Microsoft

Programming assignment 2
Unitary transformations & measurements

Mariia Mykhailova
Senior Software Engineer
Microsoft Quantum Systems

General notes

- Make sure your code compiles!
  - In the next assignments I'll ignore tasks that don't compile
- Q# does not have implicit type casting
- The second measurement on already measured qubit gives the same result as the first one
- Use DumpMachine for debugging measurement tasks
  - Prep a random input state
  - Run your solution on it
  - Use DumpMachine right before measurement: should be a basis state
- Explore testing harnesses from the katas to learn

Task 1.1: Relative phase

Implement a 2-qubit gate that has the following effect:

$$|00\rangle \to |00\rangle, |01\rangle \to -|01\rangle, |10\rangle \to |10\rangle, |11\rangle \to |11\rangle$$

- Adding -1 phase to state $|01\rangle$ is the same as applying Z gate to 2nd qubit if 1st qubit is in $|0\rangle$ state

```
(ControlledOnBitString([false], Z))([qs[0]], qs[1]);
```

- …or applying Z gate to the 2nd qubit if 1st qubit is in $|1\rangle$ state and apply Z gate to the 2nd qubit always

```
Z(qs[1]);
Controlled Z(qs[0..0], qs[1]);
```

Task 1.2: Peres gate

Implement a 2-qubit gate that has the following effect:

- Third qubit state is flipped if 1st and 2nd qubits are in $|11\rangle$ state
- Second qubit state is flipped if 1st qubit is in $|1\rangle$ state
- First qubit state doesn't change

```
CCNOT(qs[0], qs[1], qs[2]);
CNOT(qs[0], qs[1]);
```

Input			Output		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Task 1.3: Deutsch gate

$$|a,b,c\rangle = \begin{cases} i\cos\theta\,|a,b,c\rangle + \sin\theta\,|a,b,1-c\rangle & \text{if } ab = 1 \\ |a,b,c\rangle & \text{otherwise} \end{cases}$$

- Same as controlled transformation iR_x: $\begin{pmatrix} i\cos\theta & \sin\theta \\ \sin\theta & i\cos\theta \end{pmatrix}$
- Two gates: controlled R_x and controlled phase application
- You can think of phase application as acting on 1st and 2nd qubits (add i phase if they are in $|11\rangle$ state) – controlled S

```
(Controlled S)(qs[0..0], qs[1]);
(Controlled Rx)(qs[0..1], (2.0 * theta, qs[2]));
```

Task 1.4: Clone $|+\rangle$ and $|-\rangle$ states

- No-cloning theorem prohibits cloning *non-orthogonal* states
- We've seen that CNOT allows to clone orthogonal $|0\rangle$ and $|1\rangle$:

$$\mathrm{CNOT}|00\rangle = |00\rangle, \mathrm{CNOT}|10\rangle = |11\rangle$$

- Let's convert $|+\rangle$ to $|0\rangle$ and $|-\rangle$ to $|1\rangle$, clone those states and convert cloning results back to $|+\rangle$ and $|-\rangle$

```
H(data);
CNOT(data, scratch);
H(data);
H(scratch);
```

Task 2.1: $|0\rangle$ or $|1\rangle$?

Return `true` if the qubit was in $|0\rangle$ or `false` if it was in $|1\rangle$

- Measurements kata, task 1.1, but with the opposite return
  - "Consolation" problem
- Operation `M` measures the qubit in computational basis and returns constants of type Result `Zero` or `One`

```
return M(q) == Zero;
```

Task 2.2: Distinguish two 3-qubit states

$|\psi_0\rangle = \frac{1}{2}(|000\rangle + |001\rangle + |010\rangle + |100\rangle)$ (return 0) vs

$|\psi_1\rangle = \frac{1}{2}(|111\rangle + |110\rangle + |101\rangle + |011\rangle)$ (return 1)

- If we measure each qubit, we'll get one of the basis states
- All basis states in $|\psi_0\rangle$ have 0 or 1 "1"s in them, in $|\psi_1\rangle$ - 2 or 3

```
mutable nOne = 0;
for (i in 0..2) {
    set nOne += M(qs[i]) == One ? 1 | 0;
}
return nOne <= 1 ? 0 | 1;
```

Task 2.3: Distinguish four 3-qubit states

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \qquad |\psi_1\rangle = \frac{1}{\sqrt{2}}(|001\rangle + |110\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|010\rangle + |101\rangle) \qquad |\psi_3\rangle = \frac{1}{\sqrt{2}}(|100\rangle + |011\rangle)$$

- Each state has varying number of 1s in it, so can't just count 1s
- But all basis states are different! So we can identify basis state (could have done it in previous task as well)

```
let res = MultiM(qs);     // measure each qubit
if (res[0] == res[1] and res[1] == res[2]) { return 0; }
if (res[0] == res[1]) { return 1; }
if (res[0] == res[2]) { return 2; }
return 3;
```

Task 2.4: Distinguish four 2-qubit states

$$|\psi_0\rangle = \tfrac{1}{2}(|00\rangle + i|01\rangle + i|10\rangle - |11\rangle) = \tfrac{1}{2}(|0\rangle + i|1\rangle) \otimes (|0\rangle + i|1\rangle) = SH|0\rangle \otimes SH|0\rangle$$
$$|\psi_1\rangle = \tfrac{1}{2}(|00\rangle - i|01\rangle + i|10\rangle + |11\rangle) = \tfrac{1}{2}(|0\rangle + i|1\rangle) \otimes (|0\rangle - i|1\rangle) = SH|0\rangle \otimes SH|1\rangle$$
$$|\psi_2\rangle = \tfrac{1}{2}(|00\rangle + i|01\rangle - i|10\rangle + |11\rangle) = \tfrac{1}{2}(|0\rangle - i|1\rangle) \otimes (|0\rangle + i|1\rangle) = SH|1\rangle \otimes SH|0\rangle$$
$$|\psi_3\rangle = \tfrac{1}{2}(|00\rangle - i|01\rangle - i|10\rangle - |11\rangle) = \tfrac{1}{2}(|0\rangle - i|1\rangle) \otimes (|0\rangle - i|1\rangle) = SH|1\rangle \otimes SH|1\rangle$$

- Apply adjoint SH to each qubit to bring the states to basis states
- Measure basis states (pay attention to the order of the bits!)

```
ApplyToEach(Adjoint S, qs);
ApplyToEach(H, qs);
let m1 = M(qs[0]) == One ? 1 | 0;
let m2 = M(qs[1]) == One ? 1 | 0;
return m1 * 2 + m2;
```

Task 2.5: Estimate amplitudes

Given 21 qubits, each in state $\alpha|0\rangle + \beta|1\rangle$, estimate α and β.

- The probabilities of measuring 0 and 1 are $|\alpha|^2$ and $|\beta|^2$.
- To estimate the probabilities, measure each qubit, count 0s and 1s in the measurement results and divide by 21

```
mutable nZero = 0;
for (q in qs) {
    set nZero += M(q) == Zero ? 1 | 0;
}
let α = Sqrt(IntAsDouble(nZero) / IntAsDouble(Length(qs)));
let β = Sqrt(1.0 - α * α);
return (α, β);
```