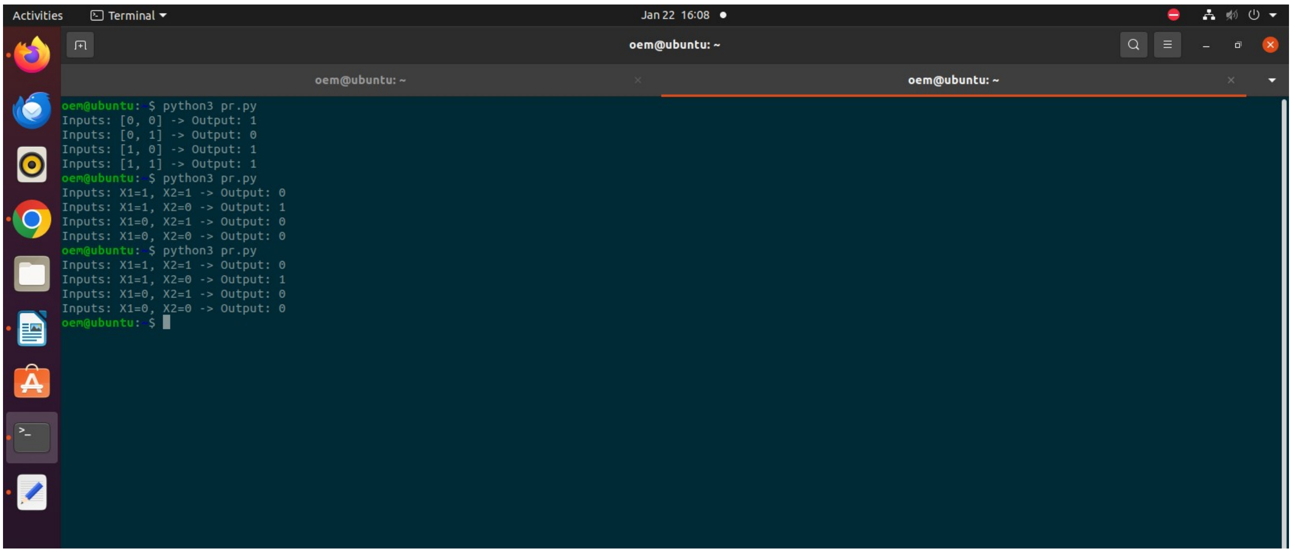
Practical 2:

```
import numpy as np
# McCulloch-Pitts Neuron Model
def mcCulloch_pitts_neuron(X1, X2, W1, W2, threshold):
    # Calculate the weighted sum
    Yin = W1 * X1 + W2 * X2
    # Apply the threshold activation function
    if Yin >= threshold:
        return 1
    else:
        return 0
# Weights and threshold
W1 = 1
W2 = -1
threshold = 1
# Test all possible combinations of inputs X1 and X2
inputs = [(1, 1), (1, 0), (0, 1), (0, 0)]
# Output the result for each input combination
for X1, X2 in inputs:
    output = mcCulloch_pitts_neuron(X1, X2, W1, W2, threshold)
    print(f"Inputs: X1={X1}, X2={X2} -> Output: {output}")
```

Output:



```
oem@ubuntu:~$ python3 pr.py
Inputs: [0, 0] -> Output: 1
Inputs: [0, 1] -> Output: 0
Inputs: [1, 0] -> Output: 1
Inputs: [1, 1] -> Output: 1
oem@ubuntu:~$ python3 pr.py
Inputs: X1=1, X2=1 -> Output: 0
Inputs: X1=1, X2=0 -> Output: 1
Inputs: X1=0, X2=1 -> Output: 0
Inputs: X1=0, X2=0 -> Output: 0
oem@ubuntu:~$ python3 pr.py
Inputs: X1=1, X2=1 -> Output: 0
Inputs: X1=1, X2=0 -> Output: 1
Inputs: X1=0, X2=1 -> Output: 0
Inputs: X1=0, X2=0 -> Output: 0
oem@ubuntu:~$
```