

```

def activation(out, b):
    if out >= b:
        return 1
    else:
        return 0

def perceptron(and_inp):
    x1=[0,0,1,1]
    x2=[0,1,0,1]
    y=[0,0,0,1]
    w=[0,0]
    b = 1
    alpha = 0.5
    i = 0
    print("Train Perceptron: ")
    while i < 4:
        summ = x1[i]*w[0] + x2[i]*w[1]
        y_pred = activation(summ, b)
        print("INPUT : " + str(x1[i]) + " , " + str(x2[i]))
        print("WEIGHTS : " + str(w[0]) + " , " + str(w[1]))
        print("Y_PRED : " + str(y_pred) + " , " + " Y_ACTUAL : " + str(y[i]))

        if (y_pred != y[i]):
            print("Updating Weights .....")
            w[0] = w[0] +alpha*(y[i]*x1[i])
            w[1] = w[1] + alpha *(y[i]*x2[i])
            print("Updated WEIGHTS : " + str(w[0]) + " , " + str(w[1]))
            i = i+1
        summ = and_inp[0]*w[0] + and_inp[1]*w[1]
        return activation(summ, b)

and_inp = [0,1]
print("AND Gate output for "+str(and_inp)+ " : " + str(perceptron(and_inp)))

```

```

☞ Train Perceptron:
INPUT : 0 , 0
WEIGHTS : 0 , 0
Y_PRED : 0 , Y_ACTUAL : 0
INPUT : 0 , 1
WEIGHTS : 0 , 0
Y_PRED : 0 , Y_ACTUAL : 0
INPUT : 1 , 0
WEIGHTS : 0 , 0
Y_PRED : 0 , Y_ACTUAL : 0
INPUT : 1 , 1
WEIGHTS : 0 , 0
Y_PRED : 0 , Y_ACTUAL : 1
Updating Weights .....
Updated WEIGHTS : 0.5 , 0.5
AND Gate output for [0, 1] : 0

```