LIBRARY MANAGEMENT SYSTEM. ABDULLAH (FA22-BCT-004).

SUMMARY:

The provided C++ code implements a comprehensive library management system designed to cater to both users and administrators. Users are empowered with functionalities such as borrowing and returning books, perusing the available book catalog, checking their borrowed book history, paying fines, and managing their login sessions. The code employs various classes, including Admin, User, Stack, Queue, and Binary Search Tree (BST), to efficiently organize and execute these diverse tasks. Notably, the inclusion of data structures such as linked lists for book and user management, stacks for fine payments, queues for book borrowing orders, and a BST for streamlined user ID management enhances the overall robustness of the system.

In parallel, the main function serves as the entry point for the program, initiating a user-friendly interface for both users and administrators. The authentication mechanism, reliant on arrays to store usernames and passwords, ensures secure access to the system. The program seamlessly integrates interactive panels for users and administrators, offering a user-centric experience for book-related activities and an admin-centric interface for more extensive library management tasks. The utilization of diverse data structures and well-defined classes underscores the program's efficiency and flexibility, providing a solid foundation for the implementation of a fully functional library management system.

LIST OF METHODS IN THE PROGRAM.

Admin Class:

- **admin_cred(const string &a, const string &b, const string c[], const string d[], int size)**: Authenticate the admin with provided credentials.
- **addBook(const std::string &bookID, const std::string &title, const std::string &author)**: Add a book to the library.
- **addUser(string userID, const string &username)**: Add a user to the system.
- **addTransaction(const std::string &uID, const std::string &bID, const std::string &dt, string borrowing)**: Add a transaction record.
- **applyChallan(string userID, double amount)**: Apply a fine (challan) to a user.
- **seeUsers()**: Display the list of users.
- **seeBooks()**: Display the list of books.
- **removeBook(const string &bookID)**: Remove a book from the library.
- **seeTransactions()**: Display the transaction history.
- **val_return(const string &a, const string c[], const string b[], int size)**: Validate and return a value based on input.

User Class:

- **user_cred(const string &a, const string &b, const string c[], const string d[], int size)**: Authenticate the user with provided credentials.
- **bookBorrow(int ID)**: Borrow a book by its ID.
- **returnBook(int ID)**: Return a borrowed book by its ID.
- **seeBookList(const string &filename)**: Display the list of available books from a file.
- **seeMyBooks()**: Display the list of books borrowed by the user.
- **payFine(double amount)**: Pay fines associated with the user.
- **calculateTotalFine()**: Calculate the total fine amount for the user.

Stack Class:

- **isEmpty() const**: Check if the stack is empty.
- **push(int value)**: Push a value onto the stack.

- **pop()**: Pop a value from the stack.
- **peek() const**: Get the top value of the stack without removing it.

Queue Class:

- **isEmpty() const**: Check if the queue is empty.
- **enqueue(int value)**: Enqueue a value into the queue.
- **dequeue()**: Dequeue a value from the queue.
- **peek() const**: Get the front value of the queue without removing it.

BST Class:

- **insert(int value)**: Insert a value into the binary search tree.
- **search(int value) const**: Search for a value in the binary search tree.
- **destroyTree(Node *&current)**: Destroy the binary search tree.
- **displayInorder(Node *current) const**: Display the values of the binary search tree in inorder traversal.