Define constants and macros: MAX\_BOOKS for 10 and FILENAME for the binary file name.

no

no

yes

yes

Input choice

Display maximum limit reached

If i<count

Declare and assign i =0

Validate input for book details

Input book number

Is count<MAX\_BOOKS

Call the function saveBookDetails(struct Book \*books, int count)

and pass the parameters books, bookCount.

Is choice =1?

Display the main menu

Dynamically allocate memory for books and Declare bookCount,choice and initialize to bookCount=0.

Assign bookCount = loadBookDetailsFromFile(books), call the function and pass parameters, to check number of books

Declare function prototypes: void saveBookDetails(struct, int), void displayCategoryWiseList(struct , int),

void displayCategoryWiseTotalAndAverage(struct, int),void displayBookDetails(struct, int, int,); void clearAllBooks(struct, int);

void saveBookDetailsToFile(struct, int);int loadBookDetailsFromFile(struct);void deleteBook(struct, int, int);

Declare an enumeration enum Category {FICTIONAL, PHYSICS, HISTORY} and Define a structure struct Book

{int bookNumber, char bookTitle[50],char author[50], int numPages, enum Category category, float bookCost

no

yes

If books[i].bookNumber == bookNumber

yes

no

Display book already exists

yes

Display Fictional, history, physics books in order

Call the function displayCategoryWiseList and pass the parameter

If choice =2

Display error opening file

BookCount++

Close the file

Write data to file using fwrite

If file != NULL

Declare a FILE \*file and open the file in "rb+" mode

8 file and open the file in "rb+" mode

Call the function saveBookDetailsToFile and pass parameters

Input book details

Set books[count].bookNumber = bookNumber

i++

yes

no

no

no

yes

If choice =3

no

yes

Call the function --- displayBookDetails and pass the parameters: books, bookCount, bookNumber

Declare and input bookNumber

Is choice=4?

no

yes

Display no books found in selected category

Display averageCost and totalCost.

Calculate averageCost=totalCost/bookCount

Is bookCount>0?

Match the books of give category, Add their costs to totalCost. Increment bookCount everytime a book of the asked category is found

Declare and Input categoryChoice

Call the function displayCategoryWiseTotalAndAverage and pass the parameters.

Search for book with same bookNumber entered.

Is choice=5?

Call the function deleteBook and pass the parameters: books, &bookCount, bookNumber.

Declare and input bookNumber

Display specified book number not found

Display details of book[i]

If books[i].bookNumber == bookNumber

Search for book with same bookNumber entered.

no

no

yes

Yes

no

Yes

If books[i].bookNumber == bookNumber

Replace details of book[i] with book [i+1] until i<count.

no

Display enter valid input

yes

no

Display exiting the program

Is choice=7?

Run the saveBookDetailsToFile function again

Set \*count=0.

yes

Call the function clearAllBooks and pass the parameters :books, &bookCount.

Is choice=6

bookCount--

Run the saveBookDetailsToFile

Function again

Display book has been deleted

Display specified book details not found.