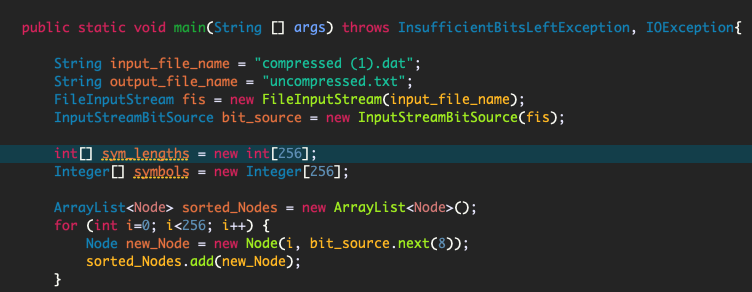
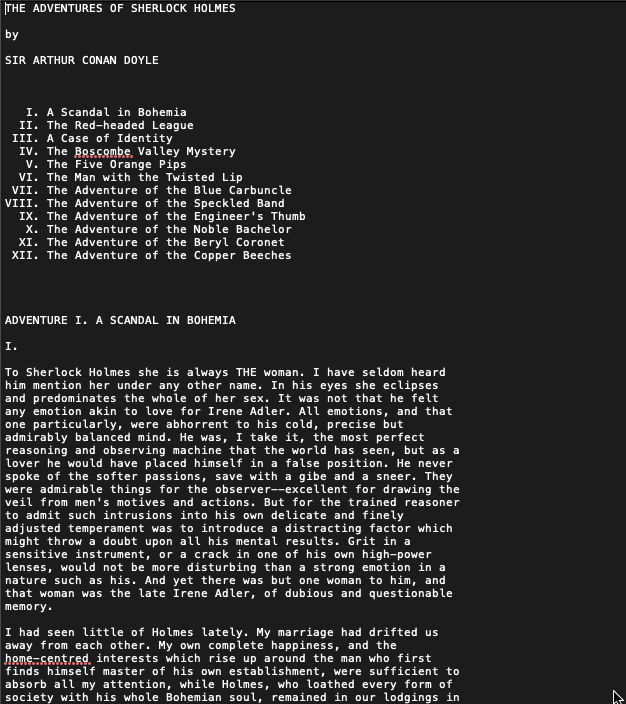
Scott Hofbauer – COMP590-A1

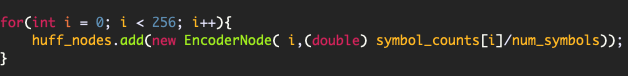
Part 3 – Entropy Calculation for the last part of the assignment, perform the following steps and answer the following questions.

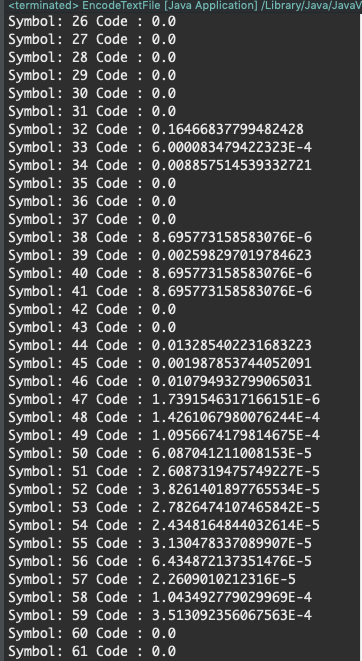
1. Use your decoder to decode the provided compressed file, producing ASCII English text.





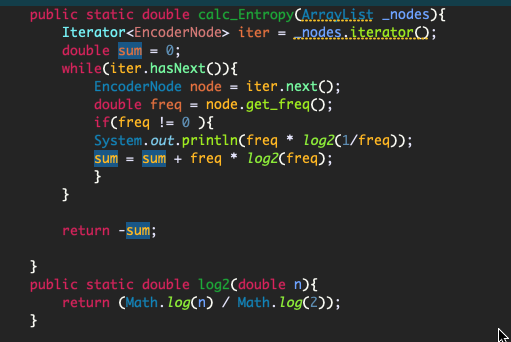
1. Calculate the probability of each symbol in the source ASCII English text file by dividing the number of occurrences for each symbol by the total symbol count.





This is just snapshot of the print out for each symbol and their frequency. One can see symbol 32 has the highest frequencies due to the fact that symbol 32 corresponds to a space is ASCII. This would logically make sense since in text file containing a book, the space value should appear the most.

1. Calculate the theoretical entropy of the source message in bits per symbol using the symbol probabilities from part 2. What is this value? 

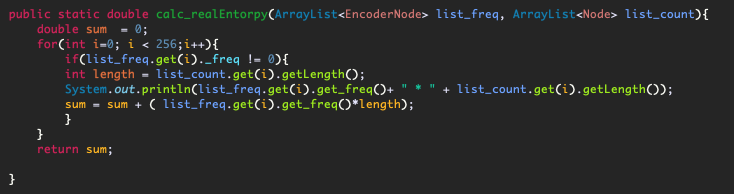




By computing the symbol frequencies and plugging them into the Entropy Formula, I used these two methods to calculate the Bits/Symbol (because there was no way I was doing this by hand) and got 4.53 bits/symbol.

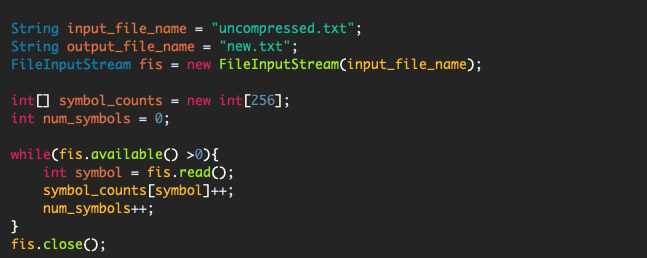
1. What is the compressed entropy achieved by the provided compressed file in bits per symbol? (Note: Do not include the overhead incurred by the 260 header bytes.)



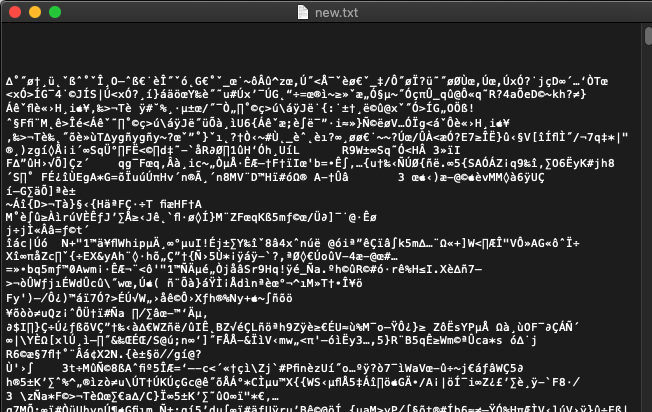


When calculating this bits/symbol we used the probabilities of each symbolled multiplied by the bit lengths of each symbol and summed them together to get a entropy value of:



1. Use your encoder to re-encode the raw English Text 

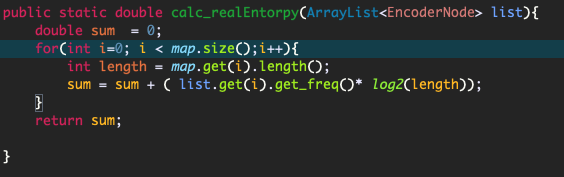
The uncompressed.txt file contains the raw English text and the encoder writes into new.txt.



This is the output of new.txt that contains the compressed version of the Sherlock Holmes book.

1. What is the compressed entropy achieved by the provided compressed file in bits per symbol? (Note: Do not include the overhead incurred by the 260 header bytes.





After running the method above given the formula we achieved **4.7 bits/Symbol** for compression. Which is close the theoretical limit of **4.53 bits/symbol** but better than the original compressed file.

1. Using your entropy value calculations as evidence, does your encoder achieve better compression or worse compression then the original compressed file?

The compression algorithm I wrote did achieve a better bits/symbol value of 4.7 versus 4.86. This can also be seen in the finder were it the total size of the files differ by approximately 20 kilobytes.

