Functions, Arrays and Structures

Workshop 8 (out of 10 marks - 5% of your final grade)

In this workshop, you will code a C-language program with functions that change the values of array elements and objects of structure type declared outside the functions.

LEARNING OUTCOMES

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

- to access elements of an externally declared array from within a function
- to assign values to an externally declared struct-type variables from within a function using a pointer
- to store related data using an array of structures
- to implement structured programming principles, including single-entry/single-exit logic
- 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 8 (**WS08**) from https://github.com/Seneca-144100/IPC-Workshops

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

In this segment, you will implement Add and Display functionality using an array of structs and pointer syntax.

1.

In this workshop, you are going to use a C structure type to represent a **Book**. Then, you will create an array of struct Book to represent an inventory of books in a bookstore. A person is able to view the inventory, add a book to the inventory, and check prices for books in the inventory. Here is the C structure type that should be used in this workshop:

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

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

Also, use **const** or **#define directives** to define the following number:
MAX_BOOKS: the maximum number of books that exists in an inventory. Assume **MAX_BOOKS** is 10**.**
MAX_TITLE_SIZE: the maximum number of characters in a book title. Assume **MAX_TITLE_SIZE** is 20, excluding the null terminator**.**

In your main function, implement the following steps:

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

```
struct Book book[MAX_BOOKS]; //An array of Book representing the inventory
int size=0; //Number of books in the inventory. The inventory is initially empty.
```

Display a welcome massage:

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

3.

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.
 0) Exit.
*/
```

4.

Write the following function to display an inventory. This function takes the address of an array of objects of type Book (**`book[]`**), and an integer **`size`** representing the size of the array. The function displays the inventory in an informative output (See sample output below).

```
void displayInventory(const struct Book book[],const int size);
```

```
==================================================
//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`

5.

Write the following empty function. This function displays "Not implemented". This function will be completed in the at home part of the lab.

```
void searchInventory(const struct Book book[],const int isbn,const int size);
```

6.

Write the following function to add a book to the inventory. This function takes the address of an array of objects of type Book (**`book[]`**), and the address of an integer `size` representing the size of the array.

```
void addBook(struct Book book[], int *size);
```

This function:

- Checks if the inventory is full (`*size== MAX_BOOKS`), and displays the message "the inventory is full".
- If the inventory is not full, do the following actions:
    - Prompts the user to input the ISBN, book title, publication year, price, and the quantity of the book that one wants to add to the inventory.
    - Then, updates the inventory. Hint: Use the **book** array, and `*size` as its index. Increment size. Once the book is added to the inventory, display a message that the book is successfully added to the inventory.

7.

Write the following empty function. This function displays "Not implemented". This function will be completed in the at home part of the lab.

```
void checkPrice(const struct Book book[],const int size);
```

10.

Write the main function that

- Defines the variables, and displays a welcome massage (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 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.
0) Exit.
```

```
Select: 4
Invalid input, try again:
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 1
The inventory is empty!
==================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 2
ISBN:234567
Title:Harry Potter
Year:2010
Price:23.99
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.
0) Exit.

Select: 2
ISBN:567890
Title:The Hunger Games
Year:2015
Price:12.67
Quantity:4
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.
0) Exit.

Select: 1


Inventory
```

```
=================================================
ISBN      Title               Year Price  Quantity
---------+-------------------+----+-------+--------
234567    Harry Potter        2010 $23.99   3
567890    The Hunger Games    2015 $12.67   4
=================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
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_app_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_w8_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 search inventory and check price function. Copy and paste your code from the in-lab segment to file bookstore_app_home.c

1.

Update the following function to perform a linear (naïve) search over the inventory array. This function receives the address of an array of type **`Book (book[])`**, an integer for the **`ISBN`** of the desired book, and an integer **`size`** representing the size of the array. This function searches through the array for a book with the desired **`ISBN`** and returns the index of the matching book if it is found, -1 if it is not found.

```
int searchInventory(const struct Book book[],const int isbn,const int size);
```

2.

Update **`addBook`** function to use **`searchInventory`** function.

This function:

- Prompts the user to input the ISBN and the quantity of the book.
- Use **searchInventory** function to search through the inventory (the **book** array) to find if the book exists in the array.
- If the book is found in the inventory, do the following:
    - Update the quantity of the book by adding the quantity read to the quantity of the book in array.
    - Display a message that the book quantity is successfully updated.
- If the book is not found in the inventory, do the following:
    - If the inventory is full (**`*size== MAX_BOOKS`**), display the message “the inventory is full”.
    - If the inventory is not full, similar to in_lab version, do the following actions:
        - Prompts the user to input the ISBN, book title, publication year, price, and the quantity of a book that one wants to add to the inventory.
        - Then, update the inventory. Hint: Use the **`book`** array, and **`*size`** as its index. Increment size. Once the book is added to the inventory, display a message that the book is successfully added to the inventory.

3.

Update the following function to check the price for a book. This function receives the address of an array of **`Book (book[])`**, and an integer representing the **`size`** of the array. The function prompts the user to input the ISBN of the book they are looking for. 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 struct Book book[],const int size);
```

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.
0) Exit.

Select: 4
Invalid input, try again:
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 1
The inventory is empty!
==================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 2
ISBN:234567
Quantity:3
Title:Harry Potter
Year:2010
Price:23.99
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.
0) Exit.

Select: 2
ISBN:567890
Quantity:2
```

```
Title:The Hunger Games
Year:2015
Price:12.67
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.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title               Year Price  Quantity
---------+-------------------+----+-------+--------
234567    Harry Potter        2010 $23.99   3
567890    The Hunger Games    2015 $12.67   2
=================================================

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 2
ISBN:234567
Quantity:2
The book exists in the repository, quantity is updated.

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

Select: 1


Inventory
=================================================
ISBN      Title               Year Price  Quantity
---------+-------------------+----+-------+--------
234567    Harry Potter        2010 $23.99   5
567890    The Hunger Games    2015 $12.67   2
=================================================
```

```
Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

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

56789
Book does not exist in the bookstore! Please try again.

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
0) Exit.

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

567890
Book 567890 costs $12.67

Please select from the following options:
1) Display the inventory.
2) Add a book to the inventory.
3) Check price.
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) When passing an array to a function is it "by address" or "by value" – explain what happens and what is meant by the two terms.

2) Explain why functions with an array parameter are usually complemented with an additional integer parameter (reference specifics in this workshop to explain your answer).

3) Why do some of the functions in this workshop specify "const" for the array parameters and not in others?

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_app_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_w8_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.