===================================================================  
 Exp No: 1 **Logic Function using McCulloch-Pitts Neuron**

**====================================================================  
  
Aim…**

To implement AND, OR, NOT, NOR, and XOR logic gates using the McCulloch-Pitts neuron model.

**Code…**

**def mcp\_neuron(inputs, weights, threshold):**

**summation = sum(i \* w for i, w in zip(inputs, weights))**

**return 1 if summation >= threshold else 0**

**def AND(x1, x2):**

**return mcp\_neuron([x1, x2], [1, 1], 2)**

**def OR(x1, x2):**

**return mcp\_neuron([x1, x2], [1, 1], 1)**

**def NOT(x1):**

**return mcp\_neuron([x1], [-1], 0)**

**def NOR(x1, x2):**

**return mcp\_neuron([x1, x2], [-1, -1], 0)**

**def XOR(x1, x2):**

**return (x1 ^ x2)**

**print("AND")**

**for x in [(0,0), (0,1), (1,0), (1,1)]:**

**print(f"{x} -> {AND(\*x)}")**

**print("\nOR")**

**for x in [(0,0), (0,1), (1,0), (1,1)]:**

**print(f"{x} -> {OR(\*x)}")**

**print("\nNOT")**

**for x in [0, 1]:**

**print(f"{x} -> {NOT(x)}")**

**print("\nNOR")**

**for x in [(0,0), (0,1), (1,0), (1,1)]:**

**print(f"{x} -> {NOR(\*x)}")**

**print("\nXOR")**

**for x in [(0,0), (0,1), (1,0), (1,1)]:**

**print(f"{x} -> {XOR(\*x)}")**

**Output…**

==== RESTART: C:\Users\tgane\OneDrive\Desktop\5th sem\dlt\exp 1\EXP no 1.py ====

AND

(0, 0) -> 0

(0, 1) -> 0

(1, 0) -> 0

(1, 1) -> 1

OR

(0, 0) -> 0

(0, 1) -> 1

(1, 0) -> 1

(1, 1) -> 1

NOT

0 -> 1

1 -> 0

NOR

(0, 0) -> 1

(0, 1) -> 0

(1, 0) -> 0

(1, 1) -> 0

XOR

(0, 0) -> 0

(0, 1) -> 1

(1, 0) -> 1

(1, 1) -> 0