

```
In [1]: #E31 Write a Program to implement Perceptron  
#1 Import Libraries  
import numpy as np
```

```
In [2]: #Inputs
```

```
In [3]: x=np.array([0,0])
```

```
In [4]: #Weights
```

```
In [5]: w=np.array([1,1])
```

```
In [6]: #Bias
```

```
In [7]: b=-1.5
```

```
In [8]: #Net input
```

```
In [9]: yin=np.dot(w,x)+b
```

```
In [10]: yin
```

```
Out[10]: -1.5
```

```
In [11]: #Activation Unit Step
```

```
In [12]: if(yin>=0):  
         print("1")  
         else:  
         print("0")
```

```
0
```

```
In [13]: #User defined function with binary step as activation
```

```
In [14]: def PercAND(x1,x2,w1,w2,b):  
         x=np.array([x1,x2])  
         w=np.array([w1,w2])  
         yin=np.dot(w,x)+b  
         if(yin>=0):  
             return 1  
         else:  
             return 0
```

```
In [15]: PercAND(0,0,1,1,-1.5)
```

```
Out[15]: 0
```

```
In [16]: PercAND(0,1,1,1,-1.5)
```

```
Out[16]: 0
```

```
In [17]: PercAND(1,0,1,1,-1.5)
```

```
Out[17]: 0
```

```
In [18]: PercAND(1,1,1,1,-1.5)
```

```
Out[18]: 1
```

```
In [ ]:
```