/\*

CS320

Arthur Flores masc0120

29 April 2015

Print the words on each line backwards

Dr. Carroll

Return Values:

0=program executed successfully

1=At least one file could not open

2=At least one file is empty

3=Max characters per line has been reached

4=No filename after -o for extra credit

I believe these return values are best because they are numerically ordered to the rational point

when the program would give out these error messages.

\*/

#include <stdio.h>

#include <stdlib.h>

#define BUF\_SIZE 100

void reverse(char \*start, char \*finish);

void reverseWords(char \*s) // some of this information was obtained by CbyDiscovery

{

char \*start = NULL;

char \*tmp = s; //tmp is being used for a word boundry

while( \*tmp ){

//This while is checking that the string starts with a nonwhitespace character

if (( start == NULL ) && ((\*tmp != ' ') )){

start=tmp;

}

if(start && ((\*(tmp+1) == ' ') || (\*(tmp+1) == '\0'))){

reverse(start, tmp);

start = NULL;

}

tmp++;

}

reverse(s, tmp-1);

}

void reverse(char \*start, char \*finish){ //reverses order of words in line

char tmp;

while (start < finish){

tmp = \*start;

\*start++ = \*finish;

\*finish-- = tmp;

}

}

int printWords(int iochar, int argcNext, FILE \*fileNext, int extraCredit, FILE \*fileOut){ //Does all the printing to screen

char array[BUF\_SIZE + 1], blank[BUF\_SIZE + 1]; //I chose this value for max charcters because

int c = 0, d = 0; //the standard number of charcters

int i=0, j=0; //per line is 80 so I set it to 100 for some wiggle

char \*tmp = array;

int count=0; //counts characters in line so array knows when to stop printing

int haschar=0;

int prevchar;

int checkedReturn=0;

int maxlineReached=0; //indicates whether a huge line was entered

int n; // to check for a no new line at end of file

//Check if file is null

if(iochar == EOF){

fprintf(stderr,"File is Empty\n");

return 2;

}

do{

if(iochar == '\n'){

if(haschar==1){

while (array[c] != '\0') { //while loop to remove spaces and tabs between words

char \*ptr = NULL;

if(c==0){

for(ptr = array; \*ptr == ' '; ptr++) c++; // removes leading spaces and tabs

}

if (array[c] == ' ' ) { // if statements removes all extra spaces between words

int tmp2 = c + 1;

if (array[tmp2] != '\0') {

while ((array[tmp2] == ' ') && array[tmp2] != '\0') {

if (array[tmp2] == ' ') {

c++;

}

tmp2++;

}

}

}

blank[d] = array[c]; //replacing old array with extra spaces with compacted array

c++;

d++;

prevchar=array[c]; //stores current character for next loop

}

if(blank[d-1] ==' ')

d--;

blank[d] = '\0';

c=0;

d=0;

blank[BUF\_SIZE+1]; //resets blank array to null

reverseWords(blank);

if(extraCredit==1){

fprintf(fileOut,"%s\n",blank);

}

else{

printf("%s\n",blank);

}

}

for(i=count+1; i>=0; i--){ //resets array to null

array[i]=0;

}

count=0;

haschar=0;

j=0;

prevchar=0;

}

else{

if(iochar == '\t') //changes all tabs to a space

iochar=' ';

if(iochar != ' ') //checks if there is a non whitespace character in line.

haschar=1;

if(j==98){

fprintf(stderr,"Max number of charcters per line has been reached\n");

checkedReturn=3;

for(i=count+1; i>=0; i--){ //resets array to null

array[i]=0;

}

maxlineReached=1;

count=0;

haschar=0;

j=0;

prevchar=0;

}

else{

array[j++]= iochar;

count++;

}

}

if(argcNext >= 2){

if(maxlineReached==1){

do{

iochar=getc(fileNext);

}

while(iochar != '\n');

maxlineReached=0;

}

iochar=getc(fileNext);

}

else{

if(maxlineReached == 1){

do{

iochar=getc(stdin);

}

while(iochar != '\n');

maxlineReached=0;

}

iochar=getc(stdin);

}

if(iochar==EOF && n==0 && prevchar != '\n'){

iochar= '\n';

n++;

}

}

while(iochar != EOF);

return checkedReturn;

}

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

//Initialized Values

FILE \*file, \*fileOut;

int c = 0, d = 0, n=0;

int iocharStart;

int wordsValue=0; //This is the return value which checks for errors

int checkReturn=0; //checks if any past return value has occured from other files

char \*extraCredit = "-o";

int exCredFound;

int exCredCheck=0;

if(argc >= 2){

for(n = 1; n <= argc-1; n++){

if(\*argv[n] == \*extraCredit){

if(n==argc-1) return 4;

fileOut=fopen(argv[n+1], "w");

exCredFound=n+1;

exCredCheck=1;

}

}

for(n = 1; n <= argc-1; n++){

if(n==exCredFound-1 || n == exCredFound){

}

else{

file=fopen(argv[n], "r");

if(file == 0){

fprintf(stderr,"File %s could not be opened\n",argv[n]);

checkReturn=1;

}

else{

iocharStart=getc(file);

wordsValue= printWords(iocharStart, argc, file,exCredCheck,fileOut);

if(wordsValue != 0) checkReturn=wordsValue;

fclose(file);

}

}

}

}

else{

iocharStart=getc(stdin);

wordsValue= printWords(iocharStart, argc, 0,0,0);

}

if(checkReturn != 0) wordsValue=checkReturn;

return wordsValue;

}