**AHMED SAMIR ABDELFATTAH 8120**

**YOUSSEF AHMED AWAD 8179**

**MOHAMED AHMED MORSY 8199**

PROJECT REPORT

Structs:

* The first struct is the date struct which consists of two fields
  + First field is **month** which is of type integer
  + Second field is **year** which is of type integer
* The second struct is the book
* First field is the **ISBN** which is an array of character
  + - * Second field is **Title** which is an array of characters
      * Third field is the **author** which is an array of characters
      * Fourth field is the **quantity** of the books which is an integer
      * Fifth field is the **price** which is a float
      * Sixth field is the **date published** which is of type date(the first struct)

Global Variables:

* N is the **number of books** which is of type integer so it can be used in all functions when the load function calculates how many books are there in the file so there is no need to keep passing the N to all functions
* x[100] is the **global array of structs** which contain all of the books as it will be used in every function without the need of passing it to all function after the load function stores all the data into the array of structs.
* const char \*m[12]= {"January","February","March","April","May","June","July","August","September","October","November","December”}; the array is a pointer to a constant character. This means that the array is a variable that stores the address of the first element of an array of characters that cannot be modified. The type of the array is "const char \*", which can be read as "pointer to constant char" .Use of pointers allow the array to be passed to the functions.

Login Function:

* In this function, the user is required to enter his username and password. This function contains 3 different arrays.
* username [ ], to store the username from the user.
* password [ ], to store password from the user.
* lines [ ], to store the data from the text file.
* After the user inputs his username and password and stored successfully, the function will try accessing the file called “credentials.txt” to compare username and password entered with the ones stored in the file.
* A Validation that the file opened and accessed correctly is done. Then, using the fgets() function , a while loop is made to copy each line in the text file to the array lines[ ] ; using the variable “j” as a counter.
* A lot of other variables was declared at the begging such as:
* i, counter for the for loop.
* j, counter for the while loop.
* flag, to check if login was successful or not.
* The while loop will keep running until it is end of file, taking each line in the file and inserting it in the array lines[ ]. “j” increments by 1 each time in the while loop, indicating a new line. This series of iterations will end up with the “j” = number of lines.
* After the lines are all stored in the array lines[ ], It is time for the for loop. Starting from “i” = 0 to i<j. “i” will increment by 2 in each loop
* An If statement will be inserted in the for loop to print “Login Successful” if each character of lines[i] is equal to each character of username[] **AND** each character of lines[i+1] is equal to each character of password[].
* We faced a problem while debugging this function. The problem was that our file added an extra character to the string in it without knowing what it is! We checked for a spacebar or an enter button, but it didn’t seem to be that. It wasn’t the “/0” either.
* To solve this problem, we used the function strncmp(), which take the 2 strings you want to compare + till which character do you want to compare. So, we set it up like that: strncmp(username,lines[i],strlen(lines[i])-1). This will return 0 if both strings are equal.

Graphical user interface, text, application

Description automatically generatedKeep in mind:

this is how the file “credentials.txt” look like.

* The if statement will set the flag variable to 0 if no match was found, then using another if statement, we will check that if flag is equal 0 then print “Login Failed”.
* This Function will return 0 if login failed , and return 1 if login successful.

LOAD function:

This function is used to scan the contents of the file (books.txt) and store it inside our structure.

* First open the file on read only mode and assign it to f1
* Check that the file is not empty
* Using an array of strings "lines[200] [200]”, we will scan each line in the file until we reach the end of the file while incrementing to get the number of books in the file
* We need to subtract 1 from the number of books variable because it increments one extra time before the loop terminates when it is end of file
* It is important to return the pointer “f1” to the beginning of the file using “rewind “function
* Now we will scan the file again to save each data type in its structure
* We face a problem that all different data of a same book are on the same line separated with commas only
* Through a loop that runs n “number of books” times, using regular expressions or regex we will force the compiler to scan each string till it sees a comma with this format %[^,], this is only needed when we scan a string, an integer will be scanned normally with %d or %f.
* Now every data type is stored in its array of structure.

Search:

* The search function works by the main algorithm of the linear search .First of all the user press the option of searching for a certain Book he is asked to type the ISBN of the book he is searching for then it is checked that the ISBN entered is only 13 digits without any numbers if the user didn’t enter the ISBN with the exact specifications he will be asked to enter the ISBN again.

Text

Description automatically generated

* Here is the output when the user enters 14 digits.

Text

Description automatically generated

* Here is the output when the user enters alphabets even when its 13 characters
* So when the user enter the ISBN with the correct specifications the ISBN is searched for by linear search so the compiler loops through the ISBN stored in the structs and compare each ISBN with the one the user entered if found then Book Found is printed and the loop stop by a break and the flag is then equal 1 and the data of the corresponding book is displayed but if the loop end by reaching the last ISBN without finding the matching ISBN the the flag remain equal 0 and Book not found is printed.

Text

Description automatically generated

Here is the output when the book is found

A picture containing text, orange

Description automatically generated

Here is the output when the Book is not found

Advanced Search:

* The advanced search works mainly by strstr function so first of all when the user chooses the advanced search option he is asked to enter a keyword which is included in the title he is searching for when the user enter the word it is stored in an array of characters the by linear search the loop passes through all titles and strstr(x,y) takes 2 parameters the first parameter x is the title of the book which is the longer string and the second parameter y is the word the user entered and if y is a substring in x the function return 1 and if its not the function return 0 so the loop passes through all the titles and by using strstr if it returns 1 then this mean that the word the user entered is a substring in the title so then the data of the book is printed and the next book title is compared but if the strstr return 0 then no data is printed and the next title is compared until the loop ends when all titles are compared.

Text

Description automatically generated

* Here is the output when the keyword is found in the title.



* Here is the output when the keyword is not found in all titles

Add Function:

* This function used to add a new book to a library system. It prompts the user to enter the book's ISBN, title, author, quantity, price, and month and year of publication.
* The function checks the validity of the ISBN by making sure that it is a string of length 13 and that all of its characters are digits.
* The function also checks the validity of the month and year of publication by calling the isvalid() function.
* If the month and year are found to be valid, they are converted from strings to integers using the atoi() function and stored in the book's published date.
* The function then increments a global counter variable "n" and stores the book's information in the array of book structures called "x".

For example:

A screenshot of a computer

Description automatically generated with medium confidence File before sample:

Text

Description automatically generatedSample Run:

Text

Description automatically generatedFile after sample:

DELETE function:

This function is used to delete the contents about a book saved in the file.

* The user is asked to enter the isbn of the book, the we validate that the isbn is exactly 13 characters and only numbers entered
* Using strlen function we get how many numbers are there in the isbn and then check if it is equal to 13 else the user is asked to enter the isbn again.
* Then check if the isbn entered is all numbers, isdigit function is used to know if this character is a digit or not, and it is looped 13 times for the 13 integers entered. If any non digit detected the user is asked to enter the isbn again.
* Once the isbn is validated, we use the strcmp function to compare the entered isbn to the saved isbn’s in the file.
* If isbn of the book is found then we replace the space taken by this book in every data type with the contents of book after it, now the place of the other book is empty so we repeat the process so that every book moves one place behind.
* Now the last place in array of structure is empty so we set a flag that if an item is indeed deleted, then we decrement the number of books available.
* If isbn is not found then we print book not found to the user.
* Sample run:
* before
* Text

  Description automatically generated
* Run
* A screenshot of a computer

  Description automatically generated with medium confidence
* After
* Text

  Description automatically generated

MODIFY function:

This function is used to change the data of any book available

* The user enters the isbn of the book that he wants to change its contents and the isbn is validated like in previous functions
* The is asked what he wants to change ex:title or name or quantity....etc, and he enters a corresponding number to the data he wants to change
* If he wants to change the month published for example, user enters the new value and it is stored in variable string and then validate it to check that only numbers are entered and that it is between 1 and 12,
* Then the new value is assigned to the old one in the struct to replace it
* The user is asked if he wants to modify again or exit

Before

Text

Description automatically generated

run

Text

Description automatically generated

After

Text

Description automatically generated

SAVE function:

This function is used to return all the changed values back to the file and save it

* First the file is opened on writing mode.
* Then looped through the number of books that we have,the contents inside the struct is printed in the same format as it should be.

Isvalid function:  
this function is used to validate month and year to only enter numbers

* We use strlen to get how many characters are there in the string
* Then use isdigit function to check every character entered if it is a digit or a letter

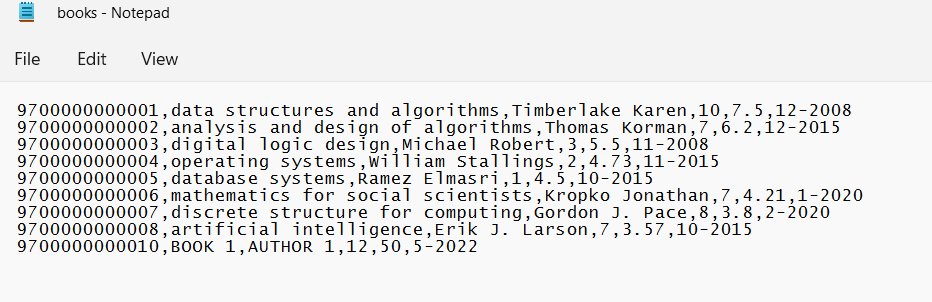
If it is a char then it return 1 to the function and the user is asked to input again

Print (Sort) Function:

* The purpose of the print function is to allow the user to sort and display a list of books based on different criteria.
* The function first asks the user to choose how they would like to sort the books by displaying a menu of options: Sort by title, Sort by price, and Sort by date.
* The user's choice is stored in the variable "choice".
* The function then checks the value of "choice" and calls one of three sorting functions (sortByTitle, sortByPrice, or sortByDate) depending on the user's selection. These sorting functions use bubble sort to arrange the books in ascending order based on the chosen criteria.
* After the books have been sorted, the function checks the value of "choice" again. If "choice" is 1, 2, or 3 (indicating that the user chose to sort the books by title, price, or date of publication, respectively) the function loops through the array of book structures and prints out the ISBN, title, author, quantity, price, and publication date of each book.
* If "choice" is any other value, the function prints an error message indicating that the user's choice was invalid.

For example:

File before Sorting:

 Sample Run:

Text

Description automatically generatedText

Description automatically generated Sort by Title:

Sort by Price:

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generatedText

Description automatically generated Sort by Date of Publication :

Menu Function:

* This function is the mother function of this project. It is the only function that runs in the Main().
* When this function is called, it sends greeting message to the user and ask him to choose between 2 options and store the option in a variable called “choice 1”.
* The 2 options are “Login” and “Quit”. If the user chose login( pressed 1), the Login function will be called.
* It is called inside an if statement. If login is successful (returned 1) , then the if statement is true. Which means that the data in file will be loaded successfully, and the menu will appear to the user.

*Text

Description automatically generatedThe menu will look like this*:

* Then the user can input any number and the function corresponding will be called.
* After each function , the goto start; statement will return the code back to the

“please choose an option ….”

* When the Save function is called (pressed 7) , the user will be asked to either save and quit or save and return back to menu.

Quit:

* When the user chooses the quit option a warning message is printed for him stating that the changes the user has done will be discarded without saving.

Text

Description automatically generated

* Here is the output when the user choose to quit then the warning is displayed then the user is allowed to choose between 2 options whether to quit the program or to return to the menu.

Text

Description automatically generated

* Here is the output when the user chooses to quit without saving .

Text

Description automatically generated

* Here is the output when the user chooses to return to the menu.

User manual:

When the user access the system he has to options whether to login or quit if he chose login then he is asked to enter his username and password if it is correct then the user has login successfully and the menu is displayed on the screen with all the options available to use after you choose what you want to do and you are done the the menu is displayed again if you want to do another change when you finish all your changes then choose save then save and quit for your changes to be save and exit the program or save and return to the menu if you need to do another change if you dont want to save and exit then choose Quit which discard all your changes so be careful so when you choose quit you have two options either to quit without saving or return to the menu again.