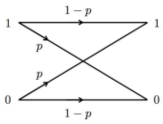
# PCS Homework 4 Spring 2025

0. Write your RUID
1. Set random seed in MATLAB using your RUID (using **rng(RUID)** function) and generate a random sequence of equally likely N=10000 input bits using a rand function and denote with **bb**.
   Write the sequence of first 10. Use these N bits in the rest of the homework.
2. Consider a Binary Symmetric Channel (BSC) below with error probability Pe =p (i.e., the probability of deciding/receiving 0/1 when 1/0 is transmitted/input is Pe=p).




   denote the sequence of N transmitted/input bits with **bb** and received/estimated bits with **be** and errors with **ee**=**bb** *xor* **be. ee** is a sequence of 0s and 1s with a 0 whenever estimated bits $\mathbf{be_i}$ are equal to transmitted $\mathbf{bb_i}$ and 1, otherwise. Inputs bits and errors are iid with P($bb_i$ =1)= .5 and P($ee_i$ =1)= p, for all i=0,…N-1 bits.
   Run the following simulation in matlab J=10 times as follows: generate N=10000 independent errors and denote resulting the j-th error sequence with $\mathbf{eeg^j}$ each having an error with probability $p^j=2^{-j}$; whenever, an error occurs (i.e., $eeg^j_i$=1) set the received bit $beg^j_i$ to be different from the corresponding input bit $bb_i$ (generated in problem 1 above). Run this for j=1…10. Your code will generate 10 error $\mathbf{eeg^j}$ and output $\mathbf{beg^j}$ sequences of length N=10000. Store/remember these 20 sequences.
   a. Compute (analytically) the probability P($be_i$ =$bb_i$), for all i=0,…N-1 bits as a function of p.
   b. Compute (analytically) the expected number of errors (i.e., expected number of times that $be_i$ and $bb_i$ are different) as a function of p.
   c. Compute (analytically) the expected average number of errors normalized per transmitted bit.
   d. Simulate, compute & plot vs $p^j$ , j=1:10, the average number of errors (places where $\mathbf{beg^j}$ and **bbg** are different) normalized per transmitted bit.
   e. Compute and plot vs $p^j$ , j=1:10 (from simulation in 2d) above), the mean square error of the average number of errors from 2d) (places where $\mathbf{beg^j}$ and **bbg** are different). Compute the error relative to the normalized expected number of errors per bit computed in 1 c) above.
3. Consider an additive white Gaussian noise (AWGN) channel whose received sequence is $rr_i$= $aa_i$+$nn_i$, i=0,…, N-1, where $aa_i$ are iid equally likely 1 and -1 obtained from **bb** in problem 1 using the mapping $aa_i$ = 2 $bb_i$ -1, and $nn_i$ are iid Gaussian random variables with mean 0, and variance *v*. Signal to noise ratio is SNR=1/v. Use a threshold detector that decides that $be_i$ =1was transmitted if $rr_i$>0 and decides $be_i$ =0, otherwise, for the i-th bit. The error occurs when $bb_i$ and $be_i$ are different and is registered in the error sequence as $ee_i$=1. For matlab simulation, similar to problem 2, you will generate a family of J=8 noise sequences $\mathbf{nn^j}$ each with different $SNR^j$, computed such that $SNRdB^j=10\log_{10}(SNR^j)$= [0:7] dB, j=1,…,8; For each $\mathbf{nn^j}$, you will generate the corresponding received sequences $\mathbf{rr^j}$ (all based on the same sequence **bb** from problem 1) and the decision bit sequences $\mathbf{be^j}$ using the threshold detector.
   a. Compute (analytically) the probability of error Pe=P($be_i$ not equal $bb_i$) as a function of SNR. (Hint: Leave the result in terms of the Q-function).
   b. Compute (analytically) the expected number of errors (i.e., expected number of times that $be_i$ and $bb_i$ are different) as a function of SNR.
   c. Simulate, compute & plot vs $SNRdB^j$ , j=1:10, the average number of bit errors (places where $\mathbf{be^j}$ and **bb** are different) normalized by the number of transmitted bits. Overlay the plot from the simulation result with the plot from the analytical Pe result computed in 3a). For plotting use semilogy(**SNRdB**, normalized **average number of errors**) function, use the *grid on* function, *hold on* to overlay two plots, and ensure that you label both the y and the x-axis appropriately

   **Report submission/output:**
   Results and MATLAB code (executable via copy and paste) in a single pdf. MATLAB code should generate all the plots in that single run and display all results requested as a result of the code run (including analytical results).