1. Consider the probability distribution \( a : 1/4, b : 1/2, c : 1/4 \).

   (a) Use arithmetic coding with scaling to code the string \( bbbba \). Show the steps in the process and the value of \( C \) which keeps track of the number of complementary bits to be output after a 0 or 1 is output. I chose this example because the scaled interval are very easy to calculate.

   (b) Use arithmetic decoding with scaling to decode 00000000001 (10 zeros followed by a 1) assuming the string decoded is of length 6.

2. Let us try LZW on a special class of inputs too. Again assume the two symbol alphabet \( \{a, b\} \). Consider the following strategy for encoding the dictionary symbols from LZW. Start with a dictionary of size 2 and use just one bit to transmit a symbol. When the dictionary fills up we double its size to 4 and use two bits to transmit a word in the dictionary. This doubling happens when ever the dictionary fills.

   (a) Encode \( a^6 \) and \( a^{28} \) with this version of LZW.

   (b) Compute the length, as a function of \( n \), of the encoding of \( a^n \) with this version of LZW. (You may restrict yourself to easy \( n \)'s to work with if that helps.)

   (c) Encode \( a^6 \) and \( a^{28} \) using the \( \gamma \)-code to represent the dictionary symbols from LZW on the strings \( a^6 \) and \( a^{28} \).

   (d) Compute the length, as a function of \( n \), of the encoding of \( a^n \) using the \( \gamma \)-code to represent the dictionary symbols of LZW. (You may restrict yourself to easy \( n \)'s to work with if that helps.)