**PROGRAM 14**

**Aim:** Write a program to implement MP Neuron using AND logic functions.

**Code:**

import numpy as np

x=np.array([[1,1],[1,0],[0,1],[0,0]])

t=np.array([[1],[0],[0],[0]])

w=np.array([[0],[0]])

theta=1

yin=np.zeros(shape=(4,1))

y=np.zeros(shape=(4,1))

yin=np.dot(x,w)

i=0

found=0

while(found==0):

     i=0

     yin=np.dot(x,w)

     #print(yin)

     while(i<4):

         if yin[i]>=theta:

             y[i]=1

             i=i+1

        else:

             y[i]=0

             i=i+1

    #print("y",y)

    #print("t",t)

     if (y==t).all():

         print("MODEL IS TRAINED ")

         print("\nOutput : \n",y)

         print("\nweights : ",w,"\n")

         print("theta : ",theta)

         found=1

    else:

        print("MODEL IS NOT TRAINED")

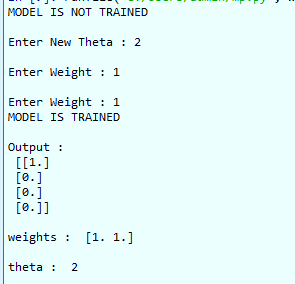
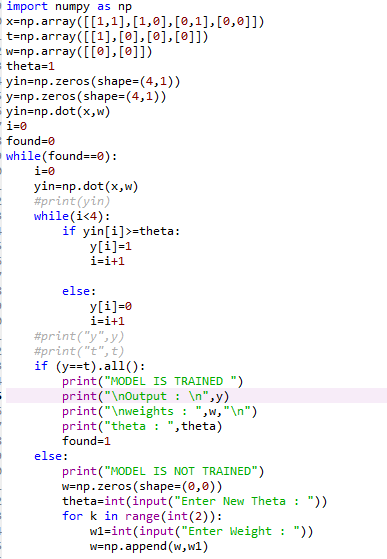
         w=np.zeros(shape=(0,0))

         theta=int(input("Enter New Theta : "))

         for k in range(int(2)):

             w1=int(input("Enter Weight : "))

             w=np.append(w,w1)



**PROGRAM 15**

**Aim:** Write a program to implement MP Neuron using OR logic functions.

**Code:**

import numpy as np

x=np.array([[1,1],[1,0],[0,1],[0,0]])

t=np.array([[1],[1],[1],[0]])

w=np.array([[0],[0]])

theta=1

yin=np.zeros(shape=(4,1))

y=np.zeros(shape=(4,1))

yin=np.dot(x,w)

i=0

found=0

while(found==0):

     i=0

     yin=np.dot(x,w)

     #print(yin)

     while(i<4):

         if yin[i]>=theta:

             y[i]=1

             i=i+1

        else:

             y[i]=0

             i=i+1

    #print("y",y)

    #print("t",t)

     if (y==t).all():

         print("MODEL IS TRAINED ")

         print("\nOutput : \n",y)

         print("\nweights : ",w,"\n")

         print("theta : ",theta)

         found=1

    else:

        print("MODEL IS NOT TRAINED")

         w=np.zeros(shape=(0,0))

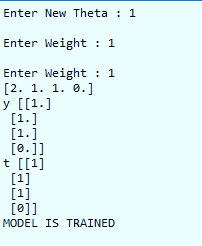
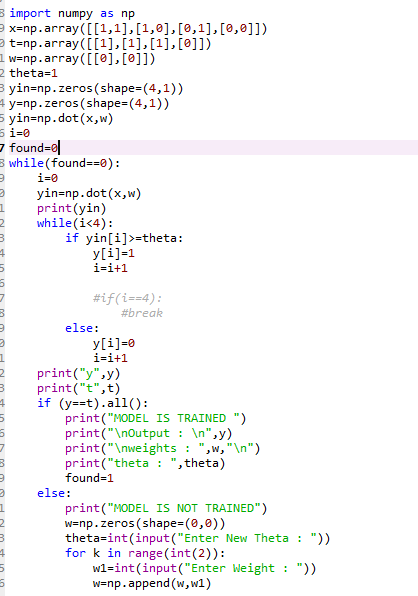
         theta=int(input("Enter New Theta : "))

         for k in range(int(2)):

             w1=int(input("Enter Weight : "))

             w=np.append(w,w1)





**PROGRAM 16**

**Aim:** Write a program to implement MP Neuron using AND-NOT logic functions.

**Code:**

import numpy as np

x=np.array([[1,1],[1,0],[0,1],[0,0]])

t=np.array([[0],[1],[0],[0]])

w=np.array([[0],[0]])

theta=1

yin=np.zeros(shape=(4,1))

y=np.zeros(shape=(4,1))

yin=np.dot(x,w)

i=0

found=0

while(found==0):

     i=0

     yin=np.dot(x,w)

     #print(yin)

     while(i<4):

         if yin[i]>=theta:

             y[i]=1

             i=i+1

        else:

             y[i]=0

             i=i+1

    #print("y",y)

    #print("t",t)

     if (y==t).all():

         print("MODEL IS TRAINED ")

         print("\nOutput : \n",y)

         print("\nweights : ",w,"\n")

         print("theta : ",theta)

         found=1

    else:

        print("MODEL IS NOT TRAINED")

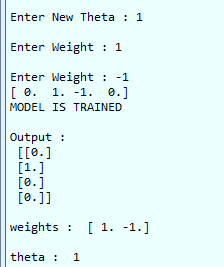
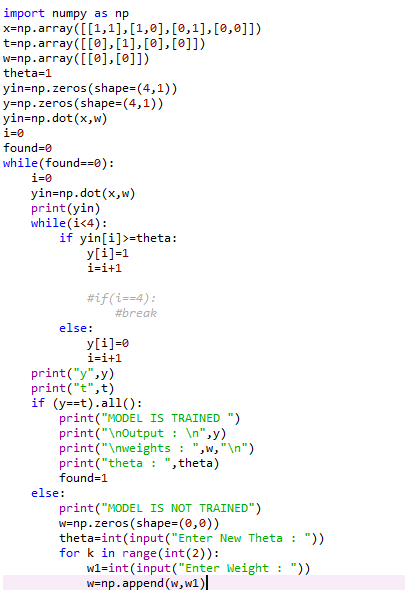
         w=np.zeros(shape=(0,0))

         theta=int(input("Enter New Theta : "))

         for k in range(int(2)):

             w1=int(input("Enter Weight : "))

             w=np.append(w,w1)



**PROGRAM 17**

**Aim:** Write a program to implement MP Neuron using NOT logic functions.

**Code:**

import numpy as np

x=np.array([[0],[1]])

t=np.array([[1],[0]])

w=np.array([0])

theta=1

yin=np.zeros(shape=(2,1))

y=np.zeros(shape=(2,1))

yin=np.dot(x,w)

i=0

found=0

while(found==0):

     i=0

     yin=np.dot(x,w)

     print(yin)

     while(i<2):

         if yin[i]>=theta:

             y[i]=1

             i=i+1

             #if(i==4):

                 #break

         else:

             y[i]=0

             i=i+1

     print("y",y)

     print("t",t)

     if (y==t).all():

         print("MODEL IS TRAINED ")

         print("\nOutput : \n",y)

         print("\nweights : ",w,"\n")

         print("theta : ",theta)

         found=1

     else:

         print("MODEL IS NOT TRAINED")

         w=np.zeros(shape=(0,0))

        theta=int(input("Enter New Theta : "))

         for k in range(int(1)):

             w=int(input("Enter Weight : "))

