**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()