BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Second Semester 2014-2015

CS F211 Data Structures and Algorithms

Lab Sheet – 9

General Instructions for Programming

1. All inputs to the program must be either (a) command line arguments (b) or read from a file (other than stdin). DO NOT READ anything from stdin and DO NOT USE PROMPTS like "Please enter a number …".

2. You are required to write the output to a file (other than stdout) and errors if any to a different output channel (stderr or another file).

3. Use standard C coding conventions for multi-file programs. Separate the following: interfaces of functions (use a ".h" file), data type definitions (use another ".h" file), ADT / algorithm implementation (use a ".c" file), and driver/test code (use another ".c" code). In general, each module has to be written in **separate** c files.

4. All files related to a lab **must** be put inside a single directory by the name of the lab (lab1, lab2, etc.).

5. Valid makefile must be present in the directory.

6. Ensure that all the code written by you are compiling correctly. Preferably use gcc with the options **`-W -Wall -O2`**, while compiling your code.

Problem: Given a large text, create an index of words appearing in the text, along with the location(s) in which these words appear.

Data source for this problem can be downloaded from : http://www.gutenberg.org/ebooks/829 (you may use any other ebook as well.)

Algorithm Outline:

1. Sanitize the text to remove all characters that are not alphabets or white spaces.
2. Maintain one of the data structures mentioned below to handle index
3. Traverse the text word by word.
4. If it is a new word,
    a. add to index (along with location),
5. else
    a. search the word in index and add the extra location.
6. Do a Depth First Traversal of implemented data structures and print the Index to a file ***(index.txt)*** indicating word and the location(s) where the word is found.

Data Structure to be used for indexing:

a. Standard Trie

b. B-Tree of order 5