3460:435/535 Algorithms - Project 2

This programming project is to learn and implement a variant of the compression algorithm LZW. Your tasks for this assignment are to (1) run LZW as it is and (2) modify LZW to incorporate the advantages of variable length coding algorithms as illustrated in Huffman coding. You must use C++. If you insists on using Python, you will be on your own but make sure you follow the same requirement.

Part I: LZW using code length 12

- Download https://www.cs.uakron.edu/~duan/classes/435/projects/project2_LZW/lzw435.cpp or the C++ LZW directly from http://rosettacode.org/wiki/LZW_compression#C.2B.2B and run the program;
- Understand how LZW works and how it's implemented;
- Modify the program to clean any unnecessary parts and be able to compress any file you are given;

Once we compile the program, we should be able to run it using: lzw435 c "filename" to compress the file with filename "filename". You program should save the compressed file as: "filename.lzw". And we should be able to run it using: "lzw435 e filename.lzw" to expand the compressed file "filename.lzw". You program should save the expanded file as: "filename2". You know your filename2 should be identical to "filename". Use "diff" to double check.

Part II: Modified LZW to allow the length of LZW codes to increase gradually.

- Your algorithm will increase the length of the code words from 9 to 16 bits.
- Do nothing when the length reaches 16 bits.

Name your modified LZW program "lzw435M.cpp". Once we compile the program, we should be able to run it using: "lzw435M.exe c filename" to compress the file "filename". You program should save the compressed file as: "filename.lzw2". And we should be able to run it using: "lzw435M.exe e filename.lzw2" to expand the compressed file "filename.lzw2". You program should save the expanded file as: "filename.2M". You know your file named "filename.2M" should be identical to that of name "filename".

Pay special attention to synchronize the compression and expansion to make sure they work correctly. Test your program thoroughly.

Graduate/honors students: A short description (1page) of the pros and cons of LZW vs Huffman coding.

What to submit.

1. Grade your own project! Fill the **grading sheet** and submit the filled sheet.
2. Submit your source code. Make sure to test your code on more than one set of data. DO NOT submit programs that are not *reasonably correct*! To be considered *reasonably correct*, a program must be completely documented and work correctly for sample data provided with the assignment. You must submit an electronic copy of the program using your Brightspace dropbox. Follow these steps:
   a) Create a folder named jsmith_2 (but use your first initial and last name).
   b) Place the two source files (lzw435.cpp and lzw435M.cpp) inside the folder.
   c) Right-click on the folder and choose Send To... Compressed Folder (or use some other Zip utility to archive the entire folder). The goal is to create a zip archive named jsmith_2.zip (your initial and name) that contains the folder which contains the source files for your project.
   d) Drop this single zipped file to the drop box.

Be sure to electronically submit your working solution before the due date! Do not submit non-working programs. The electronic submission time will be used to assess late penalties (if applicable).

Grading. Your code will be graded on **correctness**, efficiency, clarity, and elegance.