Homework 1

(Total 150 pts)

Due 5:00 pm, January 30, 2025 (Thursday)

Note: Submit electronically to Canvas three files ("HW1.pdf", as well as two M files "longest_run.m", and "rate.m").

1. Run-length coding.
   Suppose the input data source has 256 distinct symbols. Each symbol can be represented by 8 bits. If we use run-length coding, the run-lengths of the same symbol can be represented by B bits, where the run-length can be 1, 2, .,., 2^B. The choice of the B value might depend on the largest run-length of any symbol in the source. For example, consider the following input data sequence consisting of 16 symbols:

   49 50 48 49 48 48 49 **50 50 50 50** 49 49 48 50 50

   Symbol '50' has the longest run of 4, since it is repeated 4 times. Hence, if we choose $B = 2$, then the largest run-length that can be represented will be $2^B = 4$. Now, we apply run-length coding on the above sequence, such that each symbol and its corresponding run-length will be coded as a $(8 + B)$ bit binary number, where the first 8 bits are for the binary representation of the symbol, and the next B bits are for the binary representation of the run-length. Given $B = 2$, the first symbol '49', which has a run-length of 1, will be codes as a 10-bit binary number. Similarly, the next symbol '50' will be coded as another 10-bit binary number, and so on. Note, however, that the four consecutive symbol '50' in the middle will be coded as a 10-bit binary number, leading to significant data compression. In summary, the entire sequence above (consisting of 16 symbols, equivalent to 128 bits) can now be run-length coded to a compressed bitstream of 100 bits.

   It turns out that the efficiency of the run-length coding scheme will change with a different choice of B value. For example, if we choose $B = 1$ instead, then we can only represent run-length of either 1 or 2. Consequently, the above sequence of 16 symbols will be run-length coded to a compressed bitstream of 99 bits.

   Matlab programming for run-length coding on the input sequence:
   http://www.ece.uah.edu/~dwpan/course/ee614/data/coins_1d.mat
   Load the above mat file into Matlab,
   >> load ('coins_1d.mat')
   You can then see the variable 'J', which contains 73,800 symbols (uint8-type). Each of these symbols is represented by 8 bits.

   (a) Write a Matlab script, named "longest_run.m" to go through the sequence of symbols in variable 'J', in order to determine the symbol(s) with the longest run. Fill in the summary table on the next page with your answers to the following questions:
   Which symbol has the longest run?

What is the location (index) of the first such symbol in the run?
What is the largest run-length associated with such symbol?

(b) Write another Matlab script, named "rate.m" to apply run-length coding on the sequence in variable 'J', and output the number of bits (rate) of the overall compressed bitstream for four cases, where $B = 1, B = 2, B = 3, B = 4$, respectively. Fill in the summary table below with your answers.

Summary table for the results.
Note: Failure to fill in the tables below with your answers will result in points being deducted.

Symbol with the longest run	Index of the first symbol in the run	The largest run-length

Rate	$B = 1$	$B = 2$	$B = 3$	$B = 4$
Compressed Bitstream Size (in bits)				

(c) Comment on your results in terms of which value of B leads to the best compression of the input data sequence and why?

Hint: You might want to run the scripts on the example sequence with 16 symbols on the first page (copied below) to help make sure the scripts will work properly on much longer sequences. For this short sequence, you should get results below:
49 50 48 49 48 48 49 **50** **50** **50** **50** 49 49 48 50 50

Symbol with the longest run	Index of the first symbol in the run	The largest run-length
'50'	8 ('50' in red above)	4

Rate	$B = 1$	$B = 2$	$B = 3$	$B = 4$
Compressed Bitstream Size (in bits)	99	100	110	120