IO Processing Assignment

Juan Velasquez

Code:

/\*

Student Name: Juan Velasquez

Date Written: 9/30/2021

Purpose of program: This program allows the user to read and write words

to a text and bin file. The words have a length limit of 100 letters per

word and the total words allowed are 1000.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <string.h>

//Function Declarations

void addWord(char word[][100], int \*length);

void displayMenu();

void showWords(char word[][100], int length);

void getUserChoice(char \*choice);

void flush();

void loadWords(char word[][100], int \*length);

void dumpWords(char word[][100], int length);

void saveBin(char word[][100], int length);

void textFile(char word[][100], int length);

/\*

This function adds a word that the user inputs into the 2d character array.

It requires a 2d character array and a length to see how many words are in

the array currently.

\*/

void addWord(char word[][100], int \*length) {

//Checking if the array already has 1000 words

if (\*length == 999) {

/\*

If so, no more words can be added so alert the user and exit the

function.

\*/

printf\_s("Word array is full...no more words to be added...\n");

system("pause");

return;

}

/\*

If not, then ask the user for a word and add it to the 2d character

array.

\*/

printf\_s("\tEnter word %i of %i: ", (\*length + 1), 1000);

scanf("%s", word[\*length]); flush();

(\*length)++;

}

//This function displays the menu for the user

void displayMenu() {

//system("cls");

printf\_s("\n\n\n");

printf\_s("\t[A]dd a word\n");

printf\_s("\t[S]how all words\n");

printf\_s("\t[C]reate text file of words\n");

printf\_s("\t[D]elete all words\n");

printf\_s("\t[Q]uit\n\n");

printf\_s("\tEnter selection: ");

}

/\*

This function shows all of the words in the 2d array. Needs the 2d character

array and the total number of words in the array.

\*/

void showWords(char word[][100], int length) {

//Loop through the array and print each word.

for (int i = 0; i < length; i++) {

printf\_s("%i. %s\n", i + 1, word[i]);

}

system("pause");

}

//This function gets the user's choice for the menu option and returns it.

void getUserChoice(char \*choice) {

displayMenu();

scanf("%c", choice); flush();

\*choice = toupper(\*choice);

}

//

void flush() {

while (getchar() != '\n');

}

/\*

This function loads the words stored from a previously made .bin file.

It first opens the file and then proceeds to set the length and 2d word array

equal to the contents of the .bin file.

\*/

void loadWords(char word[][100], int \*length) {

FILE \*file;

char reload;

//Ask the user if they want to load from a .bin file.

printf\_s("Do you wish to reload words if found (Y or N)? ");

scanf("%c", &reload); flush();

//If not, exit the function.

if (toupper(reload) != 'Y') {

printf\_s("Reload skipped...\n");

system("pause");

return;

}

//If so, open the file under the name bin.bin, using the read bin option of fopen.

file = fopen("bin.bin", "rb");

//If the file does not exist, then print that out to the user.

if (file == NULL) {

printf\_s("I could not OPEN the bin file for reading...error or not found.\n");

system("pause");

}

/\*

If the file does exist, set the length and word variables equal to the

corresponding data from the file.

\*/

else {

fread(length, sizeof(int), 1, file);

fread(word, sizeof(char)\*100, (int)length, file);

}

//Close the file

fclose(file);

}

//This function empties the words in the word array and empties the .bin file.

void dumpWords(char word[][100], int length){

FILE \*file;

//Open the file by using the write to bin option.

file = fopen("bin.bin", "wb");

//If the file does not exist, then print that out to the user.

if (file == NULL) {

printf\_s("I could not create the bin file to save the data.\n");

system("pause");

}

/\*

Write nothing to the file making it empty, as we overwrote the data in it.

Then close it.

\*/

fclose(file);

//Loop through the 2d array and set every character equal to space.

for(int i = 0; i < length; i++){

for(int j = 0; j < 100; j++){

word[i][j] = ' ';

}

}

system("pause");

}

//This function saves the current 2d word array to a bin file named bin.bin

void saveBin(char word[][100], int length) {

FILE \*file;

//Open the file by using the write to bin option.

file = fopen("bin.bin", "wb");

//If the file does not exist, then print that out to the user.

if (file == NULL) {

printf\_s("I could not create the bin file to save the data.\n");

system("pause");

exit(-1);

}

//Write the length and each word to the .bin file

fwrite(&length, sizeof(int), 1, file);

fwrite(word, sizeof(char)\*100, length, file);

//Then close the file

fclose(file);

}

//This function saves the current 2d word array to a text file named text.txt

void textFile(char word[][100], int length) {

FILE \*file;

//Open the file by using the standard write option.

file = fopen("text.txt", "w");

//If the file does not exist, then print that out to the user.

if (file == NULL) {

printf\_s("I could not create the text file to save the data.\n");

system("pause");

exit(-1);

}

/\*

Loop through the first column of the word array and write each word to the

text file

\*/

for (int i = 0; i < length; i++){

fprintf(file, "%i. %s\n", i + 1, word[i]);

}

//Close the file

fclose(file);

}

int main(void) {

//2d char array to store the words

char words[1000][100] = { ' ' };

//Tracks how many words are in the 2d array

int length = 0;

//This is used to track the user input through the menu.

char option = 'Q';

//Ask the user if they want to load a previously made .bin file, if so, then do it.

loadWords(words, &length);

//Menu Design according to menu sample code.

do {

getUserChoice(&option);

switch (option) {

//Add a word option

case 'A':

addWord(words, &length);

break;

//Show all words option

case 'S':

showWords(words, length);

break;

//Create a text file with all the words

case 'C':

textFile(words, length);

printf\_s("\nCreated text file.\n");

break;

//Dump all words from the array and .bin file

case 'D':

dumpWords(words, length);

//Set the length to 0 as there are no more words in the 2d array

length = 0;

printf\_s("\nDeleted all words\n");

break;

//Quit option that saves the current words to a bin file.

case 'Q':

saveBin(words, length);

printf\_s("\nGoodbye!\n");

system("pause");

break;

//Incase the user's input is not valid.

default:

printf\_s("Invalid selection...\n");

system("pause");

break;

}

}

while (option != 'Q');

//Exit program

return 0;

}

Screenshots:

Text

Description automatically generatedText

Description automatically generatedGraphical user interface, text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated