INPUT:

```
import numpy as np

c1 = [1, 1, 1, 1]

c2 = [1, -1, 1, -1]

c3 = [1, 1, -1, -1]

c4 = [1, -1, -1, 1]

rc = []

print("Enter the data bits:")

d1=int(input("Enter D1:"))

d2=int(input("Enter D2:"))

d3 = (input("Enter D3 :"))

d4 = int(input("Enter D4:"))

r1 = np.multiply(c1,d1)

r2 = np.multiply(c2,d2)

r3 = np.multiply(c3,d3)

r4 = np.multiply(c4,d4)

resultant_channel = r1+r2+r3+r4;

print("Resultant Channel", resultant_channel)

Channel = int(input("Enter the station to listen for C1 = 1 C2 = 2 , C3 = 3, C4=4:”))

if Channel == 1: rc =c1

elif Channel == 2: rc =c2

elif Channel == 3: rc =c3

elif Channel == 4 ; rc =c4

inner_product = np.multiply(resultant_channel,rc)
print("Inner Product", inner_product)
```

```
res1=sum(inner_product)

data = res1/len(inner_product)

print("Data bit that was sent", data)
```

```
clear;

N = 10^6; % Number of bits or symbols

rand('state', 100);       % Initialize rand()
randn('state', 200);      % Initialize randn()

% Transmitter
ip = rand(1, N) > 0.5;   % Generate 0,1 with equal probability
s = 2 * ip - 1;          % BPSK modulation: 0 -> -1, 1 -> +1

n = 1/sqrt(2) * (randn(1, N) + 1j * randn(1, N)); % White Gaussian noise, complex

Eb_N0_dB = -3:10;        % Multiple Eb/N0 values in dB

nErr = zeros(1, length(Eb_N0_dB)); % Initialize error array

for ii = 1:length(Eb_N0_dB)
    % Noise addition
    y = s + 10^(-Eb_N0_dB(ii)/20) * n;

    % Receiver - hard decision decoding
    ipHat = real(y) > 0;

    % Count the errors
    nErr(ii) = sum(ip ~= ipHat);
end

simBer = nErr / N; % Simulated BER
theoryBer = 0.5 * erfc(sqrt(10.^(Eb_N0_dB / 10))); % Theoretical BER

% Plotting
close all;
figure;
semilogy(Eb_N0_dB, theoryBer, 'b.-'); hold on;
```

```
semilogy(Eb_N0_dB, simBer, 'mx-');
axis([-3 10 10^-5 1]);
grid on;
legend('Theory', 'Simulation');
xlabel('Eb/No (dB)');
ylabel('Bit Error Rate (BER)');
title('Bit Error Probability for BPSK modulation');
```

System Configuration Dialog---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>**enable**

Router**#configure terminal**

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)**#interface gigabitethernet 0/0/0**

Router(config-if)**#no shutdown**

Router(config-if)#

%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

Router(config-if)**#ip address 192.168.1.1 255.255.255.0**

Router(config-if)**#ex**

Router(config)**#ip dhcp pool abc 1.0 255.255.255.0**

Router(dhcp-config)**#default-router 1**

Router(dhcp-config)**#network 192.168.92.168.1.1**

Router(dhcp-config)**#dns-server 8.8.8.8**

Router(dhcp-config)**#ex**

Router(config)#%DHCPD-4-PING_CONFLICT: DHCP address conflict: server pinged

192.168.1.1.

Router#

%SYS-5-CONFIG_I: Configured from console by console