Exercise 6 Huffman Algorithm

In this exercise, you are not making a program. Instead, your pretend that you are a computer and perform the exercise in your head writing the results and steps on paper.

There is input file **14_in.txt** that contains few lines of ASCII characters. Each line is terminated by pair of characters **\r\n**. Encode the input data using Huffman algorithm.

Encoding Example

Look inside the provided example files. They are

Input file example	**5_in.txt**
Debug output example	**5_debug.txt**
Encoded file example	**5_encoded.txt**
Decoded file example	**5_decoded.txt**

The output files were produced programmatically. The program encoded the input data and then decoded it back. To make it simple, the encoded data is presented as a sequence of characters ‘0’ and ‘1’ (real life programs use bit encoding).

Given the input file **5_in.txt** data

a aa
aab

ab c

abc

The debug output file **5_debug.txt** contains

```
Encoding 5_in.txt -> 5_encoded.txt     ← Informational message

countCharacters:       ← frequency of each character

10 \n {4}              ← ASCII code, character, and its frequency

13 \r {4}

32 ' ' {3}

97 'a' {7}

98 'b' {3}

99 'c' {2}

printQueue:            ← the same but sorted by frequencies

99 'c' {2}
32 ' ' {3}
```

98 'b' {3}

10 \n {4}

13 \r {4}

97 'a' {7}

```
makeBitData:          ← calculated encoding of each character

10 \n {101}           ← ASCII code, character, and its encoding
```

13 \r {00}

32 ' ' {011}

97 'a' {11}

98 'b' {100}

99 'c' {010}

```
Decoding 5_encoded.txt -> 5_decoded.txt ← Informational message

!!! Files 5_in.txt and 5_decoded.txt are equal. ← Success!!!
```

The encoded data (**5_encoded.txt**) is

110111111001011111100001011110001101101000101111000100010 1

The decoded data (**5_decoded.txt**) is the same as the input file data.

Steps to Perform

Do the following

- Read input file **14_in.txt**
- Calculate each character frequency (do not forget about **\r** and **\n** and **white space ' '**)
- Make the sorted frequency table (sort by frequency, if two frequencies are the same, sort by ASCII codes, in increasing order)
- Build and draw the corresponding Huffman tree
- Make the encoding table (each character encoding)
- Apply the encoding to the input data and produce encoded (compressed) data
- Write the encoded data as a sequence of 0s and 1s

While performing the exercise steps, write down their results on a piece of paper, similar to the example above. You have to include all intermediate results, the Huffman tree drawing, and the encoded result.

Scan the paper pages and submit produced image file to Exercise6 folder.

Future Considerations

You shall be able to decode your file from the encoded data using your Huffman tree. Think about it - you will need it in Lab 6.