E0 251: Programming Assignment 3
Due before 11.59 PM on 03-03-2022
INDEX

For the purposes of this assignment, we define an INDEX as an alphabetical order sorted index of the unique WORDs occuring in a text file. The INDEX entry for each WORD contains a list of the line numbers on which that WORD occurs (at least once) in the text file. In this assignment you will be writing a program to generate an INDEX for an input text file using a Binary Search Tree to store WORD information accumulated from the text file.

We define a WORD as a case blind string of alphabets (from the 26 character set {a, b, c, ..., z}) separated from other WORDs by white space, numbers, punctuation marks or other printable characters. However, if 2 WORDs are separated from each other by only a hyphen, ignore the hyphen (e.g., treat “co-advisor” as the WORD “coadvisor”). Do not include 1 and 2 alphabet WORDs in the INDEX. Truncate WORDs that are longer than 20 alphabets to their first 20 alphabets.

(a) Your program should be written to take its input text from a text file specified by the only command line argument.
(b) The program should accumulate line number information for the different WORDs occurring in the input text file in a Binary Search Tree implemented using the basic Insert and Delete algorithms provided in the lecture slides with one variation – use Option 1 rather than Option 2 in handling the “Both LST and RST” case.
(c) Your program should not use any library functions other than those provided by <stdlib.h> and <string.h>.
(d) After processing the entire input text file, delete from the Binary Search Tree all WORD entries with line occurrence counts less than 3.
(e) The INDEX should be output to standard output (stdout) with the INDEX entry for each WORD on a separate output line. The WORD entries should be output in alphabetical order. The entry for a given WORD should contain the WORD followed by a space and a comma separated sorted list of the unique line numbers on which the WORD occurs in the input text file.

Submit a single achive file containing all of the source code that you wrote to implement the program as well as a Makefile if necessary. Name the archive file using the convention FirstName_LastName_Assignment3.zip (or .tar, .tgz). e.g., Matthew_Jacob_Assignment3.tgz

Example:
For input of the following 13 line text extracted from https://www.oatridge.co.uk/poems/d/dr-seuss-cat-in-the-hat.php):
and then something went BUMP!
how that bump made us jump!
we looked!
then we saw him step in on the mat!
we looked!
and we saw him!
the cat in the hat!
and he said to us,
'why do you sit there like that?'
'i know it is wet
and the sun is not sunny.
but we can have
lots of good fun that is funny!'

Output:
and 1,6,8,11
that 2,9,13
the 4,7,11