

CHALLENGE 2: DATA COMPRESSION

1 Preliminaries

1.1 Research

Data compression is the science of representing information in a compact form. It has been one of the critical enabling technologies for the on-going digital multimedia revolution for decades. We will discuss about two popular sequence compression algorithms: **Static Huffman Encoding** and **Lempel-Ziv-Welch (LZW)**. See the following [lecture notes](#) and fulfill the given requirements below:

1. State the **static Huffman** and **LZW** compression algorithms:
 - Both encode phase and decode phase must be describe carefully.
 - Each phase should be break into smaller steps.
 - Related data structures involved must be mentioned and told how they are represented and stored.
 - There should be example attached during the statement.
2. Indicate advantages and disadvantages of each algorithm.
3. Outline practical applications or upgraded versions or the 2 given algorithms.

1.2 Programming

- You are requested to build a 2-way program for Encoding and Decoding text files using the LZW compression algorithm.
- Your program must be built into an execution `".exe"` file that runs within the following command line arguments:

`A.exe Action InputPath OutputPath OutputInfo`

,in which:

- * **A.exe**: Your executional ".exe" file.
- * **Action**:
 - **-e**: encode phase. Take the input as a text file (.txt) and output as a binary file (.lzw)
 - **-d**: decode phase. Take the input as a binary file (.lzw) and output as a text file (.txt).
- * **InputPath**: Path to the input file.
- * **OutputPath**: Path to the output file.
- * **OutputInfo**: The additional information during the action process, should be shown on the console screen.
 - **-i**: size of the input file, size of the output file, space saved percentage.
 - **-d**: the LZW dictionary created during encode/decode process.
 - **-ind**: both options above.

Example:

Text file	Compressed file	-i	-d
Text file: "Data.txt"	Binary file: "Compressed.lzw"	Input size: bits Output size: bits Space saved: %	new word 1: 256 new word 2: 257 ...
WYS*WYGWYS*WYSWYSG	87 89 83 42 256 71 256 258 262 262 71	Input size: 160 bits Output size: 120 bits Space saved: 25%	WY: 256 YS: 257 S*: 258 *W: 259 WYG: 260 GW: 261 WYS: 262 S*W: 263 WYSW: 264 WYSG: 265

2 Group registration and Submission regulations

2.1 Group Registration

- This challenge requires a group of 3 - 4 students. Group members for each challenge must be different (i.e, Any pairs of students are at max ONE same group).
- Group ID is generated by concatenating the last 2 digits of each member's Student ID in ascending order.

Example:

– Given the student codes: *19127666 - 19127888 - 19127991 - 19127999*.

→ **Generated ID:** *66889199*.

– Given the student codes: *19127667 - 19127889 - 19127990*

→ **Generated ID:** *678990*.

- Group registration will be provided within the attached [link](#). **Each group's member** must fill in the registration form.

Note: Group registration and file submission time should not be different more than *15mins*.

2.2 Submission regulations

- Only 10 first submissions is accepted.
- The submission file must be in the following format: [**Group_ID.rar**] or [**Group_ID.zip**], is the compression of the [**Group_ID**] folder. This folder contains:
 - The report file must be presented as a document [**Group_ID.pdf**] or as a slideshow [**Group_ID.pptx**]. This file presented research answers from **1.1** and the information of code fragment (data structures, algorithms, functions) from **1.2**.
 - * If your submission is a slideshow, there must be explanation in the *Note* part of each slide.

- * Information (Names, Student IDs) must be declared clearly on the first page (or first slide) of your report. Your working progress (Which option did you choose? How much work have you completed?) should be demonstrated on this page, too.
 - * The report file should be **structured, logical, clear** and **coherent**. The length of the submission should not exceed 15 pages for the document file, and 30 pages for the presentation slide.
 - * All links and books related to your submission must be referenced.
 - The programming file must be a single file [**Group_ID.cpp**]. The code fragment must be clear, logical and commented.
- Submission with wrong regulation will result in a "0" (zero).
 - Submission without registration will result in a "0" (zero) for ALL group members.
 - Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board of the CLC program for further action.