//

// main.c

// Chpt11PP5

//

// Created by Randy McMillan on 11/28/13.

// Copyright (c) 2013 Randy McMillan. All rights reserved.

//

/\*

\*

\* 5. Write a program that takes words from a text file and prints each one on a separate line of an output file followed by the number of letters (alphabetic characters) in the word. Any leading or trailing punctuation marks should be removed from the word before it is printed. When all the text has been processed, display on the screen a count of the words in the file. Assume that words are groups of nonwhitespace characters separated by one or more whitespace characters.

\*

\*

\*/

// Met all the specs as opposed to the modified specs that were given during class

// Also included code to modify the .txt file name to create unique instances of the file

// Doing so made it easier to develop and debug

#include <stdio.h>

#include <string.h>

#include <time.h>

FILE \*text\_inp, \*text\_outp;

char ch;

int error = 0;

void myMain();

void openFiles();

void closeFiles();

void copyChars();

int main(int argc, const char \*argv[])

{

myMain();

return 0;

}

void myMain()

{

openFiles();

copyChars();

closeFiles();

// printf("\n");

}

void copyChars()

{

int c = 0;

int n = 1;

if (error != 1) {

for (ch = getc(text\_inp); ch != EOF; ch = getc(text\_inp)) {

if (strcmp(&ch, ".")) {

if (strcmp(&ch, " ")) {

putc(ch, text\_outp);

// printf("%c", ch);

} else {

char dig = '0' + c;

putc(' ', text\_outp);

putc(dig, text\_outp);

putc('\n', text\_outp);

// printf(" %i\n", c);

c = -1;

n++;

}

}

c++;

}

char dig = '0' + c;

putc(' ', text\_outp);

putc(dig, text\_outp);

printf("There are %i words in the file.\n", n);

}

}

void openFiles()

{

char buff[100];

time\_t now = time(0);

strftime(buff, 100, "%Y-%m-%d %H:%M:%S", localtime(&now));

text\_inp = fopen("abc.txt", "r");

text\_outp = fopen(\_\_DATE\_\_ "\_"\_\_TIME\_\_ ".txt", "w"); // unique file name

if (text\_inp == NULL) {

printf("Cannot open abc.txt for input\n");

error = 1;

}

if (text\_outp == NULL) {

printf("Cannot open results.txt for input\n");

error = 1;

}

}

void closeFiles()

{

fclose(text\_inp);

fclose(text\_outp);

}



