Text Files

Workshop 9 (out of 10 marks - 3% of your final grade)

In this workshop, you will code a C-language program that uses secondary storage in the form of a text file.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to open and close a text file
- to read records sequentially from a text file
- to append records to a text file
- to describe to your instructor what you have learned in completing this workshop

SUBMISSION POLICY

The "in-lab" section is to be completed during your assigned lab section. It is to be completed and submitted by the end of the workshop period. If you attend the lab period and cannot complete the in-lab portion of the workshop during that period, ask your instructor for permission to complete the in-lab portion after the period. If you do not attend the workshop, you can submit the “in-lab” section along with your “at-home” section (with a penalty; see below). The “at-home” portion of the lab is due on the day that is two days before your next scheduled workshop (23:59).

All your work (all the files you create or modify) must contain your name, Seneca email and student number.

You are responsible to back up your work regularly.

Late submission penalties:

- In-lab portion submitted late, with at-home portion: 0 for in-lab. Maximum of 70/70 for at-home and reflection
- If any of in-lab, at-home or reflection portions is missing the mark will be zero.

IN-LAB: (30%)

Download or clone workshop 9 (**WS09**) from https://github.com/Seneca-144100/IPC-Workshops

Write your code for this segment in the bookstore_v2_in_lab.c file provided with the Visual Studio template project inside the in_lab folder.

1. (Copy from workshop 8)

In this workshop, you are going to use the same C structure type as in workshop 8 to represent a **Book**. Then, you will create an object of struct Book to represent a single book in a bookstore. Here is the C structure type that should be used in this workshop:

```
struct Book {
   int _isbn;
   float _price;
   int _year;
   int _qty;
   char _title[MAX_TITLE_SIZE];
};
```

`_isbn`: International Standard Book Number.
`_price`: Book price.
`_year`: Publication year.
`_qty`: Book Quantity.
`_title`: Book title.

Also, use **`const`** or **`#define directives`** to define the following number:
`MAX_TITLE_SIZE`: the maximum number of characters in a book title. Assume **`MAX_TITLE_SIZE`** is 20, excluding the null terminator**.**

Look at the file **144_w9_inventory.txt**, this file is in workshop 9 directory. This file contains three records. Here is the first record:
`234562;23.99;2010;3;Harry Potter`

Here is what each field represents:
`_isbn;_price;_year;_qty;_title`.

In your main function, implement the following steps:

2.
Define the following variables in your **main** program:

```
struct Book mybook; //An object of struct Book
char filename[] = "144_w9_inventory.txt"; //Name of the file
```

[Similar to Workshop 8] Display a welcome message:

```
Welcome to the Book Store
=========================  //25 assignment operators
```

3. [Note: Only option 4 is added]
Write the following function to display a menu. This function takes no arguments and displays the following menu to the user.

```
void menu();
/*
Please select from the following options:
 1) Display the inventory.
 2) Add a book to the inventory.
 3) Check price.
 4) Calculate total capital of the store.
 0) Exit.
*/
```

4.
Write the following function to read a single record and save it to **`*b2read`**.
This function takes two arguments: the address of a FILE object, and the address of a struct Book object. The function returns the return value of scanf.
(Hint: Define a variable **`int rv = 0`**; and **`rv = scanf(…..)`**, then **`return rv`**)

```
int readRecord(FILE *fp, struct Book *b2read)
```

5.
Write the following function to display an inventory. This function takes the filename as an argument. The function displays the inventory in an informative output (The format of the output is similar to the output of workshop 8, see sample output below).

Note: In this function, you must use the function **`readRecord`**, written in step 4. Iterate a loop as long as the return value of **`readRecord`** is 5. Why 5?

```
void displayInventory(const char filename[])

==================================================
//51 "="
```

```
ISBN      Title                Year Price  Quantity
---------+--------------------+----+-------+--------
//9 dashes + 19 dashes + 4 dashes + 7 dashes+ 8 dashes
```

Use the format specifier: `%-10.0d%-20s%-5d$%-8.2f%-8d`

6.

Write the following function that takes a filename as an argument, calculates the total capital of the inventory, and returns the total capital. The total capital is computed by accumulating the multiplication of the price of each item to its quantity.
(Hint: total_capital = total_capital + book_price*book_quantity)

Note: Similar to step 5, in this function, you must use the function **readRecord**. Iterate a loop as long as the return value of **readRecord** is 5.

```
float calculateCapital(const char filename[]);
```

7.

Write the following three empty functions. These functions display "Not implemented". These functions will be completed in the at home part of the lab.

```
int searchInventory(FILE *fp,const int isbn2find)
void addBook(const char filename[], struct Book *b2Add)
void checkPrice(const char filename[],const int isbn2find)
```

8.

Write the main function that

- Defines the variables, and displays a welcome message (Step 2),
- Calls the function **menu** to display a menu to the user,
- Depending on the user's input, calls one of the following functions, and passes proper arguments:
    - If user's input is 1: Call **displayInventory.**
    - If user's input is 2: Call **addBook.**
    - If user's input is 3: Call **checkPrice.**
    - If user's input is 4: Call **calculateCapital,** and display the total capital
    - If user's input is 0: The program exits and displays a goodbye message.
      (Note: The program only exits when the user selects Exit on the menu screen (looping required).

Program completion

Your program is complete if your output matches the following output. Red numbers show the user's input.

```
Welcome to the Book Store
=========================
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title                Year Price  Quantity
---------+--------------------+----+-------+--------
234562    Harry Potter         2010 $23.99   3
567890    The Hunger Games     2015 $12.67   4
109821    Stranger Things      2017 $53.20   2
=================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 4
The total capital is: $229.05.
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 0
Goodbye!
```

IN_LAB SUBMISSION:

To test and demonstrate execution of your program use the same data as the output example above or any information needed.

If not on matrix already, upload your bookstore_v2_in_lab.c to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit 144_w9_lab <ENTER>
```

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.

AT_HOME: (30%)

In this segment, you upgrade your app to include add book, search inventory and check price functions.
Copy and paste your code from the in-lab segment to the file bookstore_v2_at_home.c
1.

Update the following function to perform a linear (naïve) search over the inventory array. This function receives the address of a file object, and an integer for the **ISBN** of the desired book. This function uses the function **readRecord** iteratively to read the records. The function returns a non-negative number if the matching book is found, -1 if it is not found.

```
int searchInventory(FILE *fp,const int isbn2find)
```

2.
Update the function **addBook** to use the function **searchInventory**. This function takes two arguments: the file name, and the address of an object book to be added to the inventory.

In this function:

- Open a file to append.
- Use **searchInventory** function to search through the inventory to find if the book exists in the inventory.
- If the book is found in the inventory, do the following:
    - Display a message that “The book exists in the repository!”.
- If the book is not found in the inventory, do the following:
    - Use the file pointer to append the book to the file.
    - Display a message “The book is successfully added to the inventory”.

```
void addBook(const char filename[], struct Book *b2Add)
```

3.
Update the following function to check the price for a book. This function receives a file name and, an integer representing the **isbn** of a book. Then, this function searches through the inventory (use **searchInventory**), and displays the cost of the book. Note that if the book does not exist in the inventory, the program displays an informative message.

```
void checkPrice(const char filename[],const int isbn2find)
```

4. Update the main function:

In case 2: prompt the user to input information for the object **mybook**.
Input isbn, price, year, quantity, and the title of the book.
Then, call **addBook.**

In case 3: prompt the user to input the ISBN of the book they are looking for. Then, call **checkPrice.**

PROGRAM COMPLETION

Your program is complete if your output matches the following output. Numbers in red color shows the user's input.

```
Welcome to the Book Store
=========================
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title                Year Price  Quantity
---------+--------------------+----+-------+--------
234562    Harry Potter         2010 $23.99   3
567890    The Hunger Games     2015 $12.67   4
109821    Stranger Things      2017 $53.20   2
=================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 3
Please input the ISBN number of the book:

234562
Book 234562 costs $23.99

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 2
ISBN:987654
Title:Tintin in Tibet
Year:1960
Price:25.60
```

```
Quantity:2
The book is successfully added to the inventory.

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title                Year Price  Quantity
---------+--------------------+----+-------+--------
234562    Harry Potter         2010 $23.99   3
567890    The Hunger Games     2015 $12.67   4
109821    Stranger Things      2017 $53.20   2
987654    Tintin in Tibet      1960 $25.60   2
=================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 2
ISBN:378021
Title:Born a crime
Year:2016
Price:40.99
Quantity:3
The book is successfully added to the inventory.

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title                Year Price  Quantity
---------+--------------------+----+-------+--------
```

```
234562    Harry Potter         2010 $23.99   3
567890    The Hunger Games     2015 $12.67   4
109821    Stranger Things      2017 $53.20   2
987654    Tintin in Tibet      1960 $25.60   2
378021    Born a crime         2016 $40.99   3
==================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 2
ISBN:234562
Title:Harry Potter
Year:2010
Price:23.99
Quantity:3
The book exists in the repository!

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 3
Please input the ISBN number of the book:

987654
Book 987654 costs $25.60

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 4
The total capital is: $403.22.
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
4) Calculate total capital of the store.
0) Exit.

Select: 0
```

```
Goodbye!
```

AT-HOME REFLECTION (40%)

Please provide brief answers to the following questions in a text file named reflect.txt.

1) What are the possible return values of the function **readRecord**? What does this return value represent?
2) What is the difference between opening a file in r, w, and a+ mode?
3) Briefly explain the value of reading data from a file as opposed to reading all data from user input.

Note: when completing the workshop reflections it is a violation of academic policy to cut and paste content from the course notes or any other published source, or to copy the work of another student.

AT_HOME SUBMISSION:

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload your bookstore_v2_at_home.c and reflect.txt to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account (replace profname.proflastname with your professors Seneca userid):

```
~profname.proflastname/submit 144_w9_home <ENTER>
```

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.