergraduate students undertaking the exam. Last year, there were $200/person scholarships, and we are eager to explore options that would allow QuantA&M members to benefit from discounted rates. Our semester fund is limited to $500, which we are considering allocating towards an exam fund.

Your guidance and assistance on these matters would be greatly appreciated. Thank you for your time, and I look forward to hearing from you soon.

What are we doing now?

Hi Sam,

Here’s what I could find out.

- Unfortunately, there are no vouchers available for the certification exam
- Scheduling is up to what Pearson provides, so I guess what you see from the website.
- A new Qiskit 1.0 exam will be coming out “later this year”

My colleague recommended:

- continue taking the courses on the learning platform
- participate in the Challenge this spring (site goes live in May),
- participate in Summer School (details coming but last year’s is here: https://qiskit.org/events/summer-school-2023)
- in the fall to host an event on campus through the Qiskit Fall Fest

What are we doing now?

Thank you for the information and for forwarding Jody's contact, I will be sure to reach out soon! We have a relatively small club, so I am hoping that further collaboration with IBMQ will help more students at Texas A&M be exposed to the interesting and growing field of quantum technologies and computing!

To clarify, do you not recommend taking the exam this summer (until the new one comes out at least)? A few of our members have expressed to me that the cost is not a deal-breaker sadly, so if we can't get funding from A&M they may wait until after graduation. I was under the impression that the Certified Associate Developer - Quantum Computation certification was a prerequisite for students interested in IBMQ internships, or at least highly recommended. Would you consider that not the case?

Also, will tickets be required once again for the summer school? I was fortunate to get a spot last year through my internship at a national lab, but I heard from our QuantA&M contact professor at the time that we had only two tickets for students at in all of Texas A&M. We have 7 very active members who are looking for opportunities with IBM's programs, as well as close to a dozen other students who have (digitally) expressed interest for next year's summer school

After this, he has not answered

What am I planning right now?

1. Finish the studying for the exam regardless, since we basically have gone through the Qiskit textbook
2. Plead for funds from Jody/CS dept.
   a. Who wants to do the exam for $200?
3. Look toward next semester book (Nielsen and Chuang)
4. Pizza party as planned regardless, since we finished one book finally
5. Next time I will be gone, we can either meet with Arjun or Andrii leading or just not meet. I have slides from last summer school I could share to go over to finish off our study

What about QRISE?

Also just an update, we have a QRISE group but nobody has done anything yet (lol). I think it is safe to say that we will not have a deliverable.

I am personally too busy to finish something this next week or two (due April 12), and I think a better use of our time would be to do the summer school studying like Mr. Hunt said to do.

I do think we can start a project soon, and a good first step would be a blog:

End of Qiskit Textbook Alternative

If we can’t get the exam certification to work, and/or it seems like it is not worth it, I think we ought to design some other deliverable to display our studies this last year (for potential recruiters).

Final products will be displayed on the QuantA&M website (which is being reworked by some CS majors with Samuel as we speak).

Potential Ideas

- Blog post explaining some part of Qiskit textbook (could be presented theoretically to students at A&M)
- Informative video about the textbook
- Short research paper (linked to the website)
- Propose a Ingenium blog post or other news post

https://engineering.tamu.edu/news/4174/submit-a-story.html

Anyways, back to the grind

Number of questions: 60
Number of questions to pass: 44

Time allowed: 90 minutes
Status: Live

Section 1: Perform Operations on Quantum Circuits	47%	
Section 2: Executing Experiments	3%	
Section 3: Implement BasicAer: Python-based Simulators	3%	
Section 4: Implement Qasm	1%	
Section 5: Compare and Contrast Quantum Information	10%	
Section 6: Return the Experiment Results	7%	
Section 7: Use Qiskit Tools	1%	
Section 8: Display and Use System Information	3%	
Section 9: Construct Visualizations	19%	
Section 10: Access Aer Provider	6%	

https://www.ibm.com/training/certification/ibm-certified-associate-developer-quantum-computation-using-qiskit-v02x-C0010300

Exam 1

1. **Which statement will create a quantum circuit with four quantum bits and four classical bits?**

   A. `QuantumCircuit(4, 4)`
   B. `QuantumCircuit(4)`
   C. `QuantumCircuit(QuantumRegister(4, 'qr0'), QuantumRegister(4, 'cr1'))`
   D. `QuantumCircuit([4, 4])`

1. **Which statement will create a quantum circuit with four quantum bits and four classical bits?**

   A. `QuantumCircuit(4, 4)`

   B. `QuantumCircuit(4)`

   C. `QuantumCircuit(QuantumRegister(4, 'qr0'), QuantumRegister(4, 'cr1'))`

   D. `QuantumCircuit([4, 4])`

A

2. **Given this code fragment, what is the probability that a measurement would result in |0> ?**

```
qc = QuantumCircuit(1)
qc.ry(3 * math.pi/4, 0)
```

A. 0.8536
B. 0.5
C. 0.1464
D. 1.0

2. Given this code fragment, what is the probability that a measurement would result in |0> ?

```
qc = QuantumCircuit(1)
qc.ry(3 * math.pi/4, 0)
```

A. 0.8536
B. 0.5
C. 0.1464
D. 1.0

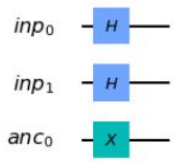
https://bloch.kherb.io/

C

3. **Assuming the fragment below, which three code fragments would produce the circuit illustrated?**

```
inp_reg = QuantumRegister(2, name='inp')
ancilla = QuantumRegister(1, name='anc')
qc = QuantumCircuit(inp_reg, ancilla)

# Insert code here
```



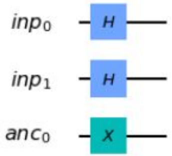
A.
```
qc.h(inp_reg)
qc.x(ancilla)
qc.draw()
```
B.
```
qc.h(inp_reg[0:2])
qc.x(ancilla[0])
qc.draw()
```
C.
```
qc.h(inp_reg[0:1])
qc.x(ancilla[0])
qc.draw()
```
D.
```
qc.h(inp_reg[0])
qc.h(inp_reg[1])
qc.x(ancilla[0])
qc.draw()
```
E.
```
qc.h(inp_reg[1])
qc.h(inp_reg[2])
qc.x(ancilla[1])
qc.draw()
```
F.
```
qc.h(inp_reg)
qc.h(inp_reg)
qc.x(ancilla)
qc.draw()
```

A
B
D

3. Assuming the fragment below, which three code fragments would produce the circuit illustrated?

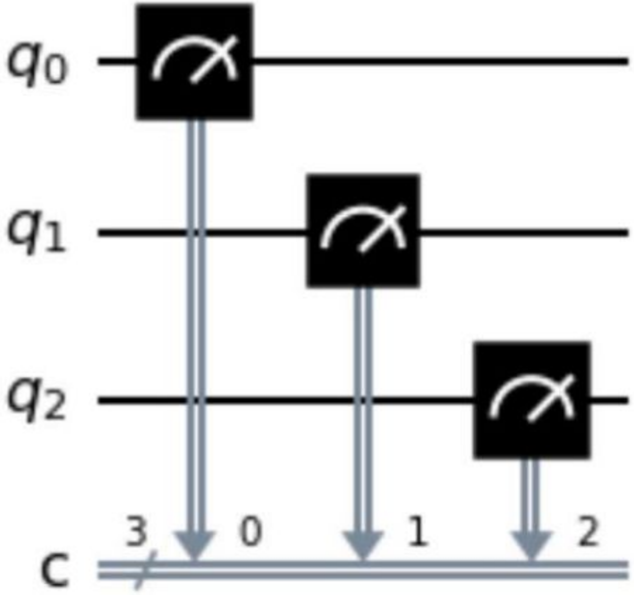
```
inp_reg = QuantumRegister(2, name='inp')
ancilla = QuantumRegister(1, name='anc')
qc = QuantumCircuit(inp_reg, ancilla)

# Insert code here
```



A.
```
qc.h(inp_reg)
qc.x(ancilla)
qc.draw()
```
B.
```
qc.h(inp_reg[0:2])
qc.x(ancilla[0])
qc.draw()
```
C.
```
qc.h(inp_reg[0:1])
qc.x(ancilla[0])
qc.draw()
```
D.
```
qc.h(inp_reg[0])
qc.h(inp_reg[1])
qc.x(ancilla[0])
qc.draw()
```
E.
```
qc.h(inp_reg[1])
qc.h(inp_reg[2])
qc.x(ancilla[1])
qc.draw()
```
F.
```
qc.h(inp_reg)
qc.h(inp_reg)
qc.x(ancilla)
qc.draw()
```

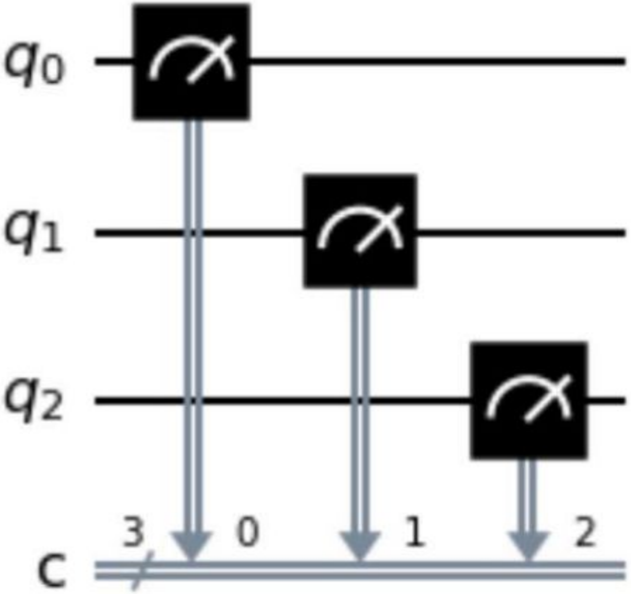
4. **Given an empty QuantumCircuit object, qc, with three qubits and three classical bits, which one of these code fragments would create this circuit?**



A. `qc.measure([0,1,2], [0,1,2])`
B. `qc.measure([0,0], [1,1], [2,2])`
C. `qc.measure_all()`
D. `qc.measure(0,1,2)`

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

A



A. `qc.measure([0,1,2], [0,1,2])`
B. `qc.measure([0,0], [1,1], [2,2])`
C. `qc.measure_all()`
D. `qc.measure(0,1,2)`

5. **Which code fragment will produce a maximally entangled, or Bell, state?**

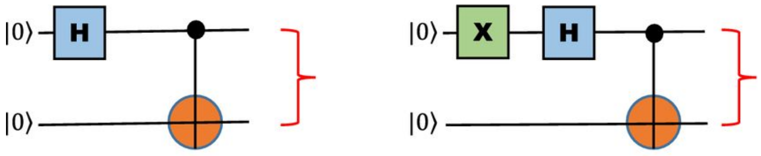
A.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.x(1)
bell.cx(0, 1)
```
B.
```
bell = QuantumCircuit(2)
bell.cx(0, 1)
bell.h(0)
bell.x(1)
```
C.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.x(1)
bell.cz(0, 1)
```
D.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.h(0)
```

5. **Which code fragment will produce a maximally entangled, or Bell, state?**

A.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.x(1)
bell.cx(0, 1)
```
B.
```
bell = QuantumCircuit(2)
bell.cx(0, 1)
bell.h(0)
bell.x(1)
```
C.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.x(1)
bell.cz(0, 1)
```
D.
```
bell = QuantumCircuit(2)
bell.h(0)
bell.h(0)
```

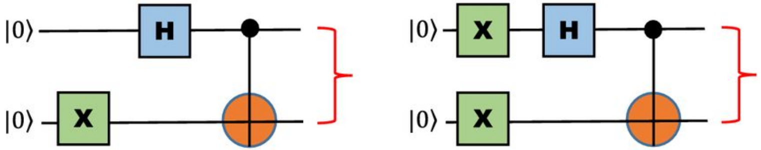
A

Don’t forget there are four bell states, they are operated on each by a H and a CX gate, but can start in any orientation



$$|\psi^{+}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$|\psi^{-}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$



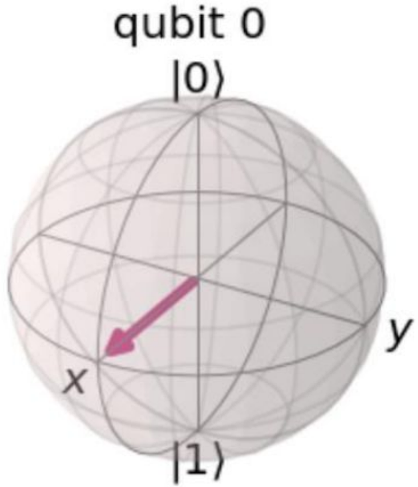
$$|\phi^{+}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

$$|\phi^{-}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

6. 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)
```



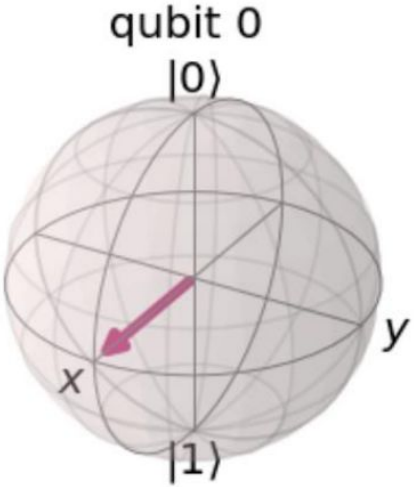
A. `qc.h(0)`
B. `qc.rx(math.pi / 2, 0)`
C. `qc.ry(math.pi / 2, 0)`
D. `qc.rx(math.pi / 2, 0)`
`qc.rz(-math.pi / 2, 0)`
E. `qc.ry(math.pi, 0)`

A
C

6. **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.h(0)`
B. `qc.rx(math.pi / 2, 0)`
C. `qc.ry(math.pi / 2, 0)`
D. `qc.rx(math.pi / 2, 0)`
`qc.rz(-math.pi / 2, 0)`
E. `qc.ry(math.pi, 0)`

7. **S-gate is a Qiskit phase gate with what value of the phase parameter?**

   A. π/4
   B. π/2
   C. π/8
   D. π

7. **S-gate is a Qiskit phase gate with what value of the phase parameter?**

A. $\pi/4$
B. $\pi/2$
C. $\pi/8$
D. π

B

8. Which two code fragments, when inserted into the code below, will produce the statevector shown in the output?

```
from qiskit import QuantumCircuit, Aer, execute
from math import sqrt

qc = QuantumCircuit(2)

# Insert fragment here

simulator = Aer.get_backend('statevector_simulator')
result = execute(qc, simulator).result()
statevector = result.get_statevector()
print(statevector)
```

Output:

```
[0.707+0.j  0.+0.j  0.+0.j  0.707+0.j]
```

A.
```
v = [1/sqrt(2), 0, 0, 1/sqrt(2)]
qc.initialize(v,[0,1])
```
B.
```
qc.h(0)
qc.cx(0,1)
```
C.
```
v1, v2 = [1,0], [0,1]
qc.initialize(v1,0)
qc.initialize(v2,1)
```
D.
```
qc.cx(0,1)
qc.measure_all()
```
E.
```
qc.h(0)
qc.h(1)
qc.measure_all()
```

A
B

8. Which two code fragments, when inserted into the code below, will produce the statevector shown in the output?

```
from qiskit import QuantumCircuit, Aer, execute
from math import sqrt

qc = QuantumCircuit(2)

# Insert fragment here

simulator = Aer.get_backend('statevector_simulator')
result = execute(qc, simulator).result()
statevector = result.get_statevector()
print(statevector)
```

Output:

```
[0.707+0.j  0.+0.j  0.+0.j  0.707+0.j]
```

A.
```
v = [1/sqrt(2), 0, 0, 1/sqrt(2)]
qc.initialize(v,[0,1])
```
B.
```
qc.h(0)
qc.cx(0,1)
```
C.
```
v1, v2 = [1,0], [0,1]
qc.initialize(v1,0)
qc.initialize(v2,1)
```
D.
```
qc.cx(0,1)
qc.measure_all()
```
E.
```
qc.h(0)
qc.h(1)
qc.measure_all()
```

9. **Which code fragment will produce a multi-qubit gate other than a CNOT ?**

   A. `qc.cx(0,1)`
   B. `qc.cnot(0,1)`
   C. `qc.mct([0],1)`
   D. `qc.cz(0,1)`

9. **Which code fragment will produce a multi-qubit gate other than a CNOT ?**

A. `qc.cx(0,1)`
B. `qc.cnot(0,1)`
C. `qc.mct([0],1)`
D. `qc.cz(0,1)`

D

10. **Which code fragment will produce a multi-qubit gate other than a Toffoli?**

A. `qc.ccx(0,1,2)`

B. `qc.mct([0,1], 2)`

C.
```
from qiskit.circuit.library import CXGate
ccx = CXGate().control()
qc.append(ccx, [0,1,2])
```

D. `qc.cry(0,1,2)`

10. Which code fragment will produce a multi-qubit gate other than a Toffoli?

A. `qc.ccx(0,1,2)`

B. `qc.mct([0,1], 2)`

C.
```
from qiskit.circuit.library import CXGate
ccx = CXGate().control()
qc.append(ccx, [0,1,2])
```

D. `qc.cry(0,1,2)`

D

MCT = Multi-Control Target

11. Which two options would place a barrier across all qubits to the QuantumCircuit below?

```
qc = QuantumCircuit(3,3)
```

A. `qc.barrier(qc)`
B. `qc.barrier([0,1,2])`
C. `qc.barrier()`
D. `qc.barrier(3)`
E. `qc.barrier_all()`

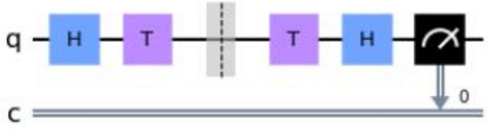
11. Which two options would place a barrier across all qubits to the QuantumCircuit below?

```
qc = QuantumCircuit(3,3)
```

A. `qc.barrier(qc)`
B. `qc.barrier([0,1,2])`
C. `qc.barrier()`
D. `qc.barrier(3)`
E. `qc.barrier_all()`

B
C

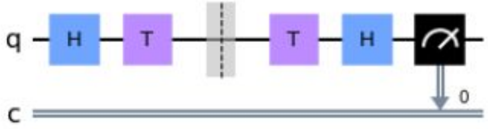
12. What code fragment codes the equivalent circuit if you remove the barrier in the following QuantumCircuit?



S=T^2

A.
```
qc = QuantumCircuit(1,1)
qc.h(0)
qc.s(0)
qc.h(0)
qc.measure(0,0)
```
B.
```
qc = QuantumCircuit(1,1)
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.z(0)
qc.h(0)
qc.measure(0,0)
```

12. **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.s(0)
qc.h(0)
qc.measure(0,0)
```
B.
```
qc = QuantumCircuit(1,1)
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.z(0)
qc.h(0)
qc.measure(0,0)
```

A

13. Given the following code, what is the depth of the circuit?

```
qc = QuantumCircuit(2, 2)

qc.h(0)
qc.barrier(0)
qc.cx(0,1)
qc.barrier([0,1])
```

A. 2
B. 3
C. 4
D. 5

13. **Given the following code, what is the depth of the circuit?**

```
qc = QuantumCircuit(2, 2)

qc.h(0)
qc.barrier(0)
qc.cx(0,1)
qc.barrier([0,1])
```

A

A. 2
B. 3
C. 4
D. 5

14. Which code snippet would execute a circuit given these parameters?

1) • Measure the circuit 1024 times,
2) • use the QASM simulator,
3) • and use a coupling map that connects three qubits linearly

```
qc = QuantumCircuit(3)

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

A.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend=qasm_sim, shots=1024,
coupling_map=couple_map)
```

B.
```
qasm_sim = Aer.getBackend('ibmq_simulator')
couple_map = [[0, 1], [0, 2]]
job = execute(qc, loop=1024, coupling_map=couple_map)
```

C.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend=qasm_sim, repeat=1024,
coupling_map=couple_map)
```

D.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(backend=qasm_sim, qc, shot=1024,
coupling_map=couple_map)
```

14. Which code snippet would execute a circuit given these parameters?

1) • Measure the circuit 1024 times,
2) • use the QASM simulator,
3) • and use a coupling map that connects three qubits linearly

```
qc = QuantumCircuit(3)

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

A.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend=qasm_sim, shots=1024,
coupling_map=couple_map)
```

B.
```
qasm_sim = Aer.getBackend('ibmq_simulator')
couple_map = [[0, 1], [0, 2]]
job = execute(qc, loop=1024, coupling_map=couple_map)
```

C.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(qc, backend=qasm_sim, repeat=1024,
coupling_map=couple_map)
```

D.
```
qasm_sim = Aer.get_backend('qasm_simulator')
couple_map = [[0, 1], [1, 2]]
job = execute(backend=qasm_sim, qc, shot=1024,
coupling_map=couple_map)
```

A

15. Which of these would execute a circuit on a set of qubits which are coupled in a custom way?

```
from qiskit import QuantumCircuit, execute, BasicAer
backend = BasicAer.get_backend('qasm_simulator')
qc = QuantumCircuit(3)

# insert code here
```

A. `execute(qc, backend, shots=1024, coupling_map=[[0,1], [1,2]])`

B. `execute(qc, backend, shots=1024, custom_topology=[[0,1],[2,3]]`

C. `execute(qc, backend, shots=1024, device="qasm_simulator", mode="custom")`

D. `execute(qc, backend, mode="custom")`

15. **Which of these would execute a circuit on a set of qubits which are coupled in a custom way?**

```
from qiskit import QuantumCircuit, execute, BasicAer
backend = BasicAer.get_backend('qasm_simulator')
qc = QuantumCircuit(3)

# insert code here
```

A. `execute(qc, backend, shots=1024, coupling_map=[[0,1], [1,2]])`
B. `execute(qc, backend, shots=1024, custom_topology=[[0,1],[2,3]]`
C. `execute(qc, backend, shots=1024, device="qasm_simulator", mode="custom")`
D. `execute(qc, backend, mode="custom")`

A

16. **Which three simulators are available in BasicAer?**

A. `qasm_simulator`
B. `basic_qasm_simulator`
C. `statevector_simulator`
D. `unitary_simulator`
E. `quantum_simulator`
F. `quantum_circuit_simulator`

16. Which three simulators are available in BasicAer?

A. `qasm_simulator`
B. `basic_qasm_simulator`
C. `statevector_simulator`
D. `unitary_simulator`
E. `quantum_simulator`
F. `quantum_circuit_simulator`

A
C
D

Basic_qasm dne
Quantum_simulator
and
quantum_circuit_si
mulator = qasm

17. Which line of code would assign a statevector simulator object to the variable `backend` ?

A. `backend = BasicAer.StatevectorSimulatorPy()`
B. `backend = BasicAer.get_backend('statevector_simulator')`
C. `backend = BasicAer.StatevectorSimulatorPy().name()`
D. `backend = BasicAer.get_back('statevector_simulator')`

17. Which line of code would assign a statevector simulator object to the variable `backend` **?**

A. `backend = BasicAer.StatevectorSimulatorPy()`
B. `backend = BasicAer.get_backend('statevector_simulator')`
C. `backend = BasicAer.StatevectorSimulatorPy().name()`
D. `backend = BasicAer.get_back('statevector_simulator')`

B

18. Which code fragment would yield an operator that represents a single-qubit X gate?

A. `op = Operator.Xop(0)`

B. `op = Operator([[0,1]])`

C.
```
qc = QuantumCircuit(1)
qc.x(0)
op = Operator(qc)
```

D. `op = Operator([[1,0,0,1]])`

18. Which code fragment would yield an operator that represents a single-qubit X gate?

A. `op = Operator.Xop(0)`

B. `op = Operator([[0,1]])`

C. `qc = QuantumCircuit(1)`
```
qc.x(0)
op = Operator(qc)
```

D. `op = Operator([[1,0,0,1]])`

C

19. What would be the fidelity result(s) for these two operators, which differ only by global phase?

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

A. `state_fidelity()` of 1.0

B. `state_fidelity()` and `average_gate_fidelity()` of 1.0

C. `average_gate_fidelity()` and `process_fidelity()` of 1.0

D. `state_fidelity()`, `average_gate_fidelity()` and `process_fidelity()` of 1.0

19. What would be the fidelity result(s) for these two operators, which differ only by global phase?

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

A. `state_fidelity() of 1.0`

B. `state_fidelity() and average_gate_fidelity() of 1.0`

C. `average_gate_fidelity() and process_fidelity() of 1.0`

D. `state_fidelity(), average_gate_fidelity() and process_fidelity() of 1.0`

C

[Operators module overview | IBM Quantum Documentation]

[qiskit-sample-test-answers/jupyter/Question-19.ipynb at master · andre-a-alves/qiskit-sample-test-answers · GitHub]

20. Given this code fragment, which output fits most closely with the measurement probability distribution?

```
qc = QuantumCircuit(2, 2)
qc.x(0)
qc.measure([0,1], [0,1])
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1000).result()
counts  = result.get_counts(qc)
print(counts)
```

A. `{'00': 1000}`
B. `{'01': 1000}`
C. `{'10': 1000}`
D. `{'11': 1000}`

20. Given this code fragment, which output fits most closely with the measurement probability distribution?

```
qc = QuantumCircuit(2, 2)
qc.x(0)
qc.measure([0,1], [0,1])
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1000).result()
counts  = result.get_counts(qc)
print(counts)
```

A. `{'00': 1000}`
B. `{'01': 1000}`
C. `{'10': 1000}`
D. `{'11': 1000}`

B

End Exam 1