

PROGRAM 14

AIM: Write a program in Python to implement Bidirectional Associative Memory (BAM) network to store and test the given patterns.

CODE:

```
import numpy as np
x1=np.array([[1,1,1,-1,1,-1,-1,1,-1,1,-1]])
x2=np.array([[1,1,1,1,-1,1,1,-1,1,1,1]])
x3=np.array([[1,1,1,-1,1,-1,-1,1,-1,1,1]])
t1 = np.array([[1],[1]])
t2 = np.array([[1],[1]])
w1=np.zeros((12,2),dtype=int)
w2=np.zeros((12,2),dtype=int)
w=np.zeros((12,2),dtype=int)
i=0
while(i!=12):
w1[i][0]=x1[0][i]*t1[0][0]
w1[i][1]=x1[0][i]*t1[1][0]
w2[i][0]=x2[0][i]*t2[0][0]
w2[i][1]=x2[0][i]*t2[1][0]
i=i+1
w=w1+w2
print('The Weight Matrix is:\n')
print(w)
Yin11=Yin12=Yin21=Yin22=Yin31=Yin32=0
y1=0
y2=0
i=0
while(i!=12):
    Yin11=Yin11+(x1[0][i]*w[i][0])
    Yin12=Yin12+(x1[0][i]*w[i][1])
    Yin21=Yin21+(x2[0][i]*w[i][0])
    Yin22=Yin22+(x2[0][i]*w[i][1])
    Yin31=Yin31+(x3[0][i]*w[i][0])
    Yin32=Yin32+(x3[0][i]*w[i][1])
    i=i+1
if(Yin11>0):
    Yin11=1
else:
    Yin11=-1
if(Yin12>0):
    Yin12=1
else:
    Yin12=-1
if(Yin21>0):
    Yin21=1
else:
    Yin21=-1
if(Yin22>0):
    Yin22=1
else:
    Yin22=-1
if(Yin31>0):
```

PROGRAM 14

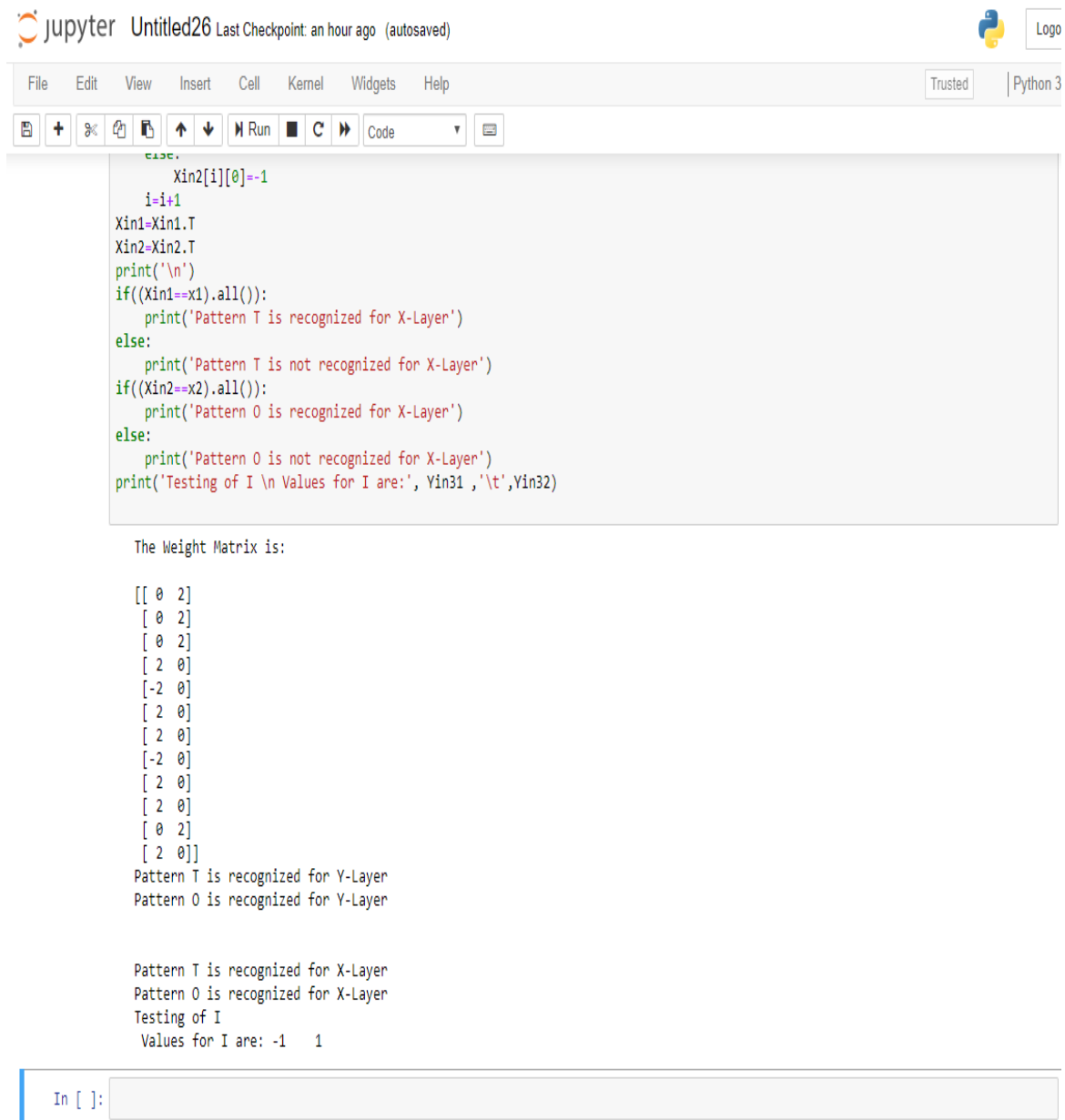
```
Yin31=1
else:
    Yin31=-1
if(Yin32>0):
    Yin32=1
else:
    Yin32=-1

if((Yin11==1) and (Yin12==1)):
    print('Pattern T is recognized for Y-Layer')
else:
    print('Pattern T is not recognized for Y-Layer')
if((Yin21==1) and (Yin22==1)):
    print('Pattern O is recognized for Y-Layer')
else:
    print('Pattern O is not recognized for Y-Layer')

i=0
Xin1=np.zeros((12,1),dtype=int)
Xin2=np.zeros((12,1),dtype=int)
while(i!=12):
    Xin1[i][0]=Xin1[i][0]+((Yin11*w[i][0])+(Yin12*w[i][1]))
    if(Xin1[i][0]>0):
        Xin1[i][0]=1
    else:
        Xin1[i][0]=-1
    Xin2[i][0]=Xin2[i][0]+((Yin21*w[i][0])+(Yin22*w[i][1]))
    if(Xin2[i][0]>0):
        Xin2[i][0]=1
    else:
        Xin2[i][0]=-1
    i=i+1
Xin1=Xin1.T
Xin2=Xin2.T
print('\n')
if((Xin1==x1).all()):
    print('Pattern T is recognized for X-Layer')
else:
    print('Pattern T is not recognized for X-Layer')
if((Xin2==x2).all()):
    print('Pattern O is recognized for X-Layer')
else:
    print('Pattern O is not recognized for X-Layer')
print('Testing of I \n Values for I are:', Yin31 ,'\t',Yin32)
```

PROGRAM 14

OUTPUT:



The image shows a Jupyter Notebook interface. At the top, the title bar says "jupyter Untitled26" with a "Last Checkpoint: an hour ago (autosaved)" status. The top right has a "Python 3" logo and a "Trusted" button. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu bar is a toolbar with icons for saving, adding cells, running, and other functions. The main area contains a code cell with the following Python code:

```
else:
    Xin2[i][0]=-1
    i=i+1
Xin1=Xin1.T
Xin2=Xin2.T
print('\n')
if((Xin1==x1).all()):
    print('Pattern T is recognized for X-Layer')
else:
    print('Pattern T is not recognized for X-Layer')
if((Xin2==x2).all()):
    print('Pattern O is recognized for X-Layer')
else:
    print('Pattern O is not recognized for X-Layer')
print('Testing of I \n Values for I are:', Yin31, '\t', Yin32)
```

The output of the code is displayed below the code cell:

```
The Weight Matrix is:

[[ 0  2]
 [ 0  2]
 [ 0  2]
 [ 2  0]
 [-2  0]
 [ 2  0]
 [ 2  0]
 [-2  0]
 [ 2  0]
 [ 2  0]
 [ 0  2]
 [ 2  0]]
Pattern T is recognized for Y-Layer
Pattern O is recognized for Y-Layer

Pattern T is recognized for X-Layer
Pattern O is recognized for X-Layer
Testing of I
Values for I are: -1  1
```

At the bottom of the notebook, there is an input prompt "In []:" followed by a text box.