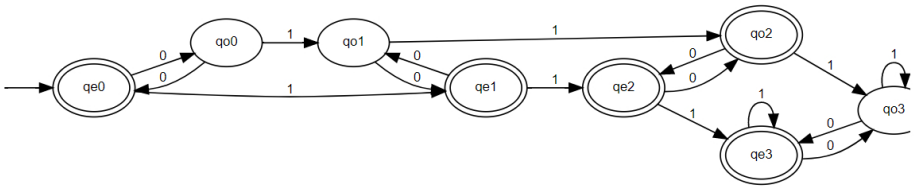
In [6]:
```
01 110101 1000010 11110000 10111011 001111 010111 01010000 11 00100001 10111000 100 10000100 1011110 00010001 11111001 10000 1000
```

Enter a DFA for the language $\{ w \in \{0,1\}^* \mid \#_0(w) \bmod 2 = 0 \vee \#_1(w) = 2 \}$, i.e. the language of strings of 0's and 1's with and even number of 0's or exactly two 1's.

In [7]:
```
dfa = '''
input_symbols 0 1
states qe0 qe1 qe2 qe3 qo0 qo1 qo2 qo3
initial qe0
final qe0 qe1 qe2 qe3 qo2
qe0 qo0 0
qe0 qe1 1
qo0 qe0 0
qo0 qo1 1

qe1 qo1 0
qe1 qe2 1
qo1 qe1 0
qo1 qo2 1

qe2 qo2 0
qe2 qe3 1
qo2 qe2 0
qo2 qo3 1

qe3 qo3 0
qe3 qe3 1
qo3 qe3 0
qo3 qo3 1

'''
```

In [8]:
```
show(dfa)
```

Out[8]:



In [9]:
```
check_answer(dfa)
```

OK

If the DFA above is not OK, inspect its execution for a selected word, e.g. 1010.

In [10]:
```
simulate_dfa(dfa,'1010')
```

State	Word
qe0	1010
qe1	101
qo1	10
qo2	1
qe2	

In [ ]:
```

```