

```

clc; clear;
%%
% Name: Devosmita Chatterjee
% Assignment2 1a&1b

x(:, :, 1)=[ [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1] ];

x(:, :, 2)=[ [ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1] ];

x(:, :, 3)=[ [ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1] ];

```

```

x(:, :, 4)=[ [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1] ];

x(:, :, 5)=[ [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1] ];

%%

y(:, :, 1)=[ [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, -1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, -1, 1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, 1, -1, -1, 1, 1, 1, -1];
              [-1, -1, -1, 1, -1, -1, 1, 1, 1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
              [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
              [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
              [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
              [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];

%%

stored = zeros(size(x,3),160);

```

```

input = zeros(size(y,3),160);

% Stored pattern
for n = 1:1:size(x,3)
    for i = 1:1:size(x,1)
        for j = 1:1:size(x,2)
            if x(i,j,n) == 1
                stored(n,(i-1)*10+j) = 1;
            else
                stored(n,(i-1)*10+j) = -1;
            end
        end
    end
end

% Input pattern
for n = 1:1:size(y,3)
    for i = 1:1:size(y,1)
        for j = 1:1:size(y,2)
            if y(i,j,n) == 1
                input(n,(i-1)*10+j) = 1;
            else
                input(n,(i-1)*10+j) = -1;
            end
        end
    end
end

%%
% Calculate weight matrix
W = zeros(size(x,1),size(x,2));

for i = 1:1:size(x,1)*size(x,2)
    for j = 1:1:size(x,1)*size(x,2)
        weight = 0;
        if (i ~= j)
            for n = 1:1:size(x,3)
                weight = stored(n,i) .* stored(n,j) + weight;
            end
            W(i,j) = (1/(size(x,1)*size(x,2)))*weight;
        end
    end
end

%%
for n = 1:1:size(y,3)
    iteration = 0;
    Lastiteration = 0;
    flag = true;
    while flag
        iteration = iteration + 1;
        % Generate random element for the asynchronous correction
        i = randi([1 size(x,1)*size(x,2)],1,1);
        sum = 0;
        for j = 1:1:size(x,1)*size(x,2)
            sum = sum + W(i, j) * input(n,j);
        end
    end
end

```

```

        % Calculate signum function
        out = 0;
        changed = 0;
        if (sum ~= 0)
            if (sum < 0)
                out = -1;
            end
            if (sum > 0)
                out = +1;
            end
            if (out ~= input(n, i))
                changed = 1;
                input(n,i) = out;
            end
        end
        if (changed == 1)
            Lastiteration = iteration;
        end
        if (iteration - Lastiteration > 10^5)
            flag = false;
        end
    end
end
%%
% (A)
A=[input(1:10);
input(11:20);
input(21:30);
input(31:40);
input(41:50);
input(51:60);
input(61:70);
input(71:80);
input(81:90);
input(91:100);
input(101:110);
input(111:120);
input(121:130);
input(131:140);
input(141:150);
input(151:160)];
disp(A)
fprintf('\n\n')
%%
% (B)
for n = 1:1:size(x,3)
    if (isequal(input,stored(n,:))==1)
        disp(n);
    elseif (isequal(input,-stored(n,:))==1)
        disp(-n);
    else
        disp(6);
    end
end
end

%[[-1,    -1,    1,    1,    1,    1,    1,    1,    -1,    -1],

```

```

% [-1, -1, 1, 1, 1, 1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1],
% [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, -1, -1, -1, -1, 1, 1, 1, -1],
% [-1, -1, 1, 1, 1, 1, 1, 1, 1, -1],
% [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1]]

```

```

% 4

```

```

clc; clear;

```

```

%%

```

```

% Name: Devosmita Chatterjee

```

```

% Assignment2 2a&2b

```

```

x(:, :, 1)=[ [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1];
  [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
  [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, 1, 1, 1, -1, -1, 1, 1, 1, -1];
  [-1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
  [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
  [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1] ];

```

```

x(:, :, 2)=[ [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
  [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
  [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
  [-1, -1, -1, 1, 1, 1, 1, -1, -1, -1];

```

[illegible]

```

        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
        [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1] ];
%%

```

```

y(:, :, 1) = [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
              [-1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [-1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
              [1, 1, 1, 1, 1, 1, 1, -1, -1, 1];
              [1, 1, 1, 1, 1, 1, 1, -1, -1, 1]];
%%

```

```

%%

stored = zeros(size(x,3),160);

```

```

input = zeros(size(y,3),160);

```

```

% Stored pattern
for n = 1:1:size(x,3)
    for i = 1:1:size(x,1)
        for j = 1:1:size(x,2)
            if x(i,j,n) == 1
                stored(n, (i-1)*10+j) = 1;
            else
                stored(n, (i-1)*10+j) = -1;
            end
        end
    end
end
end

```

```

% Input pattern
for n = 1:1:size(y,3)
    for i = 1:1:size(y,1)
        for j = 1:1:size(y,2)
            if y(i,j,n) == 1
                input(n, (i-1)*10+j) = 1;
            else
                input(n, (i-1)*10+j) = -1;
            end
        end
    end
end
end

```

```

end
%%
% Calculate weight matrix
W = zeros(size(x,1),size(x,2));

for i = 1:1:size(x,1)*size(x,2)
    for j = 1:1:size(x,1)*size(x,2)
        weight = 0;
        if (i ~= j)
            for n = 1:1:size(x,3)
                weight = stored(n,i) .* stored(n,j) + weight;
            end
        end
        W(i,j) = (1/(size(x,1)*size(x,2)))*weight;
    end
end
%%
for n = 1:1:size(y,3)
    iteration = 0;
    Lastiteration = 0;
    flag = true;
    while flag
        iteration = iteration + 1;
        % Generate random element for the asynchronous correction
        i = randi([1 size(x,1)*size(x,2)],1,1);
        sum = 0;
        for j = 1:1:size(x,1)*size(x,2)
            sum = sum + W(i, j) * input(n,j);
        end
        % Calculate signum function
        out = 0;
        changed = 0;
        if (sum ~= 0)
            if (sum < 0)
                out = -1;
            end
            if (sum > 0)
                out = +1;
            end
            if (out ~= input(n, i))
                changed = 1;
                input(n,i) = out;
            end
        end
        if (changed == 1)
            Lastiteration = iteration;
        end
        if (iteration - Lastiteration > 10^5)
            flag = false;
        end
    end
end
%%
% (A)
A=[input(1:10);
input(11:20);

```



```

clc; clear;
%%
% Name: Devosmita Chatterjee
% Assignment2 3a&3b

x(:, :, 1)=[ [ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, 1, 1, 1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1];
[ -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1] ];

x(:, :, 2)=[ [ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1];
[ -1, -1, -1, 1, 1, 1, 1, -1, -1, -1] ];

x(:, :, 3)=[ [ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, -1, -1, -1, -1, -1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1];
[ 1, 1, 1, 1, 1, 1, 1, 1, -1, -1] ];

```

```

x(:, :, 4)=[ [ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, 1, 1, 1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, 1, -1];
[ -1, -1, 1, 1, 1, 1, 1, 1, -1, -1] ];

x(:, :, 5)=[ [ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, -1, -1, -1, -1, 1, 1, -1];
[ -1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
[ -1, 1, 1, 1, 1, 1, 1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1];
[ -1, -1, -1, -1, -1, -1, -1, 1, 1, -1] ];

%%

y(:, :, 1)=[ [1, 1, 1, 1, 1, 1, 1, 1, 1, 1];
[1, 1, 1, -1, -1, -1, -1, 1, 1, 1];
[1, 1, -1, -1, -1, -1, -1, -1, 1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, -1, -1, -1, 1, 1, -1, -1, -1, 1];
[1, 1, -1, -1, -1, -1, -1, -1, 1, 1];
[1, 1, 1, -1, -1, -1, -1, 1, 1, 1];
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1] ];

%%

```

```

stored = zeros(size(x,3),160);

input = zeros(size(y,3),160);

% Stored pattern
for n = 1:1:size(x,3)
    for i = 1:1:size(x,1)
        for j = 1:1:size(x,2)
            if x(i,j,n) == 1
                stored(n,(i-1)*10+j) = 1;
            else
                stored(n,(i-1)*10+j) = -1;
            end
        end
    end
end

% Input pattern
for n = 1:1:size(y,3)
    for i = 1:1:size(y,1)
        for j = 1:1:size(y,2)
            if y(i,j,n) == 1
                input(n,(i-1)*10+j) = 1;
            else
                input(n,(i-1)*10+j) = -1;
            end
        end
    end
end

%%
% Calculate weight matrix
W = zeros(size(x,1),size(x,2));

for i = 1:1:size(x,1)*size(x,2)
    for j = 1:1:size(x,1)*size(x,2)
        weight = 0;
        if (i ~= j)
            for n = 1:1:size(x,3)
                weight = stored(n,i) .* stored(n,j) + weight;
            end
            W(i,j) = (1/(size(x,1)*size(x,2)))*weight;
        end
    end
end

%%
for n = 1:1:size(y,3)
    iteration = 0;
    Lastiteration = 0;
    flag = true;
    while flag
        iteration = iteration + 1;
        % Generate random element for the asynchronous correction
        i = randi([1 size(x,1)*size(x,2)],1,1);
        sum = 0;
        for j = 1:1:size(x,1)*size(x,2)
            sum = sum + W(i, j) * input(n,j);
        end
    end
end

```

```

end
% Calculate signum function
out = 0;
changed = 0;
if (sum ~= 0)
    if (sum < 0)
        out = -1;
    end
    if (sum > 0)
        out = +1;
    end
    if (out ~= input(n, i))
        changed = 1;
        input(n,i) = out;
    end
end
if (changed == 1)
    Lastiteration = iteration;
end
if (iteration - Lastiteration > 10^5)
    flag = false;
end
end
end
%%
% (A)
A=[input(1:10);
input(11:20);
input(21:30);
input(31:40);
input(41:50);
input(51:60);
input(61:70);
input(71:80);
input(81:90);
input(91:100);
input(101:110);
input(111:120);
input(121:130);
input(131:140);
input(141:150);
input(151:160)];
disp(A)
fprintf('\n\n')
%%
% (B)
for n = 1:1:size(x,3)
    if (isequal(input,stored(n,:))==1)
        disp(n);
    elseif (isequal(input,-stored(n,:))==1)
        disp(-n);
    else
        disp(6);
    end
end
end

```

```

% [1,      1,      1,      1,      1,      1,      1,      1,      1,      1,      1],
%      [1,      1,      1,      -1,      -1,      -1,      -1,      1,      1,      1],
%      [1,      1,      -1,      -1,      -1,      -1,      -1,      -1,      1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [ 1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      -1,      -1,      -1,      1,      1,      -1,      -1,      -1,      1],
%      [1,      1,      -1,      -1,      -1,      -1,      -1,      -1,      1,      1],
%      [1,      1,      1,      -1,      -1,      -1,      -1,      1,      1,      1],
%      [1,      1,      1,      1,      1,      1,      1,      1,      1,      1]]

% -1

```