Calculating The Compression Ratio

Write a program that prompts the user to:

1. Enter file name, then
2. displays the frequency table of the characters in the file, and
3. displays the Huffman code for each character, along with
4. the compression ratio.

In the Hoffman tree implementation, write a method that calculates the compression ratio of the Huffman Tree. Using the following formula:

$$\sum_{x \in \{a,b,\ldots,h\}} len(x) \cdot freq(x)$$

Where len(x) is the length in bits of the Huffman code for x and freq(x) is the frequency with which x occurs in the original file.

For example

Enter a text: abbcccdddd

ASCII Code	Character	Frequency	Code
97	a	1	110
98	b	2	111
99	c	3	10
100	d	4	0

(1)(3) + 2(3) + (3)(2) + (4)(1) = 19 bits

Normally ASCL characters use 8 bits so with a 10 letter word that would require 80 bits.

The compression ratio is $\left(1 - \frac{huffmanbits}{asclbits}\right) \cdot 100 = 77.5\%$

(1- Huffman bits / ascibits) * 100 = 77.5%

Compression ratios of 20% to 90% are typical, but not guaranteed.

Deliverables:

Your project should display characters of a string input, the frequency, the compressed Huffman code, and a display of the calculated compression ratio as a percent. You must use HuffmanCode.java for your implementation of the program.