Exercise 1: V2V Codes for Black and White Document Scans (Part 1/2)

Analyze a structured V2V code for coding 300dpi black and white document scans

1 Write a program that reads all binary samples of a document scan into an array of bits (e.g., of type `vector<bool>` if you use C++)

The original document files are coded in the PBM format, which is a raw data format (see description on the right hand side).

The following files (found on the course web site) should be used as examples:

- "paper300dpi-page00.pbm"
- "paper300dpi-page01.pbm"
- "paper300dpi-page02.pbm"
- "paper300dpi-page03.pbm"

structure of "pbm" files:

```
P4                // ascii (fixed)
width height      // ascii
<binary data>     // binary
```

binary data:

- samples in raster-scan order (line by line)
- each sample is represented by one bit
  - ➡ bit 0 → white sample
  - ➡ bit 1 → black sample
- 8 bits are packet in one byte, where the first sample in scan order is placed in the most significant bit
- the first byte of the binary data contains the first 8 bits in scan order, etc.

Exercise 1: V2V Codes for Black and White Document Scans (Part 2/2)

2 Extend your program as follows:

Experimentally determine the probabilities for the symbol sequences of the two codes (block code and V2V code) shown on the right hand side.

3 Develop optimal codeword tables for both cases (using the Huffman algorithm).

You can do it on paper or implement it.

4 Calculate the average codeword length (per binary sample) for both developed codes.

Which code would yield a better compression efficiency?

block code	V2V code
0000	0000 0000 0000 000
0001	0000 0000 0000 001
0010	0000 0000 0000 01
0011	0000 0000 0000 1
0100	0000 0000 0001
0101	0000 0000 001
0110	0000 0000 01
0111	0000 0000 1
1000	0000 0001
1001	0000 001
1010	0000 01
1011	0000 1
1100	0001
1101	001
1110	01
1111	1

Exercise 2: Audio Coding using Rice Codes

Investigate lossless audio coding with Rice codes. Use the example file "audioData.raw" (from the course web site) for these investigations. The file consists of raw audio data in signed 8-bit format. That means, each byte of the file represents one sample and has to be interpreted as 8-bit signed integer.

1 Write an encoder and decoder for coding the audio data using Rice codes.

- Each sample x_n should be coded as:
  abs → Rice code for abs(x_n)
  if(abs > 0)
  sign → single bit indicating the sign
- The Rice parameter should be given as input to the encoder and written at the beginning of the bitstream (e.g., using a fixed-length code of 8 bits or a unary code).
- Check that the decoder decodes the file correctly.
- Try different Rice parameters and measure the size of the generated bitstream.

2 (Optional) Try to improve your lossless audio codec by coding the audio samples using chunks of 1024 successive samples.

- Determine the optimal Rice parameter for each chunk.
- Code the Rice parameter at the beginning of each chunk.

Exercise 3: Iterative Shannon-Fano-Elias Coding

Given is an IID source with the alphabet $\mathcal{A} = \{\text{ E, R, F }\}$ and the pmf

symbol	probability
E	$^5/_8$
R	$^2/_8$
F	$^1/_8$

1. Construct the Shannon-Fano-Elias codeword for the message "REFEREE" using the iterative encoding algorithm.
   - Use the prefix-free variant (only important at the end).
   - Assume that the symbols in the alphabet are ordered as: E, R, F.
2. Verify that the original message can be correctly decoded from the codeword using the iterative decoding algorithm.

Feel free to implement the encoding and decoding (instead of doing it on paper).