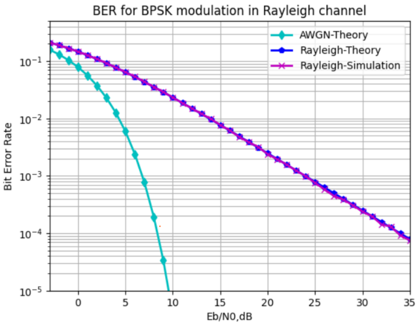
EXPT-1

```python
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=int(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, 3-3 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",data)
```

EXPT-4

```python
import numpy as np
from scipy.special import erfc
import matplotlib.pyplot as plt

N=int(1e6)
Eb_N0_dB=np.arange(-3,60)

ip= np.random.rand(N) > 0.5
s= 2*ip -1

nErr= np.zeros(len(Eb_N0_dB))
for i, Eb_N0 in enumerate(Eb_N0_dB):
  n=np.sqrt(0.5) * (np.random.randn(N)+1j*np.random.randn(N))
  h=np.sqrt(0.5) * (np.random.randn(N)+1j*np.random.randn(N))
  y= h*s +np.sqrt(10**(-Eb_N0/10))*n
  ipHat=(np.real(y/h)>0).astype(int)
  nErr[i]=np.sum(ip != ipHat)

simBer=nErr/N
theoryBerAWGN = 0.5 *erfc(np.sqrt(10**(Eb_N0_dB/10)))
theoryBer=0.5*(1- np.sqrt(10**(Eb_N0_dB/10) / (1+10**(Eb_N0_dB/10))))

plt.semilogy(Eb_N0_dB, theoryBerAWGN, 'cd-',linewidth=2)
plt.semilogy(Eb_N0_dB, theoryBer, 'bp-',linewidth=2)
plt.semilogy(Eb_N0_dB, simBer, 'mx-',linewidth=2)
plt.axis([-3,35,1e-5,0.5])
plt.grid(True,which="both")
plt.legend(['AWGN-Theory','Rayleigh-Theory','Rayleigh-Simulation'])
plt.xlabel('Eb/N0,dB')
plt.ylabel('Bit Error Rate’)
plt.title('BER for BPSK modulation in Rayleigh channel')
plt.show()
```


BER for BPSK modulation in Rayleigh channel
AWGN-Theory
Rayleigh-Theory
Rayleigh-Simulation
Bit Error Rate
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Eb/N0,dB