Joshua Pollock

CS 200 Project 3

February 22, 2017

Joshua Pollock

CS 200

Project 3 – RLE Compression

February 22, 2017

**Purpose:**

The purpose of this project is to create a C or C++ program that can compress and decompress files using Run Length Encoding (RLE). This program will be able to take in a file and compress or decompress that file based off of the filename. The filename ending with “.rle” will be sent to decompress while a regular filename will be sent to compress. If the filename is “file.rle” the file will then have an output file named “filename.plain”. We are given a skeleton program that takes in a file and outputs the file as “.rle” or “.plain” depending on what is put into it. Our job is to add on to this file and have it perform RLE encryption.

**Research:**

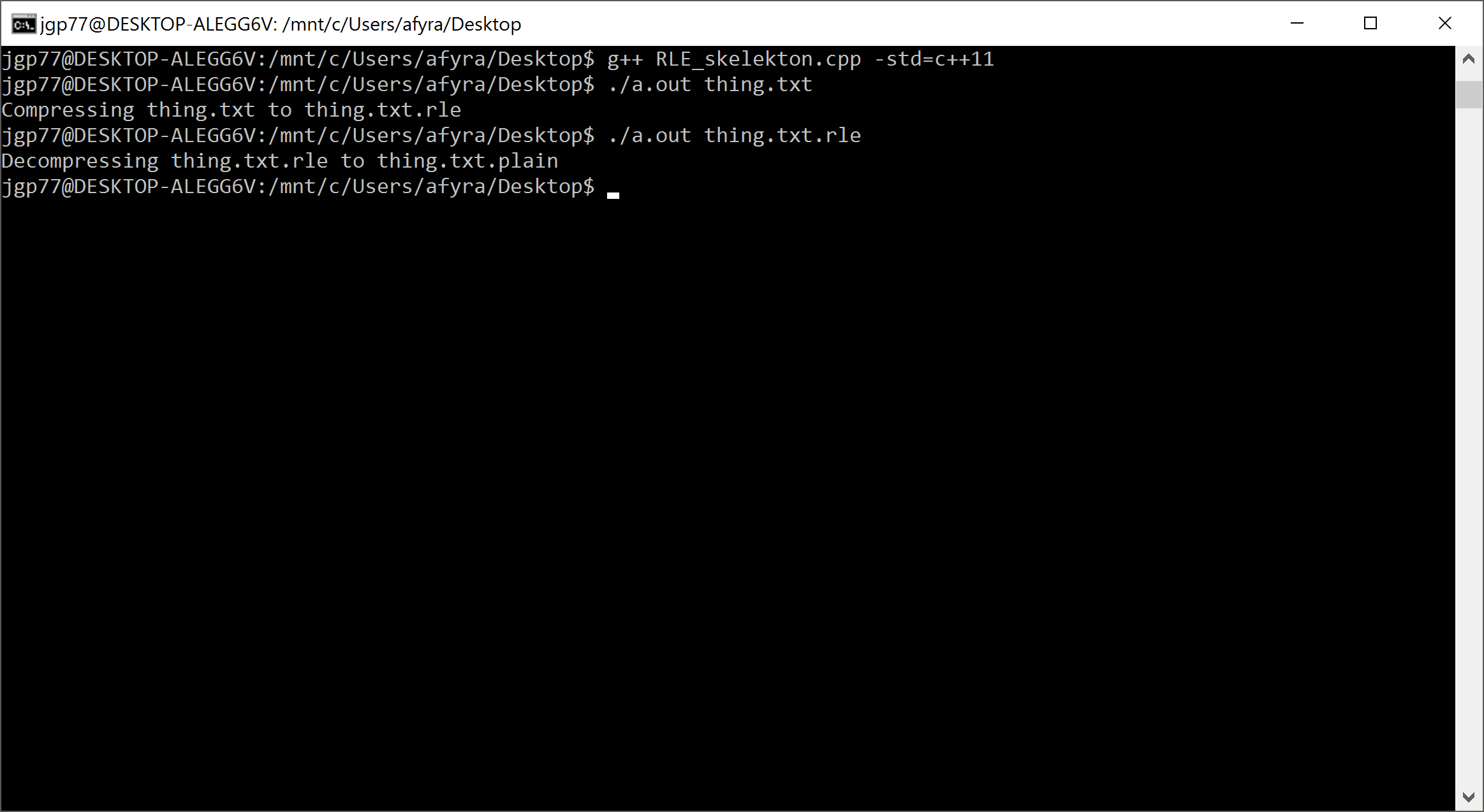
RLE encoding seemed quite difficult at first. Reading the given example did do a fine job at explaining what RLE encoding was, but it didn’t make the project seem any less difficult. One of my friends in the class, Jacob Kaufman, helped to explain what this lab was. He basically said that it was a program that uses a ton of loops to count the amount of characters in a text file. This explanation cleared up what I needed to do to create the program. I did look into the Wikipedia article about RLE encoding and browsed various sites to get the gist behind RLE encoding.  
<https://en.wikipedia.org/wiki/Run-length_encoding>

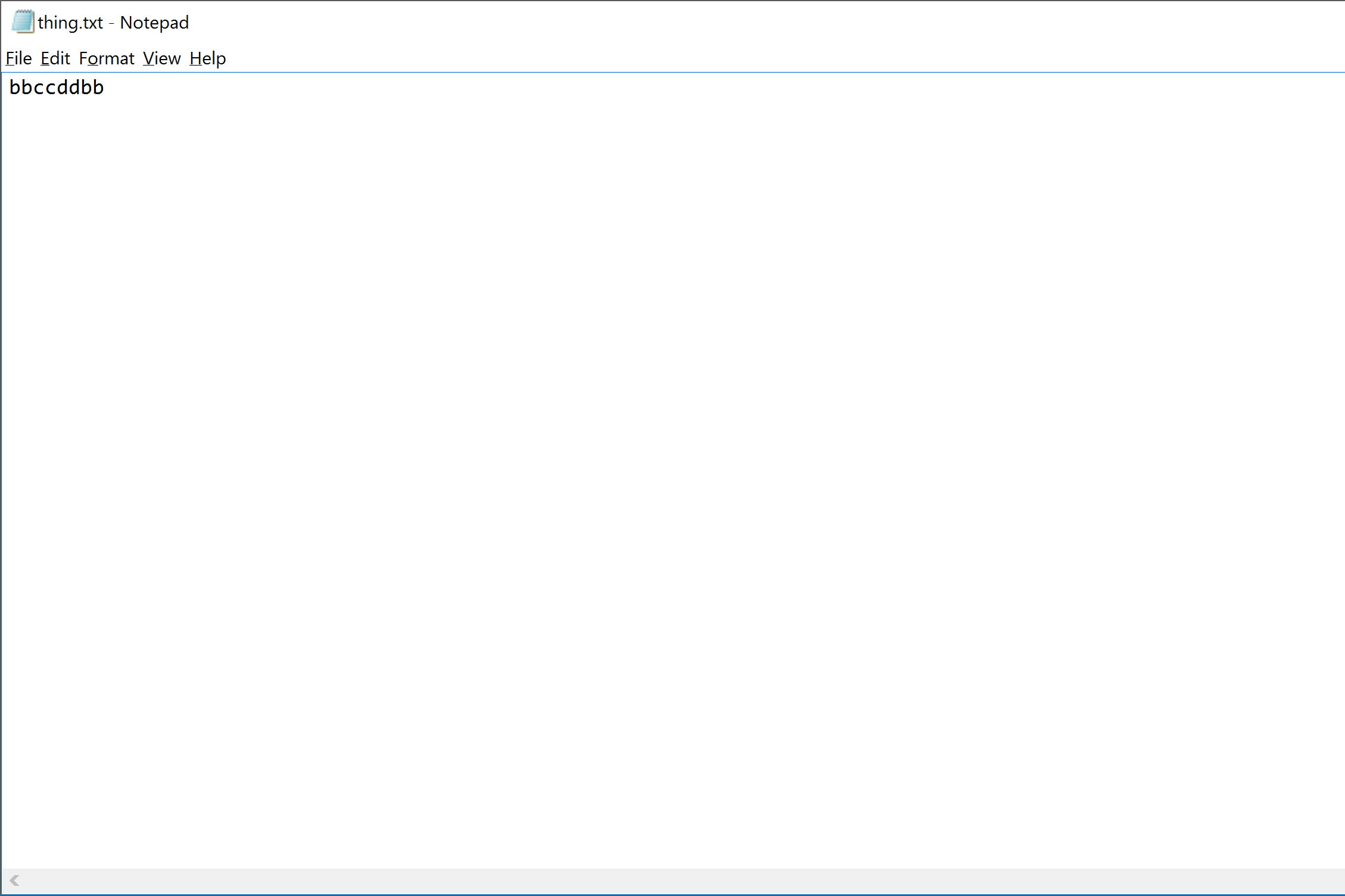
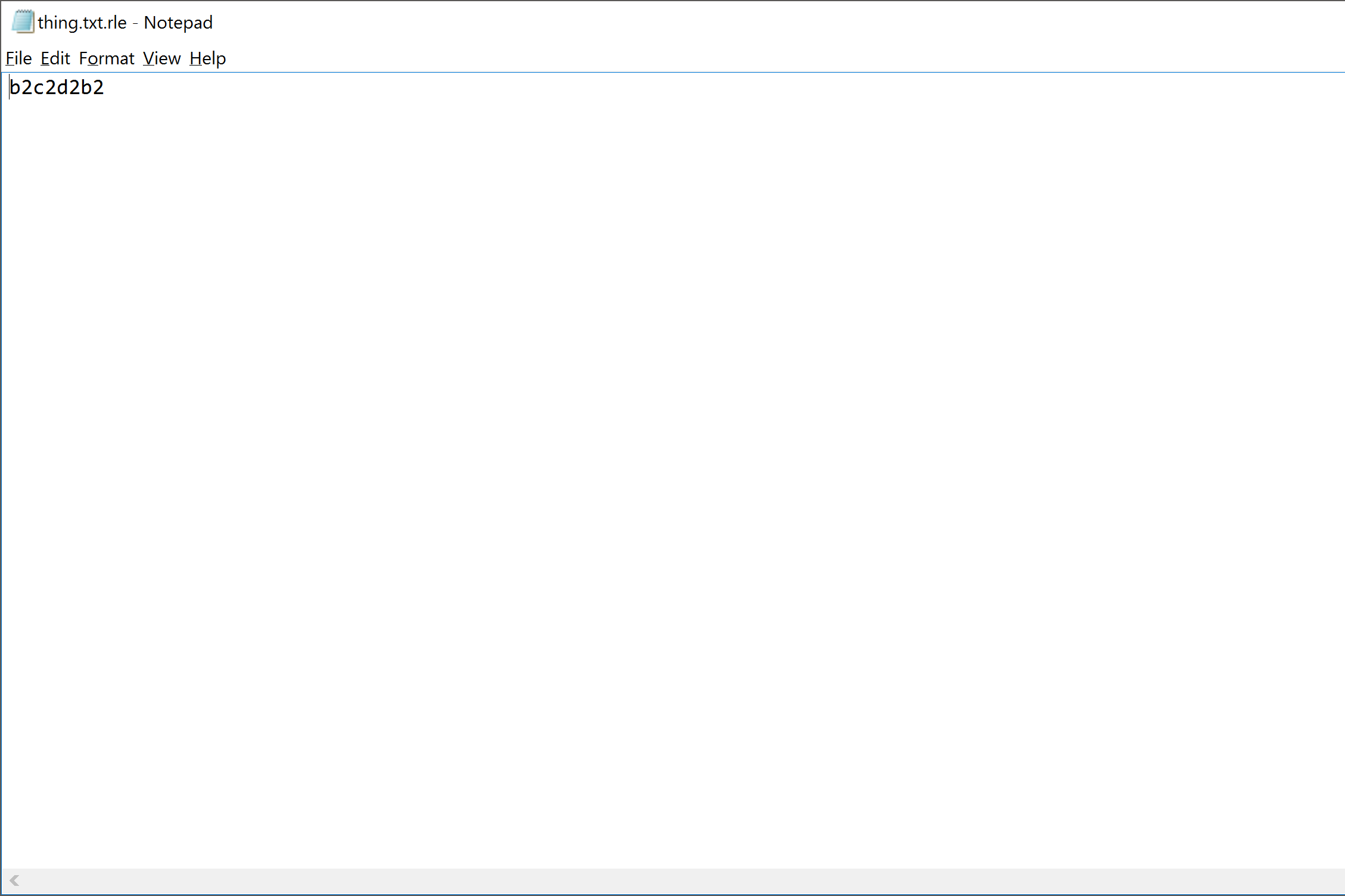
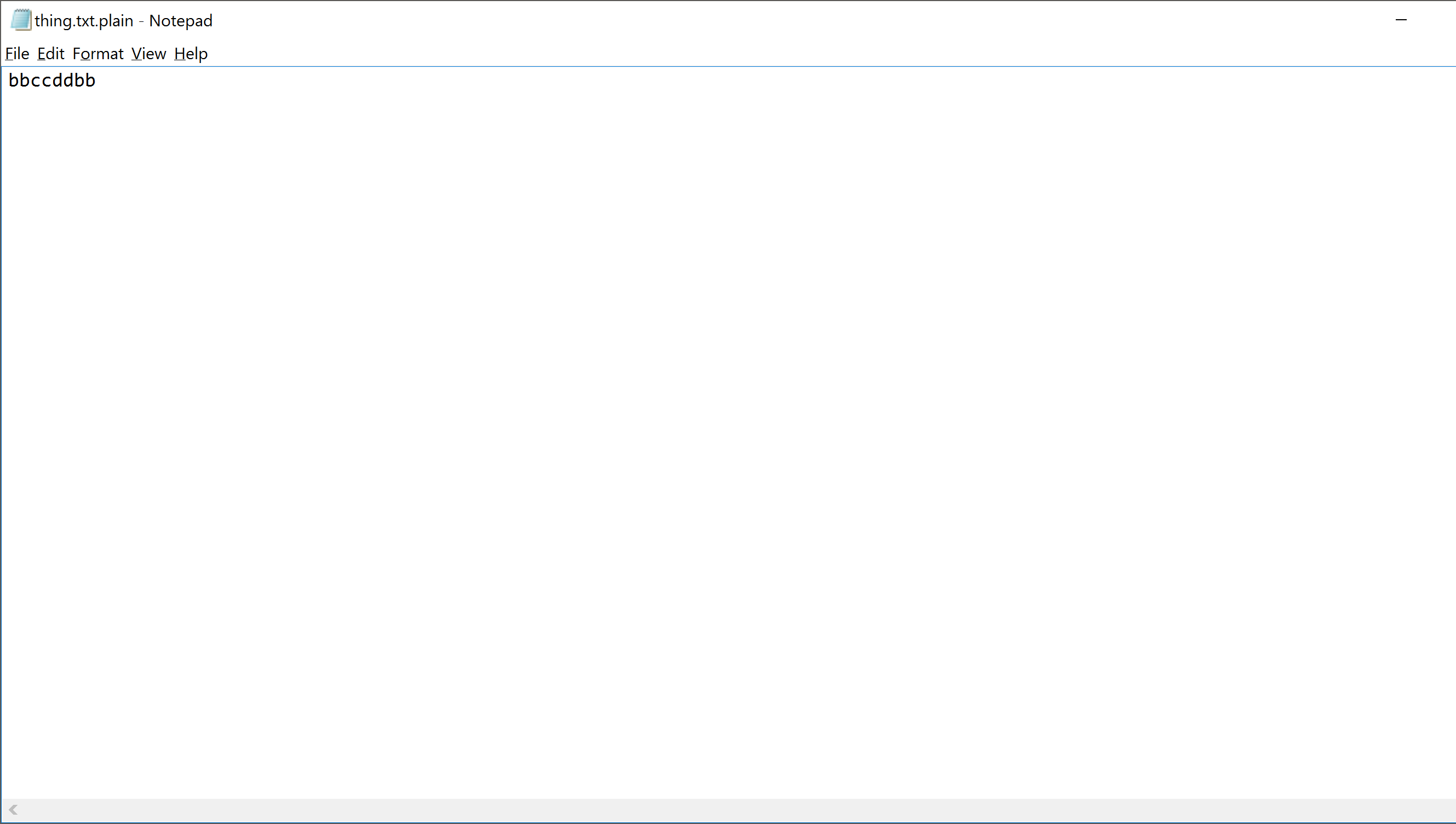
<https://www.prepressure.com/library/compression-algorithm/rle>

<https://www.youtube.com/watch?v=ypdNscvym_E>

After browsing I believed I was ready to attempt the project and started to code using the skeleton.

**Program:**

Bellow will be screen shots of the code, text file, .rle file, .plain file, and compilation/execution in the bash terminal. The code requires one arguemnt to be executed properly. Without this arguemnt I found that the code would not compile and would throw errors. By adding -std=c++11 the file compiled normally.   

**Conclusion:**

Overall, this project was interesting. I do not have much experience in C++, but I did program a bit in a language called RobotC in high school. I overcomplicated this project at first and avoided doing it as it seemed difficult. This however was not the case. The project was actually quite simple once I dumbed it down with Jacob after the class. The version of Bash I have on windows is a bit buggy and whenever I compiled the .cpp file I had to use the extension -std=c++11. Without this the code, will not compile on my machine. The compression is also a bit buggy. It is unable to use numbers in the compression and gets confused if you create multiple lines. With my current knowledge of C++, I am unable to fix these bugs. Despite this, the file I named thing.txt that contained “bbccddbb”, worked flawlessly. I am quite happy with this solution and have expanded my knowledge of C++. One of the neat things I learned from this project is that Windows 10 can install an Unbuntu terminal/Bash. This helped quite a bit in my troubleshooting of the code. The code turned out to just be many different loops and was not too difficult. I am quite happy with the solution and do not see a need to improve it currently.