IBM Professional Certification Program

SAMPLE TEST

Exam C1000-179

Fundamentals of Quantum Computing Using Qiskit v2.X Developer

Professional
Certification
Program

These questions were developed at the same time and by the same subject matter experts as the actual certification exam questions. While these sample questions will give you a good idea of the nature of the questions on the real exam, this is not a thorough representation of the material covered by the real exam, so success with these sample questions should not be considered predictive of success on the real exam.

For a more realistic idea of your readiness for the real certification exam, we suggest you take the full-length Assessment Test available from Pearson VUE. Search for **C1000-179** or **Fundamentals of Quantum Computing Using Qiskit v2.X Developer**

Section 1: Perform quantum operations

1. Which one of the following code fragments will generate the given output?

```
[[ 1.+0.j  0.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j -1.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  0.+0.j -1.+0.j]]
```

a.
```
p = Pauli('IZ')
print(p.to_matrix())
```

b.
```
p = Pauli('-II')
print(p.to_matrix())
```

c.
```
p = Pauli('-ZI')
print(p.to_matrix())
```

d.
```
p = Pauli('ZZ')
print(p.to_matrix())
```

2. Applying the Qiskit TGate to a qubit in state |1> introduces which global phase?

a. π/2 phase
b. -π/2 phase
c. -π/4 phase
d. π/4 phase

3. Given the following code fragment, what is the approximate probability that a measurement would result in a bit value of 1?

```
from qiskit import QuantumCircuit
import numpy as np

qc = QuantumCircuit(1)
qc.reset(0)
qc.ry(np.pi / 2, 0)
qc.measure_all()
```

a. 0.8536

b. 1.0

c. 0.1464

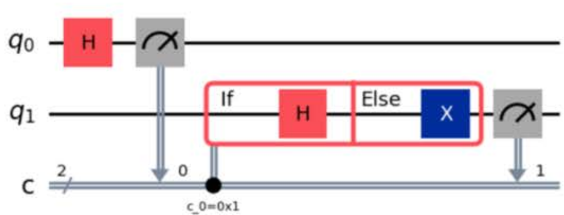
d. 0.5

<u>Section 2: Visualize quantum circuits, measurements, and states</u>

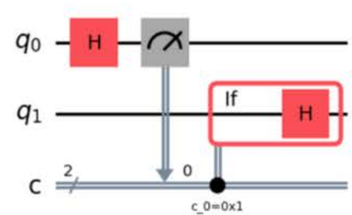
4. Which one of the following images is the output from the code below:

```
from qiskit import *

qubits = QuantumRegister(2)
clbits = ClassicalRegister(2)
circuit = QuantumCircuit(qubits, clbits)
(q0, q1) = qubits
(c0, c1) = clbits

circuit.h(q0)
circuit.measure(q0, c0)
with circuit.if_test((c0, 1)) as else_:
    circuit.h(q1)
with else_:
    circuit.x(q1)
circuit.measure(q1, c1)

circuit.draw(output="mpl")
```

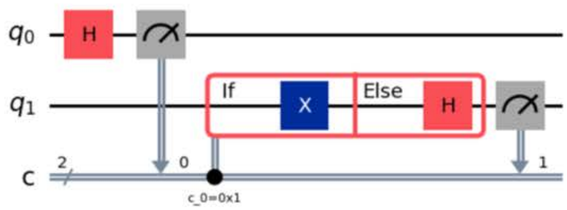
a.



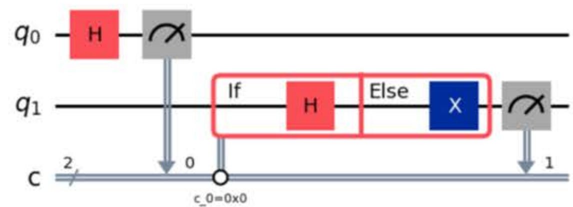
b.



c.



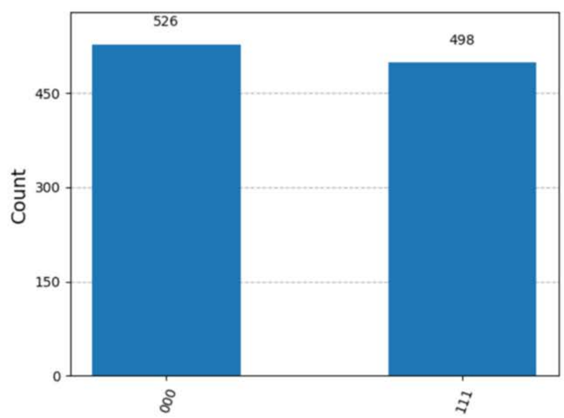
d.



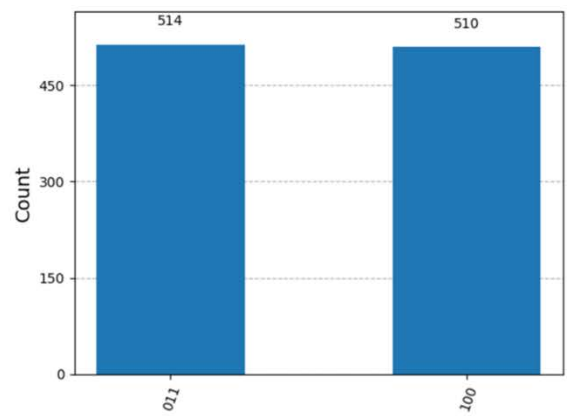
5. Given the code fragment below, which image is the expected output?

```
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_histogram

state = Statevector([0.+0.j, 0.+0.j, 0.70710678+0.j, 0.+0.j,
                     0.+0.j, -0.70710678+0.j, 0.+0.j, 0.+0.j])
counts = state.sample_counts(shots=1024)
plot_histogram(counts)
```

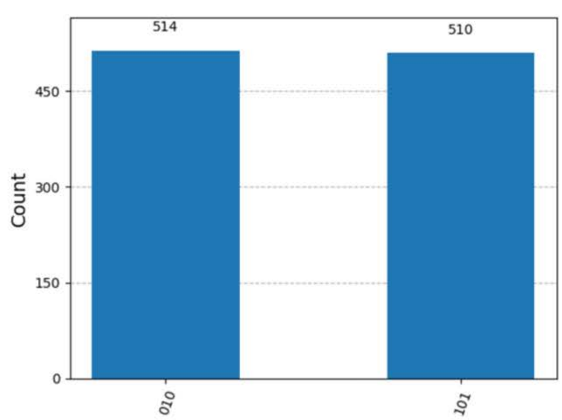
a.



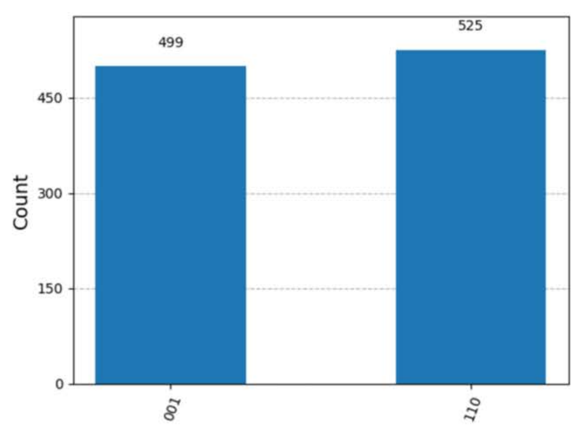
b.



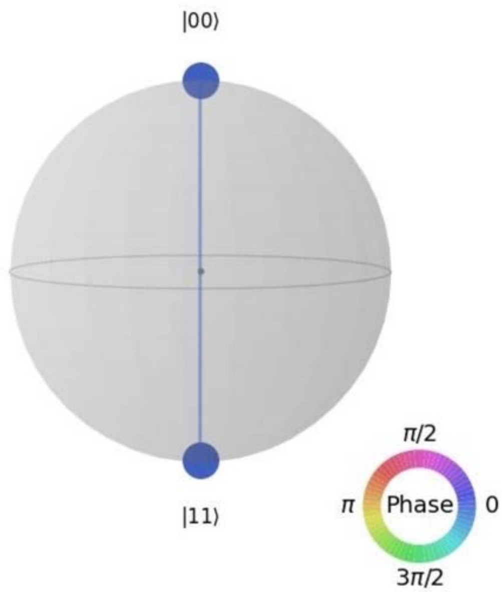
c.



d.



6. **Which one of the following code fragments will generate the given qsphere representation visualization?**
   Note: the circles on the qsphere are the same color.



a.
```
qc = QuantumCircuit(2)
qc.h(0)
qc.z(0)
qc.cx(0, 1)
state = Statevector(qc)
plot_state_qsphere(state)
```

b.
```
qc = QuantumCircuit(2)
qc.h(0)
qc.z(0)
qc.x(1)
qc.cx(0, 1)
state = Statevector(qc)
plot_state_qsphere(state)
```

c.
```
qc = QuantumCircuit(2)
qc.x(1)
qc.h(0)
qc.cx(0, 1)
state = Statevector(qc)
plot_state_qsphere(state)
```

d.
```
qc = QuantumCircuit(2)
qc.h(0)
qc.cx(0, 1)
state = Statevector(qc)
plot_state_qsphere(state)
```

Section 3: Create quantum circuits

7. Given the code fragment below, which of the following code fragments creates a rotation gate with an angle with an initially undefined value?

```
from qiskit.circuit import QuantumCircuit, Parameter,
     ParameterExpression
qc = QuantumCircuit(1)
```

a.
```
theta = 3.14
qc.rx(3.14, 0)
```

b.
```
theta = Parameter('theta')
qc.rx(theta, 0)
```

c. `qc.rx('theta', 0)`

d. `qc.rx(ParameterExpression('theta'), 0)`

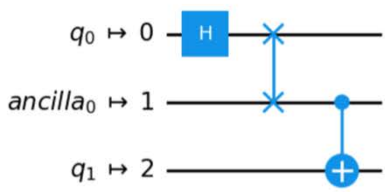
8. Which one of the following types of register stores the result of a measured circuit?

a. Ancillary register
b. Quantum register
c. Classical register
d. Circuit register

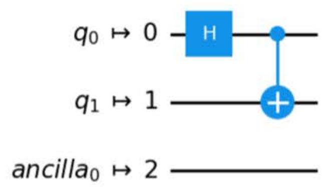
9. Given the code fragment below, which one of the following images could be produced?

```
from qiskit import QuantumCircuit
from qiskit import generate_preset_pass_manager

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

pass_manager = generate_preset_pass_manager(
         optimization_level=3,
         coupling_map=[[0, 1], [1, 2]] ,
         basis_gates=['h', 'swap', 'cx'],
         initial_layout=[0, 2]
)
tqc = pass_manager.run(qc)
tqc.draw(output="mpl")
```

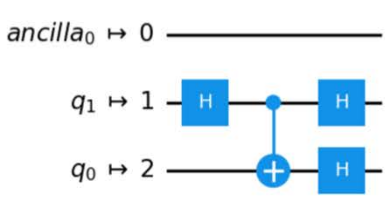
a.



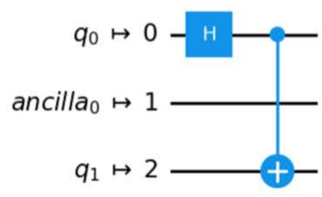
b.



c.



d.



Section 4: Run quantum circuits

10. Which three of the following are job execution modes in Qiskit Runtime?

a. classical
b. session
c. parallel
d. quantum
e. batch
f. single job

11. Which code fragment is the correct way to open a session?

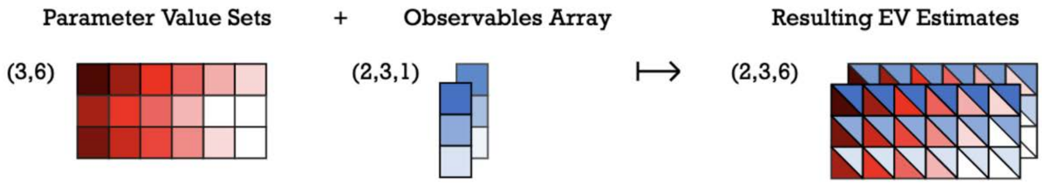
a.
```
from qiskit_ibm_runtime import Session
session = Session(system='ibm_foo')
```

b.
```
from qiskit_ibm_runtime import execute, QiskitRuntimeService
service = QiskitRuntimeService()
session = execute(service=service)
```

c.
```
from qiskit_ibm_runtime import Session, QiskitRuntimeService
service = QiskitRuntimeService()
session = Session(service.least_busy())
```

d.
```
from qiskit import QuantumCircuit
session = QuantumCircuit(2).open_session()
```

12. Which one of the following patterns, expressed in terms of array broadcasting primitives, is represented by the given image?



a. Standard multidimensional array generalization

b. All-to-all

c. Extended dimensional variation

d. Best effort broadcasting

Section 5: Use the sampler primitive

13. Given the following code fragment, which one of the following describes the `SamplerOptions` **parameter** `options.default_shots`**?**

```
...
from qiskit_ibm_runtime import Sampler

sampler = Sampler(mode=backend)
sampler.options.default_shots = ...
```

a. The sum of the number of measurements in each qubit

b. The number of randomizations we apply to the circuit

c. The number of times that we run the circuit

d. The number of sequences in dynamical decoupling

14. Given the code snippet, which one of the following is a valid way to invoke the `run` **method on an instance of** `SamplerV2`**?**

```
from qiskit_ibm_runtime import SamplerV2
...

sampler = SamplerV2(...)
```

a. `sampler.run([isa_circuit])`

b. `sampler.run(distribution, isa_circuit)`

c. `sampler.run(isa_circuit, distribution='gauss')`

d. `sampler.run([isa_circuit1, isa_circuit2], runs=1024)`

Section 6: Use the estimator primitive

15. Which one of the following describes the expected behavior of the number of shots if the value for the parameter `precision` were changed from `0.015625` to `0.03125`?

a. It increases the number of shots quadratically

b. It increases the number of shots exponentially

c. It has no effect on the number of shots

d. It decreases the number of shots

16. Which error mitigation technique can be applied using resilience options?

a. Pauli twirling

b. Dynamical decoupling

c. Zero Noise Extrapolation

d. Full quantum error correction

17. Which format should a primitive unified bloc (PUB) tuple follow for the Estimator primitive?

a. `pub = (circuit, observable, parameter_values, backend)`

b. `pub = (circuit, observable, parameter_values, precision)`

c. `pub = (circuit, observable, shots, optimization_level)`

d. `pub = (circuit, observable, resilience_level, noise_model)`

Section 7: Retrieve and analyze the results of quantum circuits

18. Which statement describes the purpose of a Qiskit Runtime session?

a. Automatically generate quantum algorithms based on user input

b. Visualise the results of quantum experiments in real time

c. Group a collection of calls to the quantum computer

d. Compile and optimise quantum circuits for different backends

19. Which two of the following pieces of information are part of the dictionary returned by `session.details()`**, assuming that** `session` **is an instance of** `qiskit_ibm_runtime.Session`**?**

a. Quantum circuit depth

b. Timestamp of the last job in the session that completed

c. Session state

d. Primitive options

e. Primitive unified blocs (PUBs) in each job

Section 8: Operate with OpenQASM

20. Which one of the following is a classical data type supported by OpenQASM 3?

a. `complex`

b. `class`

c. `char`

d. `enum`

21. Which method should be used to export a Qiskit circuit named `qc` to OpenQASM 3 and store it into a file stream named `qasmprogram`?

a. `qc.to_openqasm3(qasmprogram)`

b. `qiskit.qasm3.dump(qc, qasmprogram)`

c. `qasmprogram.export_to_qasm3(qc)`

d. `qiskit.qasm3.export(qc, qasmprogram)`

Answer Key

1. a
2. d
3. d
4. a
5. c
6. d
7. b
8. c
9. a
10. b, e, f
11. c
12. a
13. c
14. a
15. d
16. c
17. b
18. c
19. b, c
20. a
21. b