CS102 – Algorithms and Programming II
Programming Assignment 2
Spring 2023

ATTENTION:

- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:
  CS102_Sec1_Asgn2_YourSurname_YourName.zip
- Replace the variables "Sec1", "YourSurname" and "YourName" with your actual section, surname and name.
- You may ask questions on Moodle and during the scheduled Zoom meeting.
- Upload the above zip file to Moodle by the deadline (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TAs and tutors during the scheduled Zoom meeting.

GRADING WARNING:

- Please read the grading criteria provided on Moodle. The work must be done individually. Code sharing is strictly forbidden. We are using sophisticated tools to check the code similarities. The Honor Code specifies what you can and cannot do. Breaking the rules will result in disciplinary action.

Reducing Document Size

In this assignment, you are going to implement a Java program that reduces document size by replacing words with integers. You should have a method for reading and processing the input text file ("input.txt") to generate two outputs:

- The first output ("map.txt") should include which integer corresponds to which word.
- The second output ("encoded.txt") should include the document where each word is replaced by an integer based on your mapping.

Input text file is guaranteed to have all lowercase letters and no punctuation; however, you should preserve the line endings in your output. A sample input and corresponding outputs are as follows:

Sample Input/Output for the Process

input.txt	*map.txt*	*map.txt (continued)*	*encoded.txt*
the red bike was on the road i kept riding the bike near the black road the bike was black in the end because of the road such a road and such a bike	0: the 1: red 2: bike 3: was 4: on 5: road 6: i 7: kept 8: riding 9: near	10: black 11: in 12: end 13: because 14: of 15: such 16: a 17: and	0 1 2 3 4 0 5 6 7 8 0 2 9 0 10 5 0 2 3 10 11 0 12 12 14 0 5 15 16 5 17 15 16 2

You should map the same words to the same integer values; for example, all occurrences of the word “the” is mapped to the integer 0 in the sample above. Assign the integer values to the words based on their first occurrences. For this purpose, you may keep an ArrayList of words and add new words to the list if they are not already included. In the end, indexes of the words in your ArrayList would give you the mapping.

Check if your encoded files are smaller in size. Include a second strategy for creating the word mapping. Here you are free to use any approach, some suggestions include:

- Assigning random non-overlapping integers to each word.
- Assigning smaller integers to the words with high occurrence frequency.
- Assigning smaller integers to the longer words.

Compare if the second strategy is better or worse. Keep both strategies in your final implementation as different methods.

You should also have a method for the reverse operation. This reverse operation should receive “map.txt” and “encoded.txt” to produce a readable document (“decoded.txt”). Each line of “map.txt” includes the integer followed by **‘:’** character followed by the corresponding word. The reverse operation should first read “map.txt” to create an array of words. Note that the integers in “map.txt” may not be in order, so you should position each word in the corresponding index. You may first find out how many words are included in “map.txt” to construct your array accordingly. Then, read “encoded.txt” and replace each integer with the corresponding word. Do not forget to preserve the space and end line characters.

Sample Input/Output for the Reverse Process

map.txt	*map.txt (continued)*	*encoded.txt*	*decoded.txt*
11: in 0: the 1: red 16: a 17: and 8: riding 9: near 6: i 7: kept	10: black 12: end 13: because 14: of 15: such 2: bike 3: was 4: on 5: road	0 1 2 3 4 0 5 6 7 8 0 2 9 0 10 5 0 2 3 10 11 0 12 12 14 0 5 15 16 5 17 15 16 2	the red bike was on the road i kept riding the bike near the black road the bike was black in the end because of the road such a road and such a bike

Preliminary Submission: You will submit an early version of your solution before the final submission. This version should at least include the following:

- The required functionality to read and create “map.txt” and “encoded.txt” should be complete using the initial approach (word indexes are determined based on first occurrence). Do not forget to test your method with sample files.

You will have time to complete your solution after you submit your preliminary solution. You can consult the TAs and tutors during the scheduled Zoom meeting. Do not forget to make your final submission at the end. Even if you finish the assignment in the preliminary submission, you should submit for the final submission on Moodle.

Not completing the preliminary submission on time results in 50% reduction of this assignment's final grade.