虎の巻：サンプル問題の解説

サンプル問題1の解説

問 題

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])`

解 説

4つの量子ビットと4つの古典ビットをもつ量子回路をつくるのにふさわしいコードを選択肢の中から選ぶ問題である。

QuantumCircuitの引数では、Intのデータが2つ並ぶ場合にはそれぞれ指定された個数の量子ビット、古典ビットをもつ量子回路をつくることを意味する。選択肢Aはこのケースであり、問題文のとおり量子回路を作ることができるので、正解である。

選択肢Bは4量子ビットの量子回路をつくるが、古典ビットについては未定義であるため、問題文のとおりの回路とはならず、正解ではない。

選択肢Cは4個ずつの量子ビットの組qr0、cr1をもつ量子回路を作る記述になっている。2つめが古典レジスタではないため、不正解である。

選択肢Dは量子ビット、古典ビットの個数を指定するのに[ ]で括っており、記述のしかたが誤っている。このため不正解である。

従って、正解は選択肢Aである。

参 考

https://qiskit.org/documentation/locale/ja_JP/stubs/qiskit.circuit.QuantumCircuit.html

サンプル問題2の解説

問 題

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

解 説

y軸周りに3/4π回転するゲートを作用させる1量子ビットの回路がある。この回路を実行したときに|0>が測定される確率を選択肢から選ぶ問題である。

ryゲートはy軸周りに回転するゲートであり、

$$Ry(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

従って、

$$Ry(3/4\pi)\,|0\rangle = \begin{bmatrix} \cos\frac{3}{8}\pi & -\sin\frac{3}{8}\pi \\ \sin\frac{3}{8}\pi & \cos\frac{3}{8}\pi \end{bmatrix} |0\rangle$$

これを状態ベクトル

$$\phi = \alpha\,|0\rangle + \beta\,|1\rangle$$

で書くと、

$$\alpha = \cos\frac{3}{8}\pi, \beta = \sin\frac{3}{8}\pi$$

である。
測定をした際に|0>が観測される確率は状態ベクトルの振幅の2乗であるから、求める値は

$$\alpha^2 = \left(\cos\frac{3}{8}\pi\right)^2$$

である。

サンプル問題2の解説

ここで、

$$\cos\frac{3}{8}\pi < \cos\frac{4}{8}\pi = \frac{\sqrt{2}}{2} \simeq 0.7$$

であるから、

$$\alpha^2 = \left(\cos\frac{3}{8}\pi\right)^2 < 0.7^2$$

従って、回答の選択肢C　0.1464を選ぶのが正しい。

実際、右記の検算のとおり、0.1464となることがわかる。

この問題についてはブロッホ球のシミュレータでも確かめることができるが、受験では、三角関数の既知の数値との大小関係から答えを求める必要がある。

```
In [1]: import math

In [7]: beta = math.sin(math.pi/2*.75)

In [8]: alpha = math.cos(math.pi/2*.75)

In [10]: alpha ** 2 + beta ** 2
Out[10]: 1.0

In [11]: alpha
Out[11]: 0.38268343236508984

In [12]: alpha ** 2
Out[12]: 0.1464466094067263
```

サンプル問題2の解説

ウラ解説

|0〉をy軸周りに3/4π回転したときの状態ベクトルはブロッホ球の南半球側にある。

|0〉にアダマールゲートを作用させたときの状態ベクトルは

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

であるから、ブロッホ球の赤道上では|0〉、|1〉の存在確率はそれぞれ1/2である。

ブロッホ球の南半球側では、|0〉の存在確率は1/2よりも小さくなり、|1〉の存在確率は1/2より大きくなる。南極(|1〉)に近づくにつれ、|1〉の存在確率は1に近づく。

このような考察から、問題の回路での|0〉が測定される確率は0.5より小さい。また、状態ベクトルは|1〉でない。従って解としては、選択肢Cを選ぶことになる。

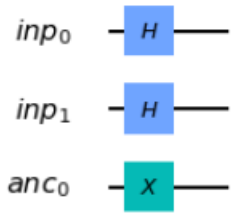
サンプル問題3の解説

問 題

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()
```

解 説

左記の図のような量子回路を構成するのに選択肢から3つ正解を選ぶ問題である。

inp0,inp1にアダマールゲートを、anc0にXゲートを作用させるコードを選択肢から選ぶ。

選択肢Aは題意の量子回路を構成できるので、正解である。

選択肢Bは題意の量子回路を構成できるので、正解である。添え字の範囲[0:2]は実際に動く範囲は0，1である。

選択肢Cは量子ビットの添え字の範囲に誤りがある。inp1にアダマールゲートがかからないため、不正解。

選択肢Dは題意の量子回路を構成できるので、正解である。アダマールゲートを作用させるinpレジスタを個別に書いている。

サンプル問題3の解説

選択肢Eは選択肢Dに似ているが、添え字の範囲が誤っているため不正解。inpレジスタの範囲はゼロから1、ancレジスタはゼロであるのが正しい。

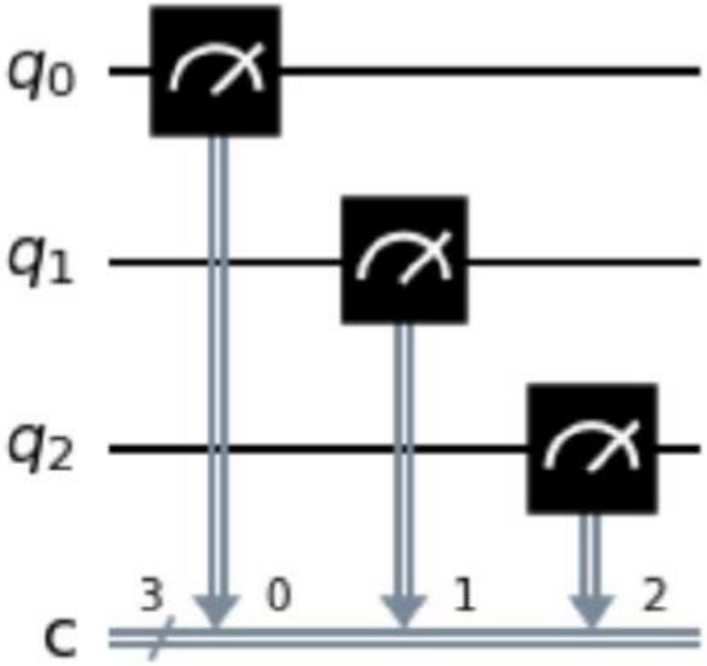
選択肢Fはアダマールをinpレジスタ全体に2回作用させており、題意とは異なるため、不正解。

従って正解は選択肢A、B、Dである。

サンプル問題4の解説

問 題

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)`

解 説

3量子ビットと3古典ビットからなる空の量子回路がある。左記の図のような回路をつくるために、どのようなコードを書けばよいか選択肢から選ぶ問題である。

QuantumCircuit.measureの書き方は
　QuantumCircuit.measure(qubit,cbit)
となっており、qubit、cbitはそれぞれ

qubit (Union[Qubit, QuantumRegister, int, slice, Sequence[Union[Qubit, int]]]) – qubit to measure.

cbit (Union[Clbit, ClassicalRegister, int, slice, Sequence[Union[Clbit, int]]]) – classical bit to place the measurement in.

となっている。

サンプル問題4の解説

選択肢の記述が、この引数の記載に合っているものを選ぶことを考えればよい。

選択肢Aは量子ビット0～2を計測し、古典ビット0～2に書くという内容になっているの正解である。

選択肢Bは[ ]で括られたデータが3つあるが、正しくは量子ビットと古典ビットの2つの組でないとおかしいので正解ではない。

選択肢Cは3つの量子ビットをいっぺんに測定できるコードだが、実際にmeasure_all()を実行すると、右図のような結果となり、問題文に示されたものとは異なる。
従って正解ではない。

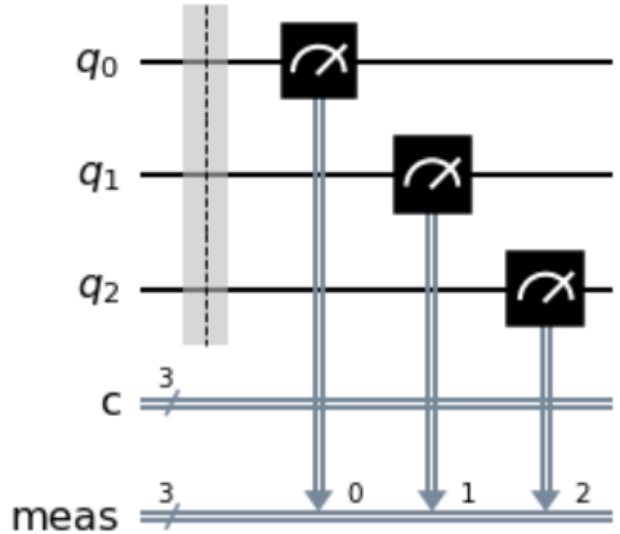
選択肢Dは引数の記載が合っておらず、syntax　errorになってしまう。

従って正解は選択肢Aである。

```
In [51]:  qc =QuantumCircuit(3,3)
          qc.measure_all()

In [47]:  qc.draw('mpl')
```

Out[47]:



参 考

https://qiskit.org/documentation/locale/ja_JP/stubs/qiskit.circuit.QuantumCircuit.measure.html

サンプル問題5の解説

問 題

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)
```

解 説

ベル状態をつくるコードを選ぶ問題である。
2量子ビットのレジスタを使うので、アダマールゲートとCNOTゲートの組み合わせとなるものがベル状態を作る量子回路である。

このため、アダマールとCNOTを含まない選択肢は不正解である。

選択肢C,DはCNOTを含まないため不正解。

選択肢Bはアダマールと CNOTを含んでいるがゲートの配置の順序が正しくない（逆）ため、ベル状態が作られず、不正解である

選択肢Aはアダマールと CNOTの順にゲートが配置されており、ベル状態がつくられる。

従って正解は選択肢Aである。

サンプル問題6の解説

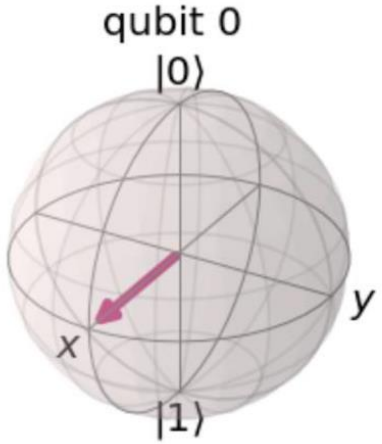
問 題

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)`



解 説

左記のコードが与えられたとき、ブロッホ球で示された状態ベクトルとなるよう、選択肢のなかから2つのコードの断片を選べという問題である。

ブロッホ球で示された状態ベクトルは|+>である。
これは、|0>にアダマールゲートを作用させた場合の状態ベクトルであり、選択肢Aは正解である。

選択肢Bは、|0>をx軸のまわりにπ/2回転させるものだが、|−i>となるため、正解ではない。

選択肢Cは|0>をy軸のまわりにπ/2回転させるもので結果は|+>となるため正解である。

選択肢Dは、|0>をx軸のまわりにπ/2回転させ、さらにz軸の周りに−π/2回転させるもの。結果は|−>となるため、正解ではない。
選択肢Eは|0>をy軸のまわりにπ回転させるもので結果は|1>となるため正解ではない。
従って正解はAとCである。

サンプル問題7の解説

問 題

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

解 説

Sゲートはフェーズゲートであるが、フェーズの値がどのような値かを問う問題である。

Pゲート（フェーズゲート）はパラメータ化されたゲートでz軸を中心にφ回転するゲートである。

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$$

Sゲートは、Pゲートで$\phi = \pi / 2$とした場合に相当する。従って正解は選択肢Bである。

参 考

https://qiskit.org/textbook/ja/ch-states/single-qubit-gates.html#sgate

サンプル問題8の解説

問 題

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()
```

解 説

アウトプットの状態ベクトルのような量子状態をつくるために、選択肢から2つ適切なコードを選び、プログラムの該当箇所に入れよという問題である。

アウトプットは2量子ビットで|00⟩、|11⟩となっているからベル状態であることを示している。
アダマールとCNOTのゲートの組み合わせでベル状態をつくることができるので、該当する選択肢を選べばよい。

QuatumCircuitのメソッドinitializeはレジスタと状態ベクトルなどを引数として渡し、指示した量子レジスタを初期化することができる。

選択肢Aは、ベル状態である状態ベクトルを量子レジスタ0と1に渡しているので正解である。

選択肢Bはアダマールとの組み合わせでベル状態をつくっており、正解である。

サンプル問題8の解説

選択肢Cは選択肢Aと似ているが、初期化する値が異なりこれではベル状態とはならないので、不正解。

選択肢Dはアダマールがなく、CNOTゲートのみ設定されており、これではベル状態は作られないため不正解である。

選択肢Eは量子ビット0，1にそれぞれアダマールゲートを置いているが、CNOTゲートがない。このためベル状態は作られないので、不正解である。

従って正解はAとBである。

サンプル問題9の解説

問 題

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)`

解 説

CNOT以外で複数量子ゲートをつくるコードを選択肢の中から選ぶ問題である。

選択肢の中にはCNOTゲートおよび同等のものがあるが、それを除いた複数量子ゲートを選べという意味である。

選択肢AはCXゲートはCNOTゲートの別名である。
選択肢BはCNOTゲートである。
選択肢CはMulti-CX gateである。CNOTの制御ビットが複数あるもので、CNOTの多量子版である。

選択肢A,B,CはCNOTゲートと同じ仲間と考えられるので、これらは答えとして適切ではない。

選択肢DはControl-Zゲートであり、複数量子ゲートであるが、CNOTではない。従って選択肢Dが正解である。

サンプル問題10の解説

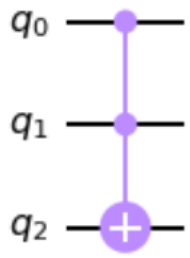
問 題

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)
```

解 説

Toffoliゲート以外に複数量子ゲートをつくるコードを選択肢の中から選ぶ問題である。
Toffoliゲートと同等のゲートを選ばないようにする。



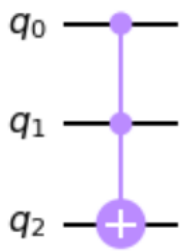
選択肢AはToffoliゲートである。

選択肢Bは multi-CXgateであり、量子ビット0，1を制御ビットとし、量子ビット2をターゲットとしてXゲートを作用させているので、Toffoli ゲートである。

選択肢CはCXgateで新たなCXゲートをつくるものだが、メソッドでcontrolを指定すると複数の制御ビットをもつCXゲートを作ることができる。
実際に確かめてみると

In [8]:
```
from qiskit.circuit.library import CXGate
#
qc= QuantumCircuit(3)
ccx = CXGate().control()
qc.append(ccx, [0,1,2])
qc.draw('mpl')
```

Out[8]:



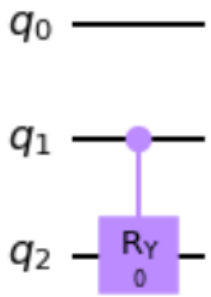
サンプル問題10の解説

となり、Toffoliゲートであることがわかる。

選択肢DはCRYゲート（Controlled-RY gate）を作るものである。
引数は、Y軸周りの回転角theta、制御ビット、ターゲットである。選択肢では量子ビットの番号のようにも見えるが、最初のゼロは回転角ゼロ（ラジアン）を意味している。量子ビット1を制御ビット、量子ビット2をターゲットとしてY軸のまわりに回転角ゼロを作用させるゲートとなっている。これは複数量子ゲートであるが、Toffoliゲートではないので正解である。従って正解はDである。

In [13]:
```
qc= QuantumCircuit(3)
qc.cry(0,1,2)
qc.draw('mpl')
```

Out[13]:



サンプル問題11の解説

問 題

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()`

解 説

問題は、選択肢の中から、量子回路qcの全ての量子ビットにバリアを設定することができる2つの選択肢を選べというものである。

量子回路にバリアを設定する場合、
QuantumCircuit.barrier(*qargs) として設定する。

普通は引数のqargsで設定するレジスタを列挙するが、qargsが空の場合には、Qiskitのドキュメントにあるとおり、回路の全ての量子ビットにバリアが設定される。

従って正解は選択肢BとCである。

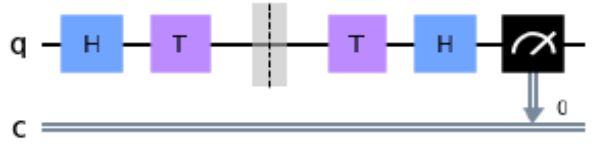
参 考

https://qiskit.org/documentation/locale/ja_JP/stubs/qiskit.circuit.QuantumCircuit.barrier.html

サンプル問題12の解説

問 題

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)
```

解 説

左記の量子回路のバリアを外した場合に等価となるコードを選択肢の中から選ぶ問題である。

バリアがあると、その両側に存在するゲート等はそれぞれ個別に実行される。バリアがない場合、隣接するゲートどうしがキャンセル、合成できる場合には、等価なものに置き換えられ、depthが浅くなるように最適化される。

ここではバリアの両側にTゲートがあるため、バリアを外すことによって、TTと等価なものに置き換わることになる。

Tゲートは $\phi = \pi/4$ のPゲートである:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$

サンプル問題12の解説

従って、TTは $\phi = \pi/2$ のPゲートに等しい。
すなわちSゲートである。

従って、

$$TT = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{2}} \end{bmatrix} = S$$

バリアを取り去ったあとの等価な回路は選択肢の中からSゲートで記述しているコードをみつければよい。

まず、選択肢Aが正解である。

選択肢Bは、HSHが単位ゲートIに等しくなるというように表現されているがHSH≠Iであるため正しくない。

選択肢CはTとTダガーに分かれて書かれているが、

$$TT^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

となり、こちらもSに等しくないため正解ではない。

選択肢DはTTがZゲートに等しくなるように書かれているが、上記の式からTT≠Zとなるため、正しくない。

従って正解は選択肢Aである。

サンプル問題13の解説

問 題

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

解 説

上記のコードから作られる量子回路のdepth（深さ）を求める問題である。

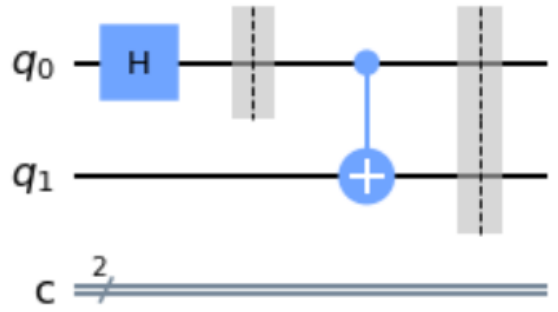
Barrierを除くと、アダマールとCXゲートからなる2量子ビットの回路になっている。

回路のdepthとは回路のインプットからアウトプットまでのクリティカルパスである。Barrierのようにコンパイルやシミュレータへの指示となるものはパスの数にカウントされないので、depthは2である。
従って正解は選択肢Aである。

In [2]:
```
qc = QuantumCircuit(2,2)
qc.h(0)
qc.barrier(0)
qc.cx(0,1)
qc.barrier([0,1])

qc.draw('mpl')
```

Out[2]:



In [3]:
```
qc.depth()
```

Out[3]: 2

サンプル問題14の解説

問 題

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)
```

解 説

左記1）～3）のパラメータを設定して量子回路を実行（execute）する場合、正しいコードを選択する問題である。

shotsを指定する場合、パラメータ1）からshots =1024を指定していないものは正解ではない。（shotsの記述が全くない場合、デフォルトは1024であるため、選択肢にそういうものがあれば正解の可能性がある）
このため、選択肢Aは正解の可能性がある。選択肢Bはloop、選択肢Cはrepeat、選択肢Dはshotとしているため不正解である。

2）からQASM　Simulatorを指定していないものは正解ではない。選択肢Bは不正解である。

3）は量子ビットの結合が直列になっていると条件であり、coupling_mapで指定されるマップの結合状況を見ると選択肢A～Dまでの記述は一緒である。

サンプル問題14の解説

このため、3）の条件だけでは不正解の選択肢を見つけることはできない。

従ってAが正解である。

参 考

https://qiskit.org/documentation/locale/ja_JP/apidoc/execute.html?highlight=execute

サンプル問題15の解説

問 題

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")`

解 説

指定した結合のしかたによって結びついている3量子ビットのセット上の量子回路を実行する場合に、#insert code here のところに当てはまるものを選択肢から選ぶ問題である。

このような場合には、executeの引数でcoupling_mapを使い、予め指定した量子ビットの結合状態の情報を渡すようにする。

選択肢C,Dについては、modeという引数が記されているが、executeにはmodeという指定はない。また、いずれも結合の情報を示す記述もない。このためC,Dは不正解。

選択肢Bは、custom_topologyと記されているが、正しくはcoupling_mapであるため、不正解。

従って、coupling_mapとその具体的な結合の情報が記されている選択肢Aが正解である。

参 考

https://qiskit.org/documentation/locale/ja_JP/apidoc/execute.html

サンプル問題16の解説

問 題

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

解 説

BasicAerで利用できる3つのsimulatorを選ぶ問題である。

BasicAerはpythonベースのシミュレータとなっており、
qasm_simulator
statevector simulator
unitary simulator
の3つからなる。

このため選択肢の中から、この3つを選べばよい。
従って正解は、A, C, Dである。

参 考

https://qiskit.org/documentation/locale/ja_JP/apidoc/providers_basicaer.html

サンプル問題17の解説

問 題

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')`

解 説

Statevector simulator のオブジェクトを変数backendにアサインするのに正しいコードを選択肢から選ぶ問題である。

いずれの選択肢もBasicAerを使うことが伺える。この場合、Backendをアサインするには、BasicAer.get_backendを使い、statevector_simulatorを指定するのが正しい。従って正解は選択肢Bである。

選択肢A、CはStatevectorSimulatorPy()の記載があるが不正解。
選択肢Dはget_backとなっており、こちらも正しくない。

参 考

https://qiskit.org/documentation/locale/ja_JP/apidoc/providers_basicaer.html

サンプル問題18の解説

問 題

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]])
```

解 説

1量子ビットのXゲートを表すオペレータを定義しているものを選択肢から選ぶ問題である。

まず、選択肢Cは1量子ビットのXゲートからオペレータを定義しているので正解である。
Qiskitではオペレータはさまざまな定義のしかたがある。

選択肢Aは不正解。Xopというゲートはない。もしXゲートで定義するのであれば
　op = Operator(Xgate())
とする。

選択肢Bはオペレータをベクトル[0,1]で定義しているが、これはXゲートではないため不正解。

選択肢Dでは、オペレータを2×2行列で定義している。[[1,0,0,1]]は2×2の単位行列を表しているので、Iゲートであり、Xゲートではないため不正解。
Xゲートとするのであれば、
　op = Operator([[0,1,1,0]])
とする。

参 考

https://qiskit.org/documentation/locale/ja_JP/tutorials/circuits_advanced/02_operators_overview.html

サンプル問題19の解説

問 題

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

解 説

2つのオペレータop_a, op_bのfidelity（忠実度）をとった結果についてどのようになるのかという問題である。
この2つのオペレータはグローバル位相の差しかない。

fidelityを調べる場合、状態ベクトル同士のfidelityを調べるにはstate_fidelityを使い、オペレータどうしのfidelityを調べるには、process_fidelity および average_gate_fidelityを使う。

オペレータ同士が直交する場合には、fidelityはゼロ、等価な場合は1となる。

この場合、オペレータどうしのfidelityを調べるのであるから、process_fidelity および　average_gate_fidelityを使うため、選択肢からこの組み合わせで記述しているものを選ぶ。
選択肢A,B,Dは状態ベクトルを調べるstate_fidelityの記述が含まれているため、不正解である。

サンプル問題19の解説

選択肢Cはprocess_fidelity および　average_gate_fidelity で記しており、fidelityはいずれも1であるとしている。

2つのオペレータはグローバル位相の差しかないため、fidelityは1であり、この記述が正しい。従って正解は選択肢Cである。

参 考

https://qiskit.org/documentation/locale/ja_JP/stubs/qiskit.quantum_info.process_fidelity.html

https://qiskit.org/documentation/locale/ja_JP/stubs/qiskit.quantum_info.average_gate_fidelity.htm

サンプル問題20の解説

問 題

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}

解 説

上記のコードにより、測定される確率分布について最も近い選択肢を選ぶ問題である。

この量子回路は、2量子ビットと2古典ビットからなる回路で、量子ビット0にXゲートがある。

量子ビット1にはゲートはなく、量子ビット0,1とも測定をして結果を古典ビット0,1にそれぞれ書く処理になっている。Qasm simulator による1000回の試行をおこなった結果をカウントするようになっている。

この量子回路を実行すると、量子ビット0は初期値|0〉はXゲートにより|1〉となる。量子ビット1は初期値|0〉のままである。
ここでは量子ビットの並びは、Qiskitの量子ビットの位取りに従う。(※)量子ビットの状態としては理想的には|01〉のみが計測される。
Qasmはノイズ等により、1000回の試行で、|01>が必ずしも1000回測定されるわけではないが、選択肢の中からもっとも近いものをえらぶとすれば、選択肢Bを選ぶのが正しい。

参 考

Qiskitでの量子ビットの位取りについては
https://qiskit.org/documentation/locale/ja_JP/tutorials/circuits/3_summary_of_quantum_operations.html
の「Qiskitにおける基底ベクトル表記の順番」を参照。