In [1]:

```
import numpy as np
```

In [2]:

```
def predict(activation):
    if activation >=0:
        return 1
    else:
        return 0
```

In [3]:

```
def perceptron(x, w, b):
    activation = np.dot(w, x) + b
    y = predict(activation)
    return y
```

In [4]:

```
def NOT(x):
    w = -1
    b = 0.5
    return perceptron(x, w, b)
```

In [5]:

```
def AND(x):
    w = np.array([1, 1])
    b = -1.5
    return perceptron(x, w, b)
```

In [6]:

```
def OR(x):
    w = np.array([1, 1])
    b = -0.5
    return perceptron(x, w, b)
```

In [7]:

```
def XOR(x):
    A = NOT(AND(x))
    B = OR(x)
    X = np.array([A, B])
    y = AND(X)
    return y
```

In [8]:

```
def main():
    examples = [[1,1],[1,0],[0,1],[0,0]]
    print("\n##### Output #####\n")
    for example in examples:
        x = np.array(example)
        print(f"\nNOT({example[0]}): {NOT(example[0])}")
        print(f"AND({example}): {AND(x)}")
        print(f"OR({example}): {OR(x)}")
        print(f"XOR({example}): {XOR(x)}")
```

In [9]:

```
main()
```

```
##### Output #####


NOT(1): 0
AND([1, 1]): 1
OR([1, 1]): 1
XOR([1, 1]): 0

NOT(1): 0
AND([1, 0]): 0
OR([1, 0]): 1
XOR([1, 0]): 1

NOT(0): 1
AND([0, 1]): 0
OR([0, 1]): 1
XOR([0, 1]): 1

NOT(0): 1
AND([0, 0]): 0
OR([0, 0]): 0
XOR([0, 0]): 0
```

In [ ]: