clc;

clear;

close all;

% Define the number of source elements (binary source: 0 and 1)

n = 2;

% Define the probability of error for the binary symmetric channel

epsilon = input('Enter the probability of error (epsilon): ');

% Define the channel matrix P(Y|X) for a binary symmetric channel

q = [1 - epsilon, epsilon; epsilon, 1 - epsilon];

disp('Channel matrix P(Y|X):');

disp(q);

% Define source probability P(X) for a binary symmetric source

% P(X) = {0.5, 0.5} as specified in the problem

p = [0.5; 0.5];

disp('Source probability P(X):');

disp(p);

% Calculate joint probability P(X,Y) = P(X) \* P(Y|X)

px = diag(p); % P(X) as a diagonal matrix

pxy = px \* q; % P(X,Y) matrix

disp('Joint probability P(X,Y):');

disp(pxy);

% Calculate P(Y)

py = p' \* q; % P(Y) as a row vector

disp('Marginal probability P(Y):');

disp(py);

% Entropy of source H(X)

Hx = -sum(p .\* log2(p));

disp('H(X):');

disp(Hx);

% Entropy of destination H(Y)

Hy = -sum(py .\* log2(py));

disp('H(Y):');

disp(Hy);

% Joint Entropy H(X,Y)

Hxy = 0;

for i = 1:n

for j = 1:n

if pxy(i,j) > 0

Hxy = Hxy - pxy(i,j) \* log2(pxy(i,j));

end

end

end

disp('H(X,Y):');

disp(Hxy);

% Conditional Entropy H(Y|X)

Hy\_given\_X = Hxy - Hx;

disp('H(Y|X):');

disp(Hy\_given\_X);

% Conditional Entropy H(X|Y)

Hx\_given\_Y = Hxy - Hy;

disp('H(X|Y):');

disp(Hx\_given\_Y);

% Mutual Information I(X;Y) = H(X) - H(X|Y)

Ixy = Hx - Hx\_given\_Y;

disp('I(X;Y):');

disp(Ixy);

% Comment on the type of channel based on calculated values

if Hx\_given\_Y == 0

disp('This channel is a lossless channel.');

elseif Ixy == 0

disp('This channel is a useless channel.');

elseif Hx == Hy && Hy\_given\_X == 0

disp('This channel is a noiseless channel.');

else

disp('This channel has some noise.');

end