Problem 1. Let the alphabet be $\mathcal{X} = \{a, b\}$. Consider the infinite sequence $X_1^\infty = abababababababab......$

(a) What is the compressibility of $\rho(X_1^\infty)$ using finite-state machines (FSM) as defined in class? Justify your answer.

(b) Design a specific FSM, call it M, with at most 4 states and as low a $\rho_M(X_1^\infty)$ as possible. What compressibility do you get?

(c) Using only the result in point (a) but no specific calculations, what is the compressibility of $X_1^\infty$ under the Lempel-Ziv algorithm, i.e., what is $\rho_{LZ}(X_1^\infty)$?

(d) Re-derive your result from point (c) but this time by means of an explicit computation.

Problem 2. From the notes on the Lempel-Ziv algorithm, we know that the maximum number of distinct words $c$ a string of length $n$ can be parsed into satisfies

$$n > c \log_K (c/K^3)$$

where $K$ is the size of the alphabet the letters of the string belong to. This inequality lower bounds $n$ in terms of $c$. We will now show that $n$ can also be upper bounded in terms of $c$.

(a) Show that, if $n \geq \frac{1}{2} m(m - 1)$, then $c \geq m$.

(b) Find a sequence for which the bound in (a) is met with equality.

(c) Show now that $n < \frac{1}{2} c(c + 1)$. 