Challenge statement

Welcome to the Femto Forest, a woodland where you can witness quantum phenomena while enjoying nature! For travellers who like to spend time outdoors, camping is the way to go. But there are many other indoorsy options as well. On the northwestern edge of the Femto Forest is Camp Iota, which has two areas. Area A has an amazing view of the mountains in the Tensor Tundra, but it's only a quarter of the size of Area B, which doesn't have that view.

At Camp Iota, the camp managers take advantage of the true quantum randomness to handle how to distribute the campsites; that way campers don't fight over who gets the best view. They do this through an app in their Quantum Smartphone, which prepares two states in a uniform superposition, measures them, and then passes the resulting states through a Quantum OR gate.

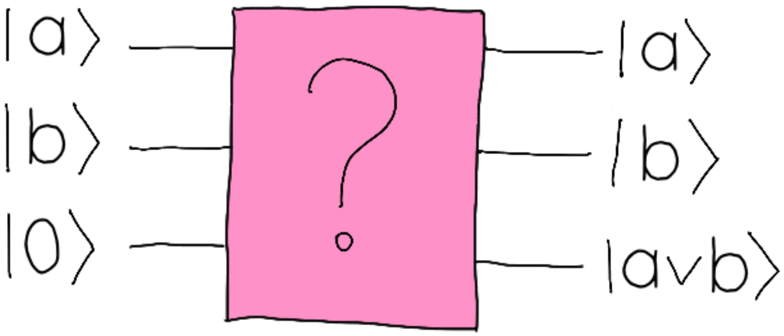
As you enter the forest, one of the Camp Iota managers notices all the QHack stickers on your suitcase. *"If you manage to tell me how the Smartphone algorithm works"* she claims with a daring grin, *"I'll let you into area A without having to test your luck"*. Excited by this unique offer, you start mentally reviewing how logic gates work with quantum computers...

How does a Quantum OR gate work? Classical computers ultimately work by applying *reversible* logical operations to bits of information. One of such such operations is the OR gate denoted by $\vee$, which acts on two bits as follows:

$$0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1.$$

The camp managers then send the campers to area A when the OR gate yields the result 0, which only happens with probability $1/4$.

We would like our quantum computers to be able to perform this basic logical operator in a reversible way. To do this, we need a circuit with three qubits. The first two qubits $|a\rangle$ and $|b\rangle$ ($a, b \in \{0, 1\}$) encode the bits on which the OR gate acts. The third one starts in the state $|0\rangle$ and ends up in $|a \vee b\rangle$. The action of the circuit is shown in the picture below.



It is possible to create this operation using only the Pauli X (`qml.PauliX`) and Toffoli (`qml.Toffoli`) gates. Your goal in this challenge is precisely to create a quantum circuit that implements an OR gate using only these gates.

Challenge Code

In the challenge template, you must complete the `or_circuit` QNode which, given an array (`np.array(int)`) of the form $[a, b, 0]$ representing the initial state $|a\rangle|b\rangle|0\rangle$, returns the state (`np.tensor`) $|a\rangle|b\rangle|a \vee b\rangle$. You may only use the Pauli X (`qml.PauliX`) and Toffoli (`qml.Toffoli`) gates.

Input

As an input to this challenge, you are given an array $[a, b, 0]$ (`np.array(int)`) with $a, b \in \{0, 1\}$, representing the initial state $|a\rangle|b\rangle|0\rangle$.

Output

The output, as displayed in the test cases below, is an `np.tensor` $[a, b, a \vee b]$ representing the state $|a\rangle|b\rangle|a \vee b\rangle$. The `qml.state()` output of `or_circuit` will be processed by the test functions to show the state in this form.

Test cases

The following **public test cases** are available for you to check your work. There are also some **hidden test cases** which we will use to check that your solution works in full generality.

```
test_input: [0,0,0]
expected_ouput: [0,0,0]

test_input: [1,0,0]
expected_ouput: [1,0,1]

test_input: [0,1,0]
expected_ouput: [0,1,1]

test_input: [1,1,0]
expected_ouput: [1,1,1]
```

Your state **may not** differ from the expected output by a global phase. The grader will compare the states element-wise to verify your answer.

Good luck!