Information Theory
Assignment 2: Lempel-Ziv Coding

Deadline: 06.10.19 23:59
Output: ZIP-file with the code and PDF-report has to be uploaded to Moodle, input files are NOT needed. Name of the archive has to be *NameSurname.zip* (For example, *IvanIvanov.zip*). Name of the single source code has to be *NameSurname.py* (For example, *IvanIvanov.py*). Name of the report has to be *NameSurname.pdf* (For example, *IvanIvanov.pdf*). No other symbols allowed
Programming language: Python 3.7
Requests:

- The program must work, the code should be readable, well-structured and should contain English comments
- NO extension of a deadline. Works sent after the deadline will NOT be evaluated
- Assignment is strictly individual
- Deviations from assignment requirements will lead to *penalties* or even *annulation* of the assignment grade
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies.

Evaluation criteria:
0 (0%) - no submission or late submission
1 (20%) - required functionality is not achieved
2 (40%) - required functionality is lower than 50%
3 (60%) - required functionality is 50-80% and/or shortcomings in report
4 (80%) - required functionality is 80-100% and/or shortcomings in report
5 (100%) - well-structured readable correct code with English comments and correct report
Task:
Write a program that compresses and decompress target files using Lempel-Ziv Coding algorithm (LZ78 also called LZ2), so you need to implement both coding and decoding. Write a report with your experiment results and conclusions.
Inputs:
Dataset prepared for the Assignment 1 should also be used for this assignment without any changes.
The link https://drive.google.com/drive/folders/1ALUvTrk-PCueXvGS2Z5IIsmkLzBhtLFp
Your source code and *"dataset/"* directory should be in the same location.

Outputs:

In the same directory with your source code after reading the inputs your program should create a directory *"NameSurnameOutputs/"* (For example, *IvanIvanovOutputs/*) inside of which all subdirectories of *"dataset/"* should be created. Each output subdirectory should contain two files for each original file from input subdirectory with the same name:

1. Compressed file (For example, for *original.** file output should be called *originalCompressed.**)
2. Decompressed file (For example, for *original.** file output should be called *originalDecompressed.**)

Compression representation:

On each step, algorithm should provide a binary code for the tree node (i, x_n), where

- i - is ancestor, which should be represented by the sequence of bytes (value of each byte's first bit should indicate if it is the last byte (1) for i or not (0)) using Big Endian notation
- x_n - is the symbol added to an ancestor on step n (n >= 1), which should be represented by the single byte using Big Endian notation

Examples:

(128,a) will be represented as 00000001 10000000 00000000, where a - is 0

(100,b) will be represented as 11100100 00000001, where b - is 1

Report:

As an experiment compresses files from Assignment 1 dataset, provide your results and conclusions in the report and answer the following questions:

1. Does the compression ratio depends on the file type? If yes, then how?
2. Does the compression ratio depends on file size? (i.e. large text file vs small one, just text *.doc vs *.doc with with additional formatting). Explain your answers

Details:

You will need to upload your program to the auto-checker system and it will have to pass all tests. The link to auto-checker will be given later