COLLEGE OF ENGINEERING & TECHNOLOGY

Department : Computer Engineering

Lecturer : Dr. NourElDin S. Eissa

Course Name : Data Structures & Algorithms

Course Code : ECE2103

TA : Eng. Magd Nemeh

Project Briefing
Huffman Text File Compressor

Contents

This document contains the briefing for the ECE2103 Data Structures & Algorithms course project for the academic term Fall 2025/2026. Read it carefully.

Description

Using what you studied in the ECE2103 course, develop an application that can compress and decompress ASCII-based text files, using Huffman lossless encoding.

Requirements

1) Read a user-specified text file and generate the frequency table based on the read data.
    a. The application must be able to read a user specified file.
    b. The input file has a maximum length of **4.00GB**.
    c. The input file is ASCII based.
2) Implement a suitable data structure (e.g., heaps) to dynamically sort the frequency table.
3) Design the required binary tree data structure that serves as a base for the Huffman tree.
4) Build the Huffman tree.
    a. The Huffman tree must be based on the tree designed in step 3.
5) Generate the variable-length codes from the Huffman tree and store them in a separate file.
    a. The generated codes must be displayed on the screen.
    b. The generated codes must be saved to a separate file, with the same filename as the input but with an extension of ".cod".
    c. This file will be used during the decompression part, to be able to decompress any file later on.
6) Compress the input file using the codes generated from the Huffman tree.
    a. The generated output must have the same filename as the input but with the extension ".com".
    b. The output file should not have a size larger than the input file.
7) Decompress any previously compressed file using the code file [from step 5].
    a. The decompressed file must match the original uncompressed file.
    b. You must use the code file generated in step 5b to retrieve the codes required for the decompression process.

Guidelines

1) An exact number of **2 students** must work on the project. Each team member will be responsible of certain parts of the project. Once assigned, individuals cannot change their respective part in the project.
2) An oral discussion session may be held after the submission of the project. In case you are subjected to further discussion after the submission of your project, it will be considered as part of your evaluation process.
3) Any submission that contains syntax errors will be rejected and a grade of **ZERO** will be assigned for the project.
4) All applications must be developed using **Microsoft Visual Studio (>=2015)** and must run on **Microsoft Windows operating systems.**
5) Parts of the project depend on the correct implementation of earlier parts. Any **missing parts of the project will negatively affect the marks assigned to the project**. Choose your team members and manage your timing carefully.

Academic Integrity Policy

- Plagiarism results in **ZERO** and **possible disciplinary action**.
- Plagiarism includes:
  - Copying another student's code.
  - Copying online sources or GitHub, even with modifications.
  - Self-plagiarism (submitting your own previous work).
  - Allowing others to copy your code.
  - Using AI to generate full or partial code.
  - Submitting code that you cannot explain.

AI usage policy:
You may use AI tools to understand concepts, but you may not submit AI-generated code.

Bonus Task

Recursive Compression (5 Marks)

Re-apply Huffman compression up to 5 times until no more gain is observed in the compressed file. Embed the number of compressions rounds in the (.cod) file, or a separate metadata file. This process must be done automatically during compression and decompression. The users will not repeatedly compress the compressed output by themselves. The task must be fully automated inside your application.

Submission Deadline

- The submission of your project will take place on Moodle.
- The discussion sessions will be held either online or physically on campus.
- The deadline for submitting the project is on the **30th of December, 2025.**
- The deadline for submitting the teams' arrangements is on the **15th of December, 2025.**