CIS 343 – Structure of Programming Languages
Winter 2016, 3/7/2016

Programming Assignment #3
Anagrams in C
Due Date: Tuesday, March 22, 2016

Objectives

- Command line arguments
- File I/O
- Pointers and linked lists
- Dynamic memory allocation
- Modularizing program with functions

Problem Specification

Write a program in C to read the words of a dictionary (a text file with one word per line consisting of only lower case letters), builds linked lists of anagrams, and prints the list of anagrams to the output file specified. The names of the input file (dictionary) and the output file are specified as command line arguments to the program.

A dynamically allocated array of structures is used to build and store lists of anagrams found in the input dictionary of words. Each element in the array is a structure with two fields – 1) a field that points to a linked list where each node stores a word that is an anagram to the words in the rest of the nodes in that linked list, and 2) a field that stores an integer that tells the number of nodes/anagrams in that linked lists.

Once the lists of anagrams is constructed, the program will print out the lists of anagrams to the output file specified on command line. You will print only the words in each linked list with two or more words (anagrams) in it.

Further details of this project will be discussed in class. It is important that you pay close attention to the in-class project discussion and ask questions to fully understand the data structure required to store the anagrams lists (array of structures with each structure pointing to a linked list).

Design Requirements

For this project, you are supplied with a C source file named `anagrams.c` that contains the functions listed below. The `main` and `freeAnagramArray` functions are already completed for you and should NOT be modified. Your task is to implement the remaining functions:

```
AryElement *buildAnagramArray(char *infile, int *aryLen);

void printAnagramArray(char *outfile, AryElement *ary, int aryLen);

void freeAnagramArray(AryElement *ary, int aryLen);
```

```
bool areAnagrams(char *word1, char *word2);

Node *createNode(char *word);
```

Program Compilation and Execution

```
$ gcc –Wall –std=c99 –o anagrams anagrams.c      (command to compile)
$ ./anagrams dictionary1.txt output1.txt        (command to run)
```

You will be provided with several test input files and the expected output files that you can use to test and verify your program.

Deliverables

1. Upload `anagrams.c` file on Blackboard by midnight on due date.
2. I will use the submission date/time on Blackboard as your official submission date/time.
3. It is your responsibility to make sure the submission on Blackboard went through successfully.
4. Because of possible portability issues, make sure your program compiles and runs on EOS machines before submitting any source file(s) on Blackboard. I will compile, run, and test your program on EOS when grading.
5. Late penalty (10% per day) applies after midnight on Tuesday, March 22nd.