```python
In [1]:   from qiskit import *
          import matplotlib.pyplot as plt
          import numpy as np
```

```python
In [2]:   oracle = QuantumCircuit(2, name='Oracle')
          oracle.cz(0,1)
          oracle.to_gate()
          oracle.draw()
```

```
Out[2]:  q_0: ──■──
                 │
         q_1: ──■──
```

```python
In [5]:   backend = Aer.get_backend('statevector_simulator')
          grover_circ=QuantumCircuit(2,2)
          grover_circ.h([0,1])
          grover_circ.append(oracle,[0,1])
          grover_circ.draw()
```

```
Out[5]:  q_0: ┤ H ├┤0        ├
                        Oracle
         q_1: ┤ H ├┤1        ├

         c: 2/══════════════════
```

```python
In [7]:   reflection = QuantumCircuit(2, name="Reflection")
          reflection.h([0,1])
          reflection.z([0,1])
          reflection.cz(0,1)
          reflection.h([0,1])
          reflection.to_gate()
```

```
Out[7]:  Instruction(name='Reflection', num_qubits=2, num_clbits=0, params=[])
```

```python
In [9]:   reflection.draw()
```

```
Out[9]:  q_0: ┤ H ├┤ Z ├──■──┤ H ├
         q_1: ┤ H ├┤ Z ├──■──┤ H ├
```

```python
In [8]:   backend = Aer.get_backend('statevector_simulator')
          grover_circ.append(reflection, [0,1])
          grover_circ.draw()
```

```
Out[8]:  q_0: ┤ H ├┤0        ├┤0           ├
                        Oracle       Reflection
         q_1: ┤ H ├┤1        ├┤1           ├

         c: 2/═══════════════════════════════
```

```python
In [11]:  grover_circ.measure([0,1],[0,1])
```

```
grover_circ.draw()
```

Out[11]:

```
q_0: ─┤ H ├┤0        ├┤0          ├┤M├───
      ├───┤│  Oracle ││ Reflection│└╥┘┌─┐
q_1: ─┤ H ├┤1        ├┤1          ├─╫─┤M├
      └───┘└─────────┘└───────────┘ ║ └╥┘
c: 2/═══════════════════════════════╩══╩═
                                    0   1
```
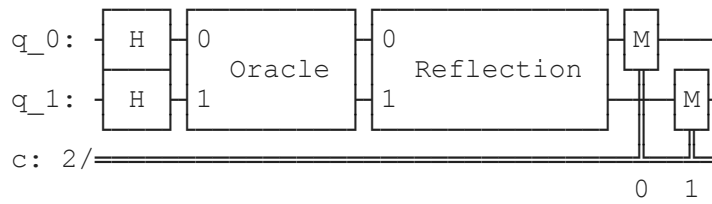
In [13]:
```
job = execute(grover_circ, backend, shots=1)
result= job.result()
result.get_counts()
```

Out[13]: {'11': 1}

In [ ]: