

```
In [2]: import numpy as np
```

```
In [3]: def unitstep(v):  
        if v>=0:  
            return 1  
        else:  
            return 0
```

```
In [5]: def PerceptronModel(x,w,b):  
        v=np.dot(w,x)+b  
        print()  
        y=unitstep(v)  
        return y
```

```
In [16]: def AND_logicFunction(x):  
        w=np.array([1,1])  
        b=-1.5  
        return PerceptronModel(x,w,b)
```

```
In [17]: test1=np.array([0,0])  
        test2=np.array([0,1])  
        test3=np.array([1,0])  
        test4=np.array([1,1])
```

```
In [18]: print("AND({},{})={}".format(0,0,AND_logicFunction(test1)))  
        print("AND({},{})={}".format(0,1,AND_logicFunction(test2)))  
        print("AND({},{})={}".format(1,0,AND_logicFunction(test3)))  
        print("AND({},{})={}".format(1,1,AND_logicFunction(test4)))
```

AND(0,0)=0

AND(0,1)=0

AND(1,0)=0

AND(1,1)=1

```
In [ ]:
```