Compressing/Decompressing a file using Huffman codes

In this assignment, you are to compress a file using Huffman codes and decompress a file encoded using Huffman codes. The inputs of the program are the following:

1. The name of the input file
2. The name of the output file
3. Mode (0/1) : 0 for compression, 1 for decompression

Compression Mode:

In compression mode, the input file is the name of the file that needs to be compressed, and the output file is the name of the file that will be created during the compression. The input file contains a sequence of 8-bit characters. The output file stores the metadata for Huffman codes followed by the encoded sequence. The program also prints the Huffman codes for each character in the file in compression mode.

Decompression mode:

In decompression mode, the input file is the file's name that is created during the compression mode. The output file will be created during the decompression. The decompression algorithm first builds the tree containing Huffman codes using the metadata and then uses the tree to decode the encoded sequence in the input file. The decoded data is written to the output file.

Sample file.

A sample input file (in.txt) is provided for compression. Below are some sample inputs/outputs of your program.

Input

```
./a.out
Enter the name of the input file
in.txt
Enter the name of the output file
out.txt
Enter mode (0 for compression, 1 for decompression)
0
```

Output:

```
a -> 0
c -> 1 0 0
b -> 1 0 1
f -> 1 1 0 0
e -> 1 1 0 1
d -> 1 1 1
```

Generating out.txt

Input:
./a.out
Enter the name of the input file
out.txt
Enter the name of the output file
res.txt
Enter mode (0 for compression, 1 for decompression)
1

Output:
Generating res.txt

Validation:

You can run the following command to verify that the decompression is successful.
diff in.txt res.txt

The above command should not print anything.

Library functions:

To implement file I/O, you can look at the following library functions:
fopen
fclose
fprintf
fscanf
ftell
etc.

What to submit:

Implement everything in one C file that you need to submit. In addition, also submit a report that answers the following questions:

1. Are you using min-heap in your implementation?
2. What is the size of the compressed file after compressing the sample input file provided with the assignment? (Run “du -h out.txt” to obtain the size of the file out.txt).
3. Describe the format of the metadata that is stored in the compressed file.