```
clc; clear;
%%
% Name: Devosmita Chatterjee
% Assignment1 a

patterns=[12,24,48,70,100,120];

p_err=zeros(1,6);

for l = 1:1:6
count = 0;
n_trials = 10^5;
for iteration = 1:1:n_trials
    p=patterns(l);

    N =120;% number of pixels of each pattern

    % Input data
    m = randi([0 1], N,p);% Generate random pattern

    x = zeros(N,p);

    for a = 1:1:N
        for j = 1:1:p
            if m(a,j) == 0
                x(a,j) = 1;
            else
                x(a,j) = -1;
            end
        end
    end

    % Calculate weight matrix
    W = zeros(N,N);

    for j = 1:1:p
        W = x(:,j)*x(:,j)'+W;
    end
    W = (1/N)*W;
    W = W - diag(diag(W));

    j1 = randi([1 p],1,1);

    a1 = randi([1 N],1,1);% Generate randomly chosen neuron for the
asynchronous update
    sum = 0;
    for b = 1:1:N
        sum = sum + W(a1, b) * x(b,j1);
    end

    % signum function
    out = 0;
    if (sum ~= 0)
```

```
        if (sum < 0)
            out = -1;
        end
        if (sum > 0)
            out = +1;
        end
    end

    if (out~=x(a1,j1))
        count=count+1;
    end
end

    % One-step error probability for each of the patterns
    p_err(l)=round(count/n_trials,4);
    iteration = iteration+1;
end
disp(['p_err = ',num2str(p_err)])% Display error probability for six
patterns

%p_err = 0.0004     0.0116     0.0554     0.0945     0.1339     0.1585
```

```
clc; clear;
%%
% Name: Devosmita Chatterjee
% Assignment1 b

patterns=[12,24,48,70,100,120];
p_err=zeros(1,6);

for l = 1:1:6
count = 0;
n_trials = 10^5;
for iteration = 1:1:n_trials
    p=patterns(l);

    N =120;% number of pixels of each pattern

    % Input data
    m = randi([0 1], N,p);% Generate random pattern

    x = zeros(N,p);

    for a = 1:1:N
        for j = 1:1:p
            if m(a,j) == 0
```

```
            x(a,j) = 1;
        else
            x(a,j) = -1;
        end
    end
end

% Calculate weight matrix
W = zeros(N,N);

for j = 1:1:p
    W = x(:,j)*x(:,j)'+W;
end
W = (1/N)*W;

j1 = randi([1 p],1,1);

a1 = randi([1 N],1,1);% Generate randomly chosen neuron for the
asynchronous update
sum = 0;
for b = 1:1:N
    sum = sum + W(a1, b) * x(b,j1);
end

% signum function
out = 0;
if (sum ~= 0)
    if (sum < 0)
        out = -1;
    end
    if (sum > 0)
        out = +1;
    end
end

if (out~=x(a1,j1))
    count=count+1;
end
end

% One-step error probability for each of the patterns
p_err(l)=round(count/n_trials,4);
iteration = iteration+1;
end
disp(['p_err = ',num2str(p_err)])% Display error probability for six
patterns

%p_err = 0.0001     0.0029     0.0126     0.0186     0.0218     0.0223
```