```
echo on
%Checking the designed weights for exercise 4.2
%Create a perceptron network
net = newp([-1 1;-1 1],1);
%Let the designed weight and bias be W=[-1,0]' and b=-0.5
net.IW{1,1}=[-1 0];
net.b{1}=-0.5;
%The four input vectors and the corresponding targets are
p = {[-1;1] [-1;-1] [0;0] [1;0]};
t = {1 1 0 0};
%Simulate the network to get the outputs
a = sim(net,p)
%Finding the error for each of the inputs
error = {a{1}-t{1} a{2}-t{2} a{3}-t{3} a{4}-t{4}}
echo off
```

```
echo on
%Checking the designed weights for exercise 18.5
%Define the binary prototype vectors
P = [-1 1 1 -1;1 -1 1 -1]'
%Create a hopfield network
net = newhop(P);
%Set the weights and bias of the network to be the designed ones
net.LW{1,1} = [2 -2 0 0;-2 2 0 0;0 0 2 -2;0 0 -2 2];
net.b{1,1} = [0;0;0;0];
%Check the designed weights by starting with the prototype vectors
Y = sim(net,2,[],P);
Y
%The result shows that the network is stable at the design prototypes
%Next, try another input that is not a design prototype, such as
A = [-0.8;0.7;0.6;-0.5]
Ya= sim(net,1,[],A);
Ya
%Observe that the vector A converges to the closer prototype vector.
echo off
```