Name: Amandeep Singh

Roll no. : 2401010047

```
class BookNode:
    def __init__(self, book_id, title, author, status="Available"):
        self.book_id = book_id
        self.title = title
        self.author = author
        self.status = status
        self.next = None

class BookLinkedList:
    def __init__(self):
        self.head = None

    def insertBook(self, book_id, title, author, status="Available"):
        new_book = BookNode(book_id, title, author, status) if not
        self.head:
            self.head = new_book
        else:
            temp = self.head while
            temp.next: temp =
            temp.next temp.next =
            new_book
        print(f"Book '{title}' added successfully.")

    def deleteBook(self, book_id):
```

```
        temp = self.head prev = None
        while temp: if temp.book_id
        == book_id: if prev:
                prev.next = temp.next
            else:
                self.head = temp.next print(f"Book ID
            {book_id} deleted successfully.") return
          prev = temp temp =
        temp.next print("Book
        not found!")

    def searchBook(self, book_id):
        temp = self.head
        while temp:
            if temp.book_id == book_id:
                print(f"\nBook Found:\nID: {temp.book_id}\nTitle: {temp.title}\nAuthor:
{temp.author}\nStatus: {temp.status}") return temp temp = temp.next
print("Book not found!") return None

    def displayBooks(self):
        if not self.head:
            print("No books in the library.")
        return print("\nCurrent Books in
        Library:") temp = self.head while
        temp:
            print(f"ID: {temp.book_id}, Title: {temp.title}, Author: {temp.author}, Status: {temp.status}")
            temp = temp.next

class Stack:
    def __init__(self):
```

```
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self): if not
        self.is_empty():
            return self.items.pop()
        return None

    def is_empty(self):
        return len(self.items) == 0

    def display(self):
        if self.is_empty(): print("No
        transactions yet.") return
        print("\nRecent Transactions:") for
        transaction in reversed(self.items):
        print(transaction)

class TransactionSystem:
    def __init__(self):
        self.book_list = BookLinkedList()
        self.transaction_stack = Stack()

    def issueBook(self, book_id):
        book = self.book_list.searchBook(book_id)
        if book and book.status == "Available":
        book.status = "Issued"
        self.transaction_stack.push(("Issue",
```

```
    book_id)) print(f"Book ID {book_id} has
    been issued.")
    else: print("Book is not available or not
        found.")

def returnBook(self, book_id):
    book = self.book_list.searchBook(book_id)
    if book and book.status == "Issued":
        book.status = "Available"
        self.transaction_stack.push(("Return", book_id))
        print(f"Book ID {book_id} has been returned.")
    else: print("Book is not issued or not
        found.")

def undoTransaction(self):
    if self.transaction_stack.is_empty(): print("No
    transactions to undo.") return action,
    book_id = self.transaction_stack.pop() book =
    self.book_list.searchBook(book_id) if not
    book: print("Book not found.") return if
    action == "Issue":
        book.status = "Available" print(f"Undo successful: Book ID
    {book_id} is now Available.") elif action == "Return":
    book.status = "Issued" print(f"Undo successful: Book ID
    {book_id} is now Issued.")

def viewTransactions(self):
    self.transaction_stack.display()
```

```
def main(): system =
    TransactionSystem()

    while True:
        print("\n--- Library Book Management System ---")
        print("1. Add Book") print("2. Delete Book")
        print("3. Search Book") print("4. Display All
        Books") print("5. Issue Book") print("6. Return
        Book") print("7. Undo Last Transaction") print("8.
        View Transactions") print("9. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            book_id = int(input("Enter Book ID: ")) title =
        input("Enter Book Title: ") author = input("Enter
        Author Name: ")
        system.book_list.insertBook(book_id, title, author)
        elif choice == '2':
            book_id = int(input("Enter Book ID to delete: "))
        system.book_list.deleteBook(book_id) elif choice
        == '3':
            book_id = int(input("Enter Book ID to search: "))
            system.book_list.searchBook(book_id) elif choice
            == '4': system.book_list.displayBooks() elif choice
            == '5':
            book_id = int(input("Enter Book ID to issue: "))
        system.issueBook(book_id) elif choice == '6':
            book_id = int(input("Enter Book ID to return: "))
        system.returnBook(book_id) elif choice == '7':
```

```
        system.undoTransaction() elif choice == '8':
        system.viewTransactions() elif choice == '9':
            print("Exiting Library System. Goodbye!")
            break
        else: print("Invalid choice! Please try
            again.")

if __name__ == "__main__":
    main()
```