CODE:

In [1]:

```
import numpy as np
class Perceptron:
    def __init__(self, learning_rate=0.1, epochs=100):
        self.learning_rate = learning_rate
        self.epochs = epochs

    def fit(self, X, y):
        self.weights = np.zeros(X.shape[1])
        self.bias = 0

        for epoch in range(self.epochs):
            for i in range(X.shape[0]):
                y_pred = self.predict(X[i])
                self.weights += self.learning_rate * (y[i] - y_pred) * X[i]
                self.bias += self.learning_rate * (y[i] - y_pred)

    def predict(self, X):
        linear_output = np.dot(X, self.weights) + self.bias
        if linear_output >= 0:
            return 1
        else:
            return 0
```

In [2]:

```
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([0, 1, 1, 1])

perceptron = Perceptron()
perceptron.fit(X, y)
test_input = np.array([1, 0])
print(perceptron.predict(test_input))
```

```
1
```

In [ ]:

```

```

RESULT:

Hence, we successfully implemented Single Layer Perceptron for Classification Problem.