**Assignment No.: 02**

**Generate ANDNOT function using McCulloch-Pitts neural net by a python program.**

**CODE:**

```
class McCullochPittsNeuron:
    def __init__(self, weights, threshold):
        self.weights = weights
        self.threshold = threshold

    def activate(self, inputs):
        net_input = sum(w * x for w, x in zip(self.weights, inputs))
        return 1 if net_input > self.threshold else 0

def ANDNOT(x1, x2, weights):
    # Create a McCulloch-Pitts neuron for ANDNOT
    threshold = 0
    andnot_neuron = McCullochPittsNeuron(weights, threshold)

    # Activate the neuron with the input values
    return andnot_neuron.activate([x1, x2])

# Take user input for weights
try:
    weight_x1 = int(input("Enter weight for x1: "))
    weight_x2 = int(input("Enter weight for x2: "))
```

```
except ValueError:
    print("Invalid input. Please enter numeric values for weights.")
    exit()

# Call the ANDNOT function with user-provided weights
weights = [weight_x1, weight_x2]

result = ANDNOT(1, 0, weights)
print(f"ANDNOT(1, 0) with weights {weights}: {result}")

result = ANDNOT(0, 1, weights)
print(f"ANDNOT(0, 1) with weights {weights}: {result}")

result = ANDNOT(0, 0, weights)
print(f"ANDNOT(0, 0) with weights {weights}: {result}")

result = ANDNOT(1, 1, weights)
print(f"ANDNOT(1, 1) with weights {weights}: {result}")
```

**OUTPUT:**

```
PS D:\ANN Practicals> & C:/Users/user/AppData/Local/P
d:/ANN Practicals/pract2.py"
Enter weight for x1: 1
Enter weight for x2: -1
ANDNOT(1, 0) with weights [1, -1]: 1
ANDNOT(0, 1) with weights [1, -1]: 0
ANDNOT(0, 0) with weights [1, -1]: 0
ANDNOT(1, 1) with weights [1, -1]: 0
PS D:\ANN Practicals> & C:/Users/user/AppData/Local/P
d:/ANN Practicals/pract2.py"
Enter weight for x1: 1
Enter weight for x2: 0
ANDNOT(1, 0) with weights [1, 0]: 1
ANDNOT(0, 1) with weights [1, 0]: 0
ANDNOT(0, 0) with weights [1, 0]: 0
ANDNOT(1, 1) with weights [1, 0]: 1
PS D:\ANN Practicals>
```