BIRZEIT UNIVERSITY

ENCS3130 – Linux Lab
Spring Semester 2022/2023
Project#1 – Shell Scripting
A Simple Dictionary-based Compression and Decompression

Deadline: August 25, 2023

Project Overview

In this project, you are required to implement a simple dictionary-based compression and decompression tool in shell scripting.

The dictionary-based compression is a lossless compression technique that relies on finding patterns in data. In the compressed file, a shorter code is substituted for the pattern. In this project, and for simplicity, we will have (assume) the following:

1. A unique binary code is assigned to each word in the uncompressed file. Thus, the compressed file will consist of binary codes only
2. The code size is 16-bit. This will allow us to encode up to 65,536 unique words
3. The uncompressed file is a text file that contains Unicode characters, i.e., each character is 16-bit
4. Your compression/decompression tool will have a dictionary stored in a text file called **dictionary.txt**. The binary code of the values of the dictionary starts from 0x0000.
5. Initially, assume the dictionary is empty. The dictionary is filled over time when more and more compression operations are performed.
6. The tool is case-sensitive.
7. Special characters, such as, spaces, punctuation, $, #, etc. are treaded as words, and hence each one will have a code in the dictionary

Example:

Assume you are asked to compress the following text. Assume the dictionary is empty, i.e., this is the first time the tool is used.

"We are studding computer architecture. Computer architecture is an important course in computer engineering. In this course, we are studying many useful topics."

The dictionary will look like that:

Code	Word
0x0000	We
0x0001	Space ‘ ’
0x0002	are
0x0003	studying
0x0004	computer
0x0005	architecture
0x0006	.
0x0007	Computer
0x0008	is
0x0009	an
0x000A	important
0x000B	course
0x000C	in
0x000D	engineering
0x000E	In
0x000F	this
0x0010	,
0x0011	we
0x0012	many
0x0013	useful
0x0014	topics
0x0015	\n

In this example, the compressed file will look like this

0x0000
0x0001
0x0002
0x0001
0x0003
0x0001
0x0004
0x0001
0x0005
0x0006

0x0001
0x0007
0x0001
0x0005
0x0001
0x0008
0x0001
0x0009
0x0001
0x000A
0x0001
0x000B
0x0001
0x000C
0x0001
0x0004
0x0015
0x000D
0x0006
0x0001
0x000E
0x0001
0x000F
0x0001
0x000B
0x0010
0x0001
0x0011
0x0001
0x0002
0x0001
0x0003
0x0001
0x0012
0x0001
0x0013
0x0001
0x0014
0x0006

The uncompressed file size = Number of characters' x 16 (size of the Unicode)

= 160 x 16 = 2560 bits = 320 bytes

The compressed file size = Number of codes x 16 (code size)

= 49 * 16 = 784 bits = 98 bytes

File Compression Ratio = uncompressed file size / compressed file size

= 2560 / 784 = **3.265**

Program Menu (Program usage flow):

1. The program asks the user if the dictionary.txt file exist or not
2. If yes, the program asks the user to enter the path of dictionary.txt, read this path, and load the dictionary into the appropriate data structure.
3. If no, the program creates a new empty dictionary.txt
4. The program asks the user whether he or she wants to do compression or decompression
   - c, compress, or compression means compression.
   - d, decompress, decompression means decompression
   - The options above are case-insensitive
   - Any other option, the program prints the appropriate error message
5. In the case of compression,
   - The program asks the user to enter the path of the file to be compressed
   - The program reads the file and compresses it by substituting the appropriate codes from the dictionary
   - If the program encounters a word in the input file that does not exist in the dictionary, then the program appends it to the dictionary and uses its new code in the compression operation
   - The program computes the compression ratio and prints it on the screen
   - The program writes the compressed data in the compressed file
   - The program saves changes on the dictionary
6. In the case of decompression,
   - The program asks the user to enter the path of the file to be decompressed
   - If the file has codes that do not exist in the dictionary, an appropriate error message is displayed
   - The program decompresses the file by substituting the correct words from the dictionary
   - The program writes the decompressed data in the uncompressed file

Teamwork:

You can work on this project in teams of up to two students only

Submission

You need to submit the complete Shell scripting code as a reply to this message.

Grading Criteria

Criteria	**Grade**
Code Structure, Organization, and Documentation	10%
Program Running	25%
Reading/Writing from/to Text Files	10%
Maintaining Dictionary	15%
Performing Compression	25%
Performing Decompression	25%
Calculating Compression Ratio	10%
Discussion	30%
Total	**150%**