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. In some situations a data file has the property of having a relatively small “working set”. This means that the current symbol most often comes from a fairly small set of symbols. For example, consider the string $x$ of symbols in the alphabet $\{a, b, c, d, e, f\}$:

   $$x = abccaabbcabddbcbdceddeddeecdeeffddefdfdeeff$$

which tends to have a working set of about size 3.

In the move-to-front algorithms we first give an initial index to each symbol as follows:

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f}
\end{array}
\]

Suppose symbol $x$ with index $i$ is encountered in the input stream. The index $i$ is output. Then the index of $x$ becomes 0 and all the symbols indexed $< i$ have their index increased by 1. For example the input $y$

$$y = bbbfbb$$

has output 100510 because after the first $b$ is input the indexing becomes

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\text{b} & \text{a} & \text{c} & \text{d} & \text{e} & \text{f}
\end{array}
\]

and after the $f$ is input then the indexing becomes

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\text{f} & \text{b} & \text{a} & \text{c} & \text{d} & \text{e}
\end{array}
\]
and after the fourth b is input the indexing becomes

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>f</td>
<td>a</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

(a) Compute the empirical entropy of the string \(x\). (The empirical entropy is done using the frequencies of the symbols found in the string.)

(b) Compute the empirical entropy of the string output in the move-to-front algorithm executed on \(x\).

(c) In move-to-front compression both the encoder and decoder know the initial indexing and the output of the move-to-front algorithm is losslessly encoded, say with arithmetic coding. Give one example of a data set that might be amenable to move-to-front compression, and explain why it is so. English text is not an example.