Problem 1: For an alphabet $A = \{a_1, a_2, a_3\}$ with probabilities $P(a_1) = 0.6$, $P(a_2) = 0.3$, and $P(a_3) = 0.1$. (3pts)

(a) Design a 3-bit Tunstall code for this alphabet.
(b) Find the redundancy (code rate – entropy rate).

Problem 2: Suppose you saw this one game in which, a guy repeatedly tosses a fair coin (at least he claims that it is a fair coin, and hence with probability $P(\text{head}) = 1/2$) until either (a) the outcome is head or (b) the number of consecutive tail outcomes reaches 4. (8pts)

(a) Code these outcomes using Golomb code with $m = 4$. What is average code rate? (note that the uncoded outcomes have the forms: 1, 01, 001, 0001, 0000, with 1 representing head and 0 representing tail.)
(b) Being a very observant person, you notice that the guy is actually cheating his audiences by using a biased coin, in which the probability of head $P(\text{head})$ is not 1/2. Using $m = 4$, can you derive the equation for the average Golomb code rate as a function of $P(\text{head})$?
(c) If you are going to code these outcomes using runlength code, what is the average code rate as a function of probability $P(\text{head})$, assuming you always code the run-lengths of tails using 2-bit fixed-length code?

Problem 3: Do problem 7 in chapter 4. This problem refers to integer arithmetic coding with scaling. Show steps by steps during encoding and decoding. (8pts)

Problem 4: (bonus) Show that the remainder bits in Golomb code can be viewed as a prefix code (1pt).