import pickle  
import os  
  
*# File paths*students\_file = 'students.pkl'  
books\_file = 'books.pkl'  
book\_issues\_file = 'book\_issues.pkl'  
  
*# Initialize files if they do not exist*if not os.path.exists(students\_file):  
 with open(students\_file, 'wb') as f:  
 pickle.dump({}, f)  
  
if not os.path.exists(books\_file):  
 with open(books\_file, 'wb') as f:  
 pickle.dump({}, f)  
  
if not os.path.exists(book\_issues\_file):  
 with open(book\_issues\_file, 'wb') as f:  
 pickle.dump({}, f)  
  
  
def load\_data(file\_path):  
 with open(file\_path, 'rb') as f:  
 return pickle.load(f)  
  
  
def save\_data(file\_path, data):  
 with open(file\_path, 'wb') as f:  
 pickle.dump(data, f)  
  
  
def create\_student\_record():  
 students = load\_data(students\_file)  
 student\_id = input("Enter student ID: ")  
 name = input("Enter student name: ")  
 course = input("Enter student course: ")  
 students[student\_id] = {'name': name, 'course': course}  
 save\_data(students\_file, students)  
 print("Student record created successfully!")  
  
  
def display\_all\_students():  
 students = load\_data(students\_file)  
 for student\_id, details in students.items():  
 print(f"ID: {student\_id}, Name: {details['name']}, Course: {details['course']}")  
  
  
def display\_specific\_student():  
 students = load\_data(students\_file)  
 student\_id = input("Enter student ID: ")  
 if student\_id in students:  
 print(f"ID: {student\_id}, Name: {students[student\_id]['name']}, Course: {students[student\_id]['course']}")  
 else:  
 print("Student not found")  
  
  
def modify\_student\_record():  
 students = load\_data(students\_file)  
 student\_id = input("Enter student ID: ")  
 if student\_id in students:  
 name = input("Enter new name: ")  
 course = input("Enter new course: ")  
 students[student\_id] = {'name': name, 'course': course}  
 save\_data(students\_file, students)  
 print("Student record updated successfully!")  
 else:  
 print("Student not found")  
  
  
def delete\_student\_record():  
 students = load\_data(students\_file)  
 student\_id = input("Enter student ID: ")  
 if student\_id in students:  
 del students[student\_id]  
 save\_data(students\_file, students)  
 print("Student record deleted successfully!")  
 else:  
 print("Student not found")  
  
  
def create\_book():  
 books = load\_data(books\_file)  
 book\_id = input("Enter book ID: ")  
 title = input("Enter book title: ")  
 author = input("Enter book author: ")  
 books[book\_id] = {'title': title, 'author': author}  
 save\_data(books\_file, books)  
 print("Book record created successfully!")  
  
  
def display\_all\_books():  
 books = load\_data(books\_file)  
 for book\_id, details in books.items():  
 print(f"ID: {book\_id}, Title: {details['title']}, Author: {details['author']}")  
  
  
def display\_specific\_book():  
 books = load\_data(books\_file)  
 book\_id = input("Enter book ID: ")  
 if book\_id in books:  
 print(f"ID: {book\_id}, Title: {books[book\_id]['title']}, Author: {books[book\_id]['author']}")  
 else:  
 print("Book not found")  
  
  
def modify\_book():  
 books = load\_data(books\_file)  
 book\_id = input("Enter book ID: ")  
 if book\_id in books:  
 title = input("Enter new title: ")  
 author = input("Enter new author: ")  
 books[book\_id] = {'title': title, 'author': author}  
 save\_data(books\_file, books)  
 print("Book record updated successfully!")  
 else:  
 print("Book not found")  
  
  
def delete\_book\_record():  
 books = load\_data(books\_file)  
 book\_id = input("Enter book ID: ")  
 if book\_id in books:  
 del books[book\_id]  
 save\_data(books\_file, books)  
 print("Book record deleted successfully!")  
 else:  
 print("Book not found")  
  
  
def book\_issue():  
 book\_issues = load\_data(book\_issues\_file)  
 books = load\_data(books\_file)  
 students = load\_data(students\_file)  
  
 issue\_id = input("Enter issue ID: ")  
 book\_id = input("Enter book ID: ")  
 student\_id = input("Enter student ID: ")  
 issue\_date = input("Enter issue date (YYYY-MM-DD): ")  
  
 if book\_id in books and student\_id in students:  
 book\_issues[issue\_id] = {'book\_id': book\_id, 'student\_id': student\_id, 'issue\_date': issue\_date}  
 save\_data(book\_issues\_file, book\_issues)  
 print("Book issued successfully!")  
 else:  
 print("Invalid book ID or student ID")  
  
  
def book\_deposit():  
 book\_issues = load\_data(book\_issues\_file)  
 issue\_id = input("Enter book issue ID: ")  
 if issue\_id in book\_issues:  
 del book\_issues[issue\_id]  
 save\_data(book\_issues\_file, book\_issues)  
 print("Book deposited successfully!")  
 else:  
 print("Issue ID not found")  
*# Functions to display all data*def display\_all\_data():  
 print("\nAll Students:")  
 display\_all\_students()  
 print("\nAll Books:")  
 display\_all\_books()  
 print("\nAll Book Issues:")  
 display\_all\_book\_issues()  
  
def display\_all\_book\_issues():  
 book\_issues = load\_data(book\_issues\_file)  
 for issue\_id, details in book\_issues.items():  
 print(f"Issue ID: {issue\_id}, Book ID: {details['book\_id']}, Student ID: {details['student\_id']}, Issue Date: {details['issue\_date']}")  
  
  
def main():  
 while True:  
 print("\nMenu:")  
 print("1. BOOK ISSUE")  
 print("2. BOOK DEPOSIT")  
 print("3. ADMINISTRATION MENU")  
 print(" 1. CREATE STUDENT RECORD")  
 print(" 2. DISPLAY ALL STUDENTS RECORD")  
 print(" 3. DISPLAY SPECIFIC STUDENT RECORD")  
 print(" 4. MODIFY STUDENT RECORD")  
 print(" 5. DELETE STUDENT RECORD")  
 print(" 6. CREATE BOOK")  
 print(" 7. DISPLAY ALL BOOKS")  
 print(" 8. DISPLAY SPECIFIC BOOK")  
 print(" 9. MODIFY BOOK")  
 print(" 10. DELETE BOOK RECORD")  
 print("4. EXIT")  
  
 choice = input("Enter your choice: ")  
  
 if choice == '1':  
 book\_issue()  
 elif choice == '2':  
 book\_deposit()  
 elif choice == '3':  
 sub\_choice = input("Enter administration choice (1-10): ")  
 if sub\_choice == '1':  
 create\_student\_record()  
 elif sub\_choice == '2':  
 display\_all\_students()  
 elif sub\_choice == '3':  
 display\_specific\_student()  
 elif sub\_choice == '4':  
 modify\_student\_record()  
 elif sub\_choice == '5':  
 delete\_student\_record()  
 elif sub\_choice == '6':  
 create\_book()  
 elif sub\_choice == '7':  
 display\_all\_books()  
 elif sub\_choice == '8':  
 display\_specific\_book()  
 elif sub\_choice == '9':  
 modify\_book()  
 elif sub\_choice == '10':  
 delete\_book\_record()  
 elif choice == '4':  
 break  
 else:  
 print("Invalid choice, please try again")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()