**ระบบจัดการห้องสมุด**

**Library book lending and return system**

**นางสาว สุกัญญา มาพระยืน รหัส 6806022510211 Sec 2**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต**

**สาขาวิศกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

**ปีการศึกษา 2568**

**ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

คำนำ

โครงการ “ระบบจัดการห้องสมุดไบนารี” จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อพัฒนาระบบจัดการห้องสมุดขนาดเล็กที่สามารถทำงานบน Command-Line Interface (CLI) โดยเน้นการเรียนรู้และสาธิตการจัดการไฟล์ข้อมูลด้วยภาษา Python ผ่านการใช้ ไฟล์ไบนารี (Binary File) และ โมดูล struct ซึ่งช่วยให้ผู้พัฒนาเข้าใจหลักการจัดเก็บข้อมูล การอ่าน-เขียนข้อมูลแบบไบนารี และการจัดการข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ

ระบบนี้ถูกออกแบบเพื่อช่วยให้ บรรณารักษ์ หรือเจ้าหน้าที่ผู้ดูแลระบบ สามารถทำงานประจำในห้องสมุดได้ง่ายและรวดเร็ว เช่น การเพิ่ม ลบ แก้ไข และค้นหาข้อมูลหนังสือและสมาชิก การบันทึกและติดตามประวัติการยืม-คืนหนังสือ รวมถึงการสร้างรายงานสรุปภาพรวมของห้องสมุด นอกจากนี้ โครงการยังมุ่งเน้นการพัฒนาฟังก์ชัน CRUD (Create, Read, Update, Delete) และการสร้างรายงาน (.txt) เพื่อให้ผู้ใช้สามารถตรวจสอบสถานะของระบบได้อย่างชัดเจน

ในการทำโครงการครั้งนี้ ผู้จัดทำได้ศึกษาแนวคิดและเทคนิคการจัดการไฟล์ไบนารี การใช้งาน struct ใน Python การจัดการสตริงและตัวเลข การจัดการวันเวลาแบบ Unix Timestamp รวมถึงการออกแบบเมนูและผังการทำงาน เพื่อให้ระบบสามารถทำงานได้อย่างถูกต้อง ครอบคลุมความต้องการของผู้ใช้งาน และรองรับข้อมูลหนังสือและสมาชิกในระดับ หลักพันระเบียน พร้อมข้อมูลการยืม-คืนในระดับ หลักหมื่นระเบียน

สุดท้ายนี้ ผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษาและผู้เกี่ยวข้องทุกท่านที่ให้คำแนะนำและความช่วยเหลือ ทำให้โครงการนี้สามารถสำเร็จลุล่วงตามวัตถุประสงค์และสามารถนำไปใช้ประโยชน์ได้จริง

สารบัญ

**หน้า**

คำนำ ก

สารบัญ ข

สารบัญ(ต่อ) ค

สารบัญรูปภาพ ง

สารบัญรูปภาพ(ต่อ) จ

สารบัญตาราง ฉ

บทที่ 1 บทนำ 7

1.1 วัตถุประสงค์ของโครงงาน 7

1.2 ขอบเขตของโครงาน 7

1.3 ประโยชน์ที่ได้รับ 8

1.4 เครื่องมือที่คาดว่าจะต้องใช้ 8

บทที่ 2 ระบบจัดการห้องสมุด 9

2.1 แฟ้มข้อมูลหนังสือ books.dat 9

2.2 แฟ้มข้อมูลสมาชิก members.dat 10

2.3 แฟ้มข้อมูลการยืม-คืน lendings.dat 11

2.4 ไฟล์ library\_report.txt 12

บทที่ 3 การใช้งานระบบจัดการห้องสมุด 14

3.1 การใช้งานระบบจัดการห้องสมุด 14

ใช้สำหรับออกจากเมนูหนังสือและกลับไปยังเมนูหลัก 15

3.2 การใช้งานโปรแกรมเพิ่มข้อมูล 18

3.3 การใช้งานโปรแกรมแสดงข้อมูล 19

3.4 การใช้งานโปรแกรมแก้ใขข้อมูล 20

3.5 การใช้งานโปรแกรมลบข้อมูล 22

3.6 การใช้งานโปรแกรมยืม – คืนหนังสือ 23

บทที่ 4 อธิบายการทำงานของ Code 26

4.1 ภาพรวมของระบบ 26

4.2 ฟังก์ชันหลักในแต่ละโมดูล 26

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 43

5.1 สรุปผลการดำเนินงาน 43

5.2 ปัญหาและอุปสรรคในการดำเนินงาน 43

5.3 ข้อเสนอแนะ 44

5.4 สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน 44

สารบัญรูปภาพ

**หน้า**

[**รูปภาพที่** 2‑1 library\_report.txt12](#_Toc210495190)

[**รูปภาพที่** 3‑1 **การเลือกใช้งานฟังก์ชั่น** 14](#_Toc210495191)

[**รูปภาพที่** 3‑2 **เมนูของ** Book15](#_Toc210495192)

[**รูปภาพที่** 3‑3 **การเลือกเมนูของ** Member15](#_Toc210495193)

[**รูปภาพที่** 3‑4 **เมนูของยืม-คืน** 16](#_Toc210495194)

[**รูปภาพที่** 3‑5 **หัวข้อการสร้างรายงาน** 17](#_Toc210495195)

[**รูปภาพที่** 3‑6 **หัวข้อการปิดใช้งานโปรแกรม** 18](#_Toc210495196)

[**รูปภาพที่** 3‑7 **การเลือกเมนูจัดการหนังสือ** 18](#_Toc210495197)

[**รูปภาพที่** 3‑8 **การเพิ่มหนังสือ** 18](#_Toc210495198)

[**รูปภาพที่** 3‑9 **การเพิ่มสมาชิก** 19](#_Toc210495199)

[**รูปภาพที่** 3‑10 **การดูหนังสือทั้งหมด** 19](#_Toc210495200)

[**รูปภาพที่** 3‑11 **รายการหนังสือทั้งหมด** 19](#_Toc210495201)

[**รูปภาพที่** 3‑12 **รายการสมาชิกทั้งหมด** 20](#_Toc210495202)

[**รูปภาพที่** 3‑13 **การดูประวัติยืม - คืน** 20](#_Toc210495203)

[**รูปภาพที่** 3‑14 **แสดงรายการยืม - คืน** 20](#_Toc210495204)

[**รูปภาพที่** 3‑15 **การแก้ใขเมนูหนังสือ** 21](#_Toc210495205)

[**รูปภาพที่** 3‑16 **แสดงตัวอย่างการแก้ใขหนังสือ** 21](#_Toc210495206)

[**รูปภาพที่** 3‑17 **การแก้ใขข้อมูลสมาชิก** 21](#_Toc210495207)

[**รูปภาพที่** 3‑18 **ตัวอย่างการแก้ใขข้อมูลสมาชิก** 22](#_Toc210495208)

[**รูปภาพที่** 3‑19 **การลบข้อมูลหนังสือ** 22](#_Toc210495209)

[**รูปภาพที่** 3‑20 **ตัวอย่างการลบหนังสือ** 22](#_Toc210495210)

[**รูปภาพที่** 3‑21 **การลบข้อมูลสมาชิก** 23](#_Toc210495211)

[**รูปภาพที่** 3‑22 **ตัวอย่างการลบข้อมูลสมาชิก** 23](#_Toc210495212)

[**รูปภาพที่** 3‑23 **การคืนหนังสือ** 23](#_Toc210495213)

[**รูปภาพที่** 3‑24 **การกรอกข้อมูลของการยืมหนังสือ** 24](#_Toc210495214)

[**รูปภาพที่** 3‑25 **การใช้งานโปรแกรมการคืนหนังสือ** 24](#_Toc210495215)

[**รูปภาพที่** 3‑26 **การกรอกข้อมูลของการคืนหนังสือ** 24](#_Toc210495216)

[**รูปภาพที่** 3‑27 **การดูประวัติการยืม - คืนทั้งหมด** 24](#_Toc210495217)

[**รูปภาพที่** 3‑28 **ข้อมูลของการยืม - คืนหนังสือทั้งหมด** 25](#_Toc210495218)

[**รูปภาพที่** 4‑1 **ฟังก์ชั่น** add\_book27](#_Toc210495219)

[**รูปภาพที่** 4‑2 **ฟังก์ชัน** update\_book29](#_Toc210495220)

[**รูปภาพที่** 4‑3 **ฟังก์ชัน** delete\_book31](#_Toc210495221)

[**รูปภาพที่** 4‑4 **ฟังก์ชัน** add\_member32](#_Toc210495222)

[**รูปภาพที่** 4‑5 **ฟังก์ชัน** view\_all\_members33](#_Toc210495223)

[**รูปภาพที่** 4‑6 **ฟังก์ชัน** view\_all\_members34](#_Toc210495224)

[**รูปภาพที่** 4‑7 **ฟังก์ชัน** delete\_member35](#_Toc210495225)

[**รูปภาพที่** 4‑8 **ฟังก์ชัน** borrow\_book36](#_Toc210495226)

[**รูปภาพที่** 4‑9 **ฟังก์ชัน** return\_book37](#_Toc210495227)

[**รูปภาพที่** 4‑10 **ฟังก์ชัน** generate\_report40](#_Toc210495228)

[**รูปภาพที่** 4‑11 **ฟังก์ชัน** main42](#_Toc210495229)

****

สารบัญตาราง

**หน้า**

[**ตารางที่ 2.1 แฟ้มข้อมูลหนังสือ** 9](#_Toc210372847)

[**ตารางที่** 2.2 **แฟ้มข้อมูลสมาชิก** 10](#_Toc210372848)

[**ตารางที่** 2.3 **แฟ้มข้อมูลการยืม-คืน** 11](#_Toc210372849)

****

# บทนำ

## วัตถุประสงค์ของโครงงาน

1.1.1 เพื่อพัฒนาระบบจัดการห้องสมุดขนาดเล็กที่สามารถทำงานผ่านCommand-Line Interface (CLI) และรองรับการจัดเก็บข้อมูลแบบไฟล์ไบนารี

1.1.2 เพื่อให้ผู้ใช้งานหลัก เช่น บรรณารักษ์ หรือเจ้าหน้าที่ผู้ดูแลระบบ สามารถเพิ่ม ลบ แก้ไข และค้นหาข้อมูลหนังสือและสมาชิกได้อย่างสะดวกและรวดเร็ว

1.1.3 เพื่อสนับสนุนการ บันทึกและติดตามประวัติการยืม-คืนหนังสือ อย่างเป็นระบบและแม่นยำ

1.1.4 เพื่อสร้าง รายงานสรุปภาพรวมของห้องสมุด เช่น ข้อมูลหนังสือทั้งหมด ข้อมูลสมาชิก และรายการการยืม-คืน เพื่อช่วยในการวิเคราะห์และบริหารจัดการห้องสมุด

1.1.5 เพื่อให้ผู้พัฒนามีโอกาสเรียนรู้และฝึกฝนทักษะการจัดการข้อมูล ไฟล์ไบนารี, การใช้ struct ใน Python, การจัดการสตริงและตัวเลข, และการออกแบบระบบ CRUD อย่างเป็นระบบ

## ขอบเขตของโครงาน

1.2.1 ระบบจัดการห้องสมุดมีฟังก์ชั่นพื้นฐานทั้งหมด 14 ฟังก์ชั่น เช่น

1. เพิ่มหนังสือ

2. แก้ใขหนังสือ

3. แสดงหนังสือทั้งหมด

4. ลบหนังสือ

5. กลับไปที่เมนู

6. เพิ่มสมาชิก

7. ลบสมาชิก

8. แก้ใขสมาชิก

9. แสดงสมาชิกทั้งหมด

10. ยืมหนังสือ

11. คืนหนังสือ

12. ดูประวัติทั้งหมด

13. สร้างรายงาน

14. ออกจากโปรแกรม

1.2.2 ระบบจัดการห้องสมุดประกอบด้วย 4 ไฟล์ ได้แก่

1. แฟ้มข้อมูลหนังสือ books.dat

2. แฟ้มข้อมูลสมาชิก members.dat

3. แฟ้มข้อมูลการยืม-คืน lendings.dat

4. ไฟล์ library\_report.txt

1.2.3 ระบบจัดการห้องสมุดมีการจัดเก็บ ข้อมูลสรุปสถานะล่าสุดของห้องสมุด ลงในไฟล์ข้อความ (Text File) ชื่อ library\_report.txt โดยมีรายละเอียดดังนี้:

1.2.3.1 ข้อมูลที่บันทึกในไฟล์รายงาน

1.2.3.2 ข้อมูลหนังสือทั้งหมด เช่น รหัสหนังสือ ชื่อหนังสือ และจำนวนเล่มที่มีอยู่

1.2.3.3 ข้อมูลสมาชิกทั้งหมด เช่น รหัสสมาชิก และชื่อ-สกุล เบอร์โทร

1.2.3.4 ข้อมูลการยืม-คืนทั้งหมด เช่น รหัสการยืม รหัสหนังสือ รหัสสมาชิก วันที่ยืม และวันที่คืน

## ประโยชน์ที่ได้รับ

1.3.1 ช่วยให้สามารถจัดการข้อมูลหนังสือและสมาชิกได้อย่างสะดวก รวดเร็ว และแม่นยำ

1.3.2 ฝึกทักษะการเขียนโปรแกรมภาษา Python สำหรับระบบจัดการข้อมูลแบบ Command-Line Interface (CLI)

1.3.3 เข้าใจหลักการ CRUD (Create, Read, Update, Delete) และการจัดการข้อมูลใน ไฟล์ไบนารี (Binary File)

## เครื่องมือที่คาดว่าจะต้องใช้

1.4.1 โปรแกรม Visual Studio Code

# **ระบบจัดการห้องสมุด**

## แฟ้มข้อมูลหนังสือ books.dat

แฟ้มนี้ใช้จัดเก็บข้อมูลเกี่ยวกับหนังสือทั้งหมดในห้องสมุด แต่ละระเบียน (Record) ของหนังสือมีโครงสร้างดังนี้:

|  |  |  |  |
| --- | --- | --- | --- |
| ฟิลด์ | ชนิด | ขนาด | ตัวอย่าง |
| is\_active | char | 1 | b'A' |
| book\_id | integer | 4 | 1001 |
| isbn | |  | | --- | |  |  |  | | --- | | string | | 16 | |  | | --- | |  |  |  | | --- | | '978-0134685991' | |
| title | string | 128 | 'Python เบื้องต้น' |
| author | string | 64 | 'สุกัญญา มาพระยืน' |
| quantity | short | 2 | 5 |

ตารางที่ **2.1** แฟ้มข้อมูลหนังสือ

2.1.1 is\_active สถานะ

ใช้ตรวจสอบว่าหนังสือยังใช้งานอยู่หรือถูกลบ เพื่อให้โปรแกรมไม่แสดงหรือให้ยืมหนังสือที่ถูกลบแล้ว ตัวอย่าง: หนังสือ “Python เบื้องต้น” ถูกลบ is\_active = 'D' เมนูค้นหาหนังสือจะไม่แสดงรายการนี้

2.1.2 book\_id รหัสหนังสือ

ใช้ระบุเอกลักษณ์ของหนังสือแต่ละเล่ม ใช้ค้นหา แก้ไข หรือลบ ตัวอย่าง: book\_id = 1001 ใช้ค้นหาหนังสือ “Python เบื้องต้น” เพื่อแก้ไขจำนวนเล่ม

2.2.3 isbn รหัส ISBN

ใช้ตรวจสอบและค้นหาหนังสือตามรหัสมาตรฐาน ตัวอย่าง: isbn = "978-616-182-012-3" ระบบค้นหาหนังสือโดยตรงโดยไม่ต้องพิมพ์ชื่อ

2.2.4 title ชื่อหนังสือ

ใช้แสดงชื่อหนังสือและค้นหา ตัวอย่าง: ผู้ใช้พิมพ์ “Python” → ระบบแสดงหนังสือที่ชื่อมีคำว่า “Python” ทั้งหมด

2.2.5 author ชื่อผู้แต่ง

ใช้ค้นหาและแสดงข้อมูลผู้แต่ง ตัวอย่าง**:** ผู้ใช้ค้นหา author = “สุกัญญา มาพระยืน” ระบบแสดงหนังสือทุกเล่มที่ผู้แต่งคนนี้เขียน

2.2.6 quantity จำนวนเล่ม

ใช้ตรวจสอบจำนวนหนังสือในสต็อกก่อนให้ยืม และคำนวณจำนวนหลังยืม/คืน  
ตัวอย่าง: quantity = 5 สมาชิกยืม 1 เล่ม quantity ลดเหลือ 4

## แฟ้มข้อมูลสมาชิก members.dat

แฟ้มนี้ใช้จัดเก็บข้อมูลเกี่ยวกับข้อมูลสมาชิกทั้งหมด แต่ละระเบียน (Record) โครงสร้างดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| ฟิลด์ | ชนิด | ขนาด | ตัวอย่าง |
| is\_active | Char | 1 | b'A' |
| member\_id | integer | 4 | 2001 |
| name | string | 64 | 'สุกัญญา มาพระยืน' |
| phone | number | 16 | 011-1111-1111 |

ตารางที่ **2.2** **แฟ้มข้อมูลสมาชิก**

2.2.1 is\_active สถานะสมาชิก

ตรวจสอบว่าสมาชิกยังใช้งานอยู่หรือถูกลบ ตัวอย่าง: member\_id = 2003, is\_active = 'D' ระบบปฏิเสธการยืมหนังสือ

2.2.2 member\_id รหัสสมาชิก

ใช้ระบุสมาชิกแต่ละคนไม่ซ้ำกัน ใช้ค้นหา แก้ไข หรือลบ ตัวอย่าง: member\_id = 2001 ระบบเรียกดูชื่อสมาชิก = “สุกัญญา มาพระยืน”

2.2.3 name ชื่อ-สกุลสมาชิก

|  |
| --- |
| แสดงชื่อสมาชิกและค้นหา ตัวอย่าง: ผู้ใช้ค้นหาชื่อ “สุกัญญา” ระบบแสดงรายละเอียดสมาชิกและประวัติการยืม |

2.2.4 phone เบอร์โทรสมาชิก

ติดต่อสมาชิกกรณีหนังสือยังไม่คืน

## แฟ้มข้อมูลการยืม-คืน lendings.dat

แฟ้มนี้ใช้จัดเก็บข้อมูลเกี่ยวกับการยืม - คืน แต่ละระเบียน (Record) โครงสร้างดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| ฟิลด์ | ชนิด | ขนาด | ตัวอย่าง |
| is\_active | char | 1 | b'A' |
| lending\_id | integer | 4 | 3001 |
| book\_id | integer | 4 | 1002 |
| member\_id | integer | 4 | 2001 |
| borrow\_date | double | 8 | 1694332800.0 |
| return\_date | double | 8 | 0.0 (ยังไม่คืน) |

ตารางที่ **2.3** **แฟ้มข้อมูลการยืม-คืน**

2.3.1 is\_active สถานะสมาชิก

ตรวจสอบว่าสมาชิกยังใช้งานอยู่หรือถูกลบ ตัวอย่าง: member\_id = 2003, is\_active = 'D' ระบบปฏิเสธการยืมหนังสือ

2.3.2 lending\_id ระบุรายการการยืม

ระบุรายการการยืมแต่ละรายการ ตัวอย่าง: lending\_id = 3001 ใช้คืนหนังสือ “Computer Networks” ของ member\_id = 2001

2.3.3 book\_id รหัสหนังสือ

|  |
| --- |
|  |

|  |
| --- |
| เชื่อมรายการการยืมกับหนังสือ ตัวอย่าง: book\_id = 1002 ระบบลด quantity ของหนังสือเล่มนั้นเมื่อยืมออก |

2.3.4 member\_id รหัสสมาชิก

เชื่อมรายการการยืมกับสมาชิก ตัวอย่าง: member\_id = 2001 ระบบบันทึกว่า “สุกัญญา” ยืมหนังสือเล่มนี้

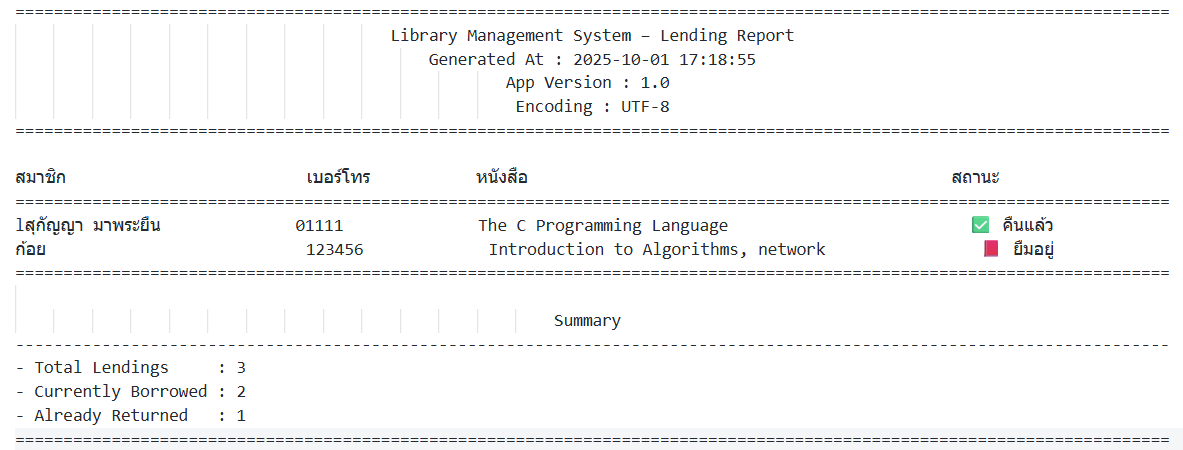
2.3.5 borrow\_date วันที่ยืม

บันทึกวันที่เริ่มยืม ใช้ตรวจสอบกำหนดวันคืน ตัวอย่าง: borrow\_date = 2025-09-10 → ใช้คำนวณกำหนดคืนภายใน 7 วัน

2.3.6 return\_date วันที่คืน

บันทึกวันที่คืนหนังสือ หรือ 0.0 ถ้ายังไม่คืน ตัวอย่าง: return\_date = 0.0 → ระบบแสดง “ยังไม่คืน” ในรายงาน

## ไฟล์ library\_report.txt

****

รูปภาพที่ ‑ library\_report.txt

2.4.1 header\_text ส่วนหัวรายงาน

แสดงชื่อรายงานเพื่อให้ผู้ใช้เข้าใจว่าข้อมูลด้านล่างคือรายงานการยืม-คืน

2.4.2 generated\_at วันที่และเวลาที่รายงาน

ใช้ตรวจสอบว่าไฟล์รายงานนี้อัปเดตล่าสุดเมื่อไร

2.4.3 app\_version เวอร์ชั่นโปรแกรม 1.0

ใช้ระบุว่าไฟล์รายงานนี้ถูกสร้างจากเวอร์ชันใดของโปรแกรม (เพื่อการอัปเกรดหรือ Debug)

2.4.4 encoding การเข้ารหัสไฟล์

เพื่อให้มั่นใจว่าอักขระภาษาไทย/อังกฤษจะแสดงผลถูกต้อง

|  |
| --- |
|  |

|  |
| --- |
|  |

2.4.5 book\_table\_ header หัวตาราง

ทำให้ผู้ใช้เข้าใจข้อมูลแต่ละคอลัมน์

2.4.6 book\_records ข้อมูลตาราง

ใช้บอกว่าใครยืมหนังสืออะไร และยังไม่คืนหรือคืนแล้ว

2.4.7 total\_lendings

เพื่อแสดงจำนวนธุรกรรมการยืมทั้งหมดในรายงานนี้

2.4.8 currently\_borrowed

ใช้ตรวจสอบจำนวนหนังสือที่ยังอยู่นอกห้องสมุด

2.4.9 already\_returned

ใช้ตรวจสอบว่ามีการคืนสำเร็จไปแล้วกี่ครั้ง

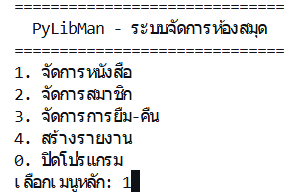
# การใช้งานระบบจัดการห้องสมุด

ระบบจัดการห้องสมุดที่พัฒนาขึ้นนี้มีวัตถุประสงค์เพื่ออำนวยความสะดวกในการบริหารจัดการข้อมูลหนังสือ ข้อมูลสมาชิก และการยืม–คืนหนังสือให้มีความถูกต้อง รวดเร็ว และลดความซ้ำซ้อนในการทำงาน ระบบถูกออกแบบมาให้ใช้งานง่ายผ่านเมนูหลักที่ชัดเจน โดยแบ่งการทำงานออกเป็นโมดูล เช่น การจัดการหนังสือ การจัดการสมาชิก การจัดการการยืม–คืน และการออกรายงาน

ผู้ใช้สามารถเริ่มต้นใช้งานระบบได้ตั้งแต่การเพิ่มข้อมูลหนังสือและสมาชิกเข้าสู่ฐานข้อมูล จากนั้นจึงสามารถทำการบันทึกการยืม–คืนได้ ระบบจะจัดเก็บข้อมูลทั้งหมดในรูปแบบไฟล์ไบนารีเพื่อความรวดเร็วในการเข้าถึง และยังสามารถสร้างรายงานสรุปผล เช่น จำนวนการยืม–คืนทั้งหมด รายการหนังสือที่กำลังถูกยืม และรายการที่ถูกส่งคืนแล้ว

**สำหรับผู้ใช้งานโปรแกรม**

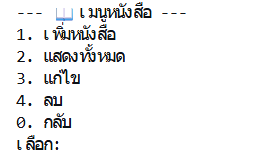
## การใช้งานระบบจัดการห้องสมุด



รูปภาพที่ **3‑1** **การเลือกใช้งานฟังก์ชั่น**

### 3.1.1 เมนูจัดการหนังสือ

เมนูนี้ใช้สำหรับจัดเก็บและปรับปรุงข้อมูลหนังสือในระบบ โดยมีตัวเลือกดังนี้



รูปภาพที่ **3‑2** **เมนูของ** Book

3.1.1.1 เพิ่มหนังสือ

ผู้ใช้สามารถกรอกข้อมูลหนังสือใหม่ เช่น รหัสหนังสือ, ชื่อหนังสือ, ผู้แต่ง, จำนวนหนังสือ เมื่อกดบันทึก ข้อมูลจะถูกเก็บลงในแฟ้ม books.dat

3.1.1.2 แสดงทั้งหมด

แสดงรายชื่อหนังสือทั้งหมดในระบบ ไม่ว่าจะกำลังว่างหรือถูกยืมไปแล้ว ช่วยให้ผู้ใช้หรือผู้ดูแลรู้ว่ามีหนังสืออะไรอยู่ในห้องสมุดบ้าง

3.1.1.3 แก้ใข

ใช้ปรับปรุงข้อมูลหนังสือที่บันทึกไว้ เช่น แก้ชื่อหนังสือ หรือแก้ชื่อผู้แต่ง

3.1.1.4 ลบ

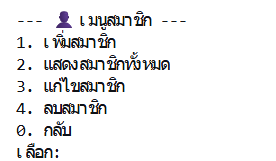
ใช้ลบข้อมูลหนังสือออกจากระบบ เช่น หนังสือสูญหายหรือชำรุด

3.1.1.3 กลับ

## ใช้สำหรับออกจากเมนูหนังสือและกลับไปยังเมนูหลัก

3.1.2 เมนูจัดการสมาชิก

เมนูนี้ใช้สำหรับจัดเก็บและปรับปรุงข้อมูลสมาชิกในระบบ โดยมีตัวเลือกดังนี้



รูปภาพที่ **3‑3** **การเลือกเมนูของ** Member

3.1.2.1 เพิ่มสมาชิก

ผู้ใช้สามารถกรอกข้อมูลสมาชิกใหม่ เช่น รหัสสมาชิก, ชื่อ-นามสกุล, ที่อยู่, เบอร์โทรศัพท์ เมื่อกดบันทึก ข้อมูลจะถูกเก็บลงในแฟ้ม members.dat

3.1.2.2 แสดงสมาชิก

แสดงรายชื่อสมาชิกทั้งหมดในระบบ ไม่ว่าจะเป็นสมาชิกที่ใช้งานอยู่หรือไม่ได้ใช้งาน ช่วยให้ผู้ใช้หรือผู้ดูแลทราบข้อมูลสมาชิกทั้งหมดที่ลงทะเบียนไว้

3.1.2.3 แก้ใขสมาชิก

ใช้ปรับปรุงข้อมูลสมาชิกที่บันทึกไว้ เช่น แก้ชื่อ-นามสกุล, ที่อยู่ หรือเบอร์โทรศัพท์ กรณีที่สมาชิกมีการเปลี่ยนแปลงข้อมูล

3.1.2.4 ลบสมาชิก

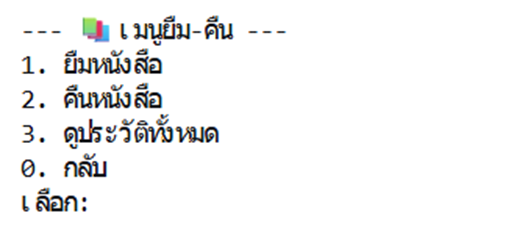
ใช้ลบข้อมูลสมาชิกออกจากระบบ เช่น สมาชิกลาออก หรือยกเลิกการเป็นสมาชิก (ควรตรวจสอบว่าไม่มีหนังสือค้างยืมก่อนลบ)

3.1.2.5 กลับ

ใช้สำหรับออกจากเมนูสมาชิกและกลับไปยังเมนูหลัก

3.1.3 เมนูจัดการยืม – คืน

เมนูนี้ใช้สำหรับจัดการการยืมและคืนหนังสือของสมาชิก โดยมีตัวเลือกดังนี้



รูปภาพที่ **3‑4 เมนูของยืม-คืน**

3.1.3.1 ยืมหนังสือผู้ใช้สามารถบันทึกการยืมหนังสือโดยระบุรหัสสมาชิกและรหัสหนังสือ ระบบจะบันทึกวันที่ยืมและกำหนดวันที่ต้องคืน พร้อมทั้งอัพเดทสถานะหนังสือเป็น "กำลังถูกยืม" ข้อมูลจะถูกเก็บลงในแฟ้ม borrow.dat

3.1.3.2 คืนหนังสือ

ใช้บันทึกการคืนหนังสือเมื่อสมาชิกนำหนังสือมาคืน โดยระบุรหัสสมาชิกและรหัสหนังสือที่ต้องการคืน ระบบจะบันทึกวันที่คืนและเปลี่ยนสถานะหนังสือเป็น "ว่าง" พร้อมคำนวณค่าปรับ (ถ้ามี) กรณีคืนเกินกำหนด

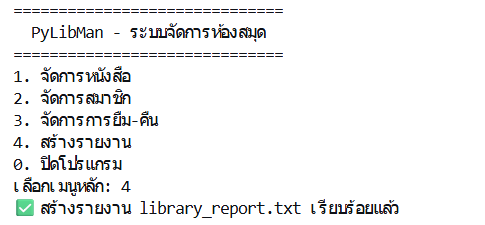
3.1.3.3 ดูประวัติทั้งหมด

แสดงประวัติการยืม-คืนหนังสือทั้งหมดในระบบ หรือค้นหาตามรหัสสมาชิก เพื่อตรวจสอบว่าสมาชิกท่านใดยืมหนังสือเล่มไหนไปบ้าง และมีหนังสือค้างคืนหรือไม่

3.1.3.4 กลับ

ใช้สำหรับออกจากเมนูยืม-คืนและกลับไปยังเมนูหลัก

3.1.4 สร้างรายงาน

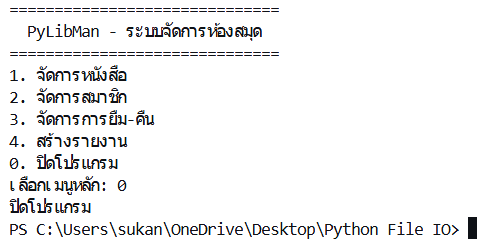


รูปภาพที่ **3‑5** **หัวข้อการสร้างรายงาน**

เมนูนี้ใช้สำหรับสร้างรายงานสรุปข้อมูลต่างๆ ในระบบห้องสมุด

เมื่อผู้ใช้เลือกเมนูหมายเลข 4 ระบบจะสร้างรายงานและบันทึกลงในไฟล์ library\_report.txt โดยอัตโนมัติ จากนั้นจะแสดงข้อความ "สร้างรายงาน library\_report.txt เรียบร้อยแล้ว" เพื่อแจ้งให้ผู้ใช้ทราบว่าการสร้างรายงานสำเร็จ

3.1.5 ปิดโปรแกรม



รูปภาพที่ **3‑6** **หัวข้อการปิดใช้งานโปรแกรม**

เมื่อผู้ใช้เลือกตัวเลือก 0. ปิดโปรแกรม จากเมนูหลัก ระบบจะทำการดังนี้

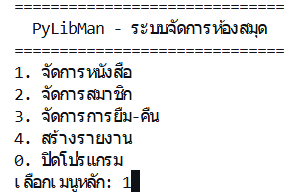
3.1.5.1 บันทึกข้อมูลทั้งหมดที่มีการเปลี่ยนแปลงลงในแฟ้มข้อมูล(books.dat,members.dat, borrow.dat)

3.1.5.2 แสดงข้อความ "ขอบคุณที่ใช้บริการระบบห้องสมุด PyLibMan" หรือข้อความอำลาที่เหมาะสม

3.1.5.3 ปิดโปรแกรมและสิ้นสุดการทำงาน

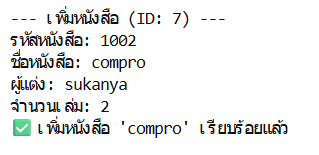
## การใช้งานโปรแกรมเพิ่มข้อมูล

3.2.1 เมื่อเลือกเมนูหมายเลข 1 จัดการหนังสือ



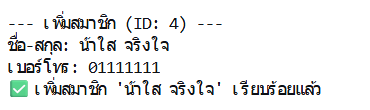
รูปภาพที่ **3‑7** **การเลือกเมนูจัดการหนังสือ**

3.2.2 และเลือกเมนูหมายเลข 1 เพิ่มหนังสือ จะปรากฎหัวข้อการกรอกข้อมูลของหนังสือที่จะเพิ่ม เช่น รหัสหนังสือ ชื่อหนังสือ ผู้แต่ง จำนวนเล่ม



รูปภาพที่ **3‑8** **การเพิ่มหนังสือ**

3.2.3 เมื่อเลือกเมนูหมายเลข 2 จัดการสมาชิก

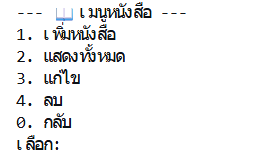


รูปภาพที่ **3‑9** **การเพิ่มสมาชิก**

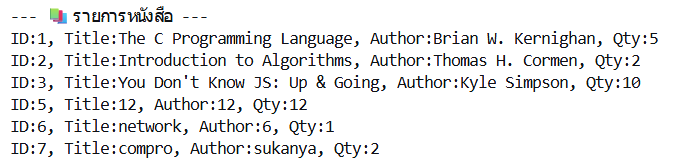
3.2.3 เมื่อเลือกเมนูหมายเลข 1 เพิ่มสมาชิก จะปรากฎหัวข้อการกรอกข้อมูลของสมาชิกที่จะเพิ่ม เช่น ชื่อ - สกุล เบอร์โทร

## การใช้งานโปรแกรมแสดงข้อมูล

3.3.1 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการหนังสือ กรอกหมายเลข 2 เพื่อดูเนื้อหาของหนังสือทั้งหมด



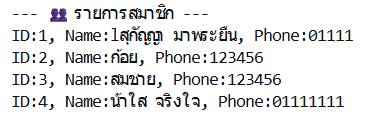
รูปภาพที่ **3‑10** **การดูหนังสือทั้งหมด**

 3.3.2 เมื่อกรอกหมายเลข 2 แล้ว จะปรากฏข้อมูลของหนังสือทั้งหมด

รูปภาพที่ **3‑11** **รายการหนังสือทั้งหมด**

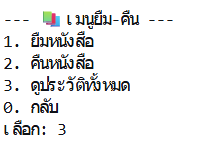
3.3.3 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการสมาชิก กรอกหมายเลข 2 เพื่อดูเนื้อหาของสมาชิกทั้งหมด

3.3.4 เมื่อกรอกหมายเลข 2 แล้ว จะปรากฏข้อมูลของสมาชิกทั้งหมด

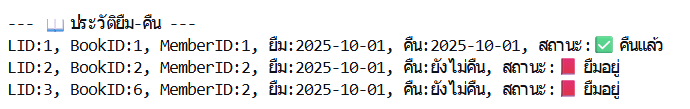


รูปภาพที่ **3‑12 รายการสมาชิกทั้งหมด**

3.3.5 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการการยืม - คืน กรอกหมายเลข 3 เพื่อดูเนื้อหาของการยืม - คืน ทั้งหมด



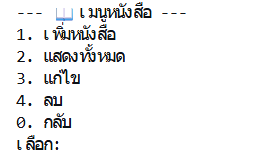
รูปภาพที่ **3‑13** **การดูประวัติยืม - คืน**

3.3.6 เมื่อกรอกหมายเลข 3 แล้ว จะปรากฏข้อมูลของการยืม – คืนทั้งหมด

รูปภาพที่ **3‑14** **แสดงรายการยืม - คืน**

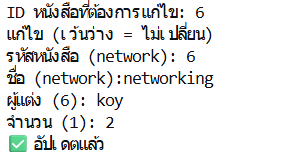
## การใช้งานโปรแกรมแก้ใขข้อมูล

3.4.1 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการหนังสือ กรอกหมายเลข 3 การใช้งานโปรแกรมแก้ใขข้อมูลหนังสือ



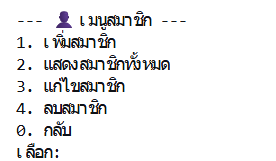
รูปภาพที่ **3‑15** **การแก้ใขเมนูหนังสือ**

3.4.2 เมื่อกรอกหมายเลข 3 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของหนังสือที่จะแก้ใข เช่น รหัสหนังสือ ชื่อ ผู้แต่ง จำนวน



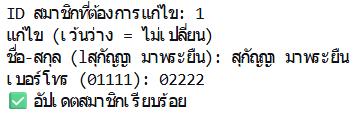
รูปภาพที่ **3‑16 แสดงตัวอย่างการแก้ใขหนังสือ**

3.4.3 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการสมาชิก กรอกหมายเลข 3 การใช้งานโปรแกรมแก้ใขข้อมูลสมาชิก



รูปภาพที่‑ **การแก้ใขข้อมูลสมาชิก**

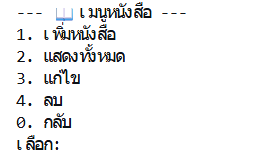
3.4.4 เมื่อกรอกหมายเลข 3 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของสมาชิกที่จะแก้ใข เช่น ชื่อ - สกุล เบอร์โทร



รูปภาพที่ **3‑18** **ตัวอย่างการแก้ใขข้อมูลสมาชิก**

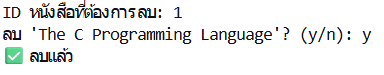
## การใช้งานโปรแกรมลบข้อมูล

3.5.1 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการหนังสือ กรอกหมายเลข 4 การใช้งานโปรแกรมลบข้อมูลหนังสือ



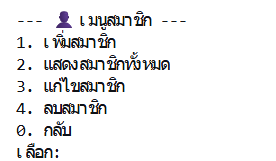
รูปภาพที่ **3‑19** **การลบข้อมูลหนังสือ**

3.5.2 เมื่อกรอกหมายเลข 4 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของหนังสือที่จะลบ เช่น รหัสหนังสือ และมีการยืนยันในการลบข้อมูล



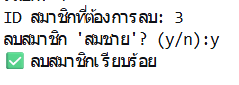
รูปภาพที่‑ **ตัวอย่างการลบหนังสือ**

3.5.3 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการสมาชิก กรอกหมายเลข 4 การใช้งานโปรแกรมลบข้อมูลสมาชิก



รูปภาพที่ **3‑21** **การลบข้อมูลสมาชิก**

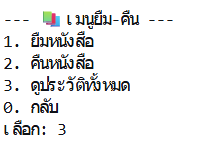
3.5.3 เมื่อกรอกหมายเลข 4 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของสมาชิกที่จะลบ เช่น รหัสสมาชิก และมีการยืนยันในการลบข้อมูล



รูปภาพที่ **3‑22** **ตัวอย่างการลบข้อมูลสมาชิก**

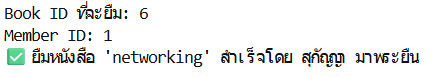
## การใช้งานโปรแกรมยืม – คืนหนังสือ

### 3.6.1 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการการยืม - คืน กรอกหมายเลข 1 การใช้งานโปรแกรมการยืมหนังสือ



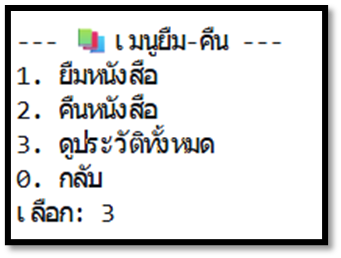
รูปภาพที่ **3‑23** **การคืนหนังสือ**

### 3.6.2 เมื่อกรอกหมายเลข 1 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของการยืมหนังสือที่จะยืม เช่น รหัสหนังสือ รหัสสมาชิก



รูปภาพที่ **3‑24** **การกรอกข้อมูลของการยืมหนังสือ**

### 3.6.3 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการการยืม - คืน กรอกหมายเลข 2 การใช้งานโปรแกรมการคืนหนังสือ



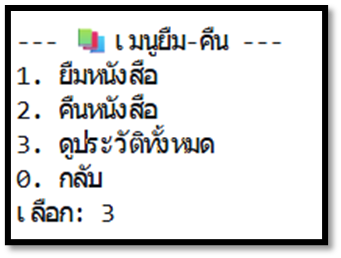
### รูปภาพที่ **3‑25** การใช้งานโปรแกรมการคืนหนังสือ

3.6.4 เมื่อกรอกหมายเลข 2 แล้ว จะปรากฎหัวข้อการกรอกข้อมูลของการคืนหนังสือที่ต้องการจะคืน เช่น รหัสการยืมหนังสือ

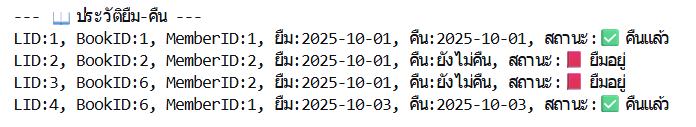


รูปภาพที่ **3‑26 การกรอกข้อมูลของการคืนหนังสือ**

3.6.5 เมื่อผู้ใช้งานอยู่ที่เมนู จัดการการยืม - คืน กรอกหมายเลข 3 เพื่อดูประวัติการยืม - คืนทั้งหมด



รูปภาพที่ **3‑27** **การดูประวัติการยืม - คืนทั้งหมด**

3.6.6 เมื่อกรอกหมายเลข 3 แล้ว จะปรากฏข้อมูลของการยืม - คืนหนังสือทั้งหมด

รูปภาพที่ **3‑28 ข้อมูลของการยืม - คืนหนังสือทั้งหมด**

# อธิบายการทำงานของ **Code**

## ภาพรวมของระบบ

ระบบจัดการห้องสมุด PyLibMan ใช้การจัดเก็บข้อมูลแบบ Binary File ด้วยโมดูล struct ของ Python ประกอบด้วย 5 โมดูลหลัก:

* + - 1. books.py - จัดการหนังสือ
      2. members.py - จัดการสมาชิก
      3. lendings.py - จัดการยืม-คืน
      4. report.py - สร้างรายงาน
      5. main.py – เมนูหลัก

## ฟังก์ชันหลักในแต่ละโมดูล

### 4.2.1 โมดูล books.py

### 4.2.1.1 ฟังก์ชัน add\_book()

วัตถุประสงค์: เพิ่มหนังสือใหม่เข้าระบบ

ขั้นตอนการทำงาน:

1. เรียกใช้ get\_last\_id() เพื่อหา ID ล่าสุดในไฟล์ แล้วเพิ่ม 1 เป็น ID ใหม่

get\_last\_id(...) เป็นฟังก์ชันที่ไปอ่านไฟล์ฐานข้อมูล (BOOKS\_FILE) แล้วคำนวณว่า ID ของหนังสือเล่มล่าสุดคืออะไร จากนั้นนำ last\_id + 1 เพื่อสร้าง book\_id ใหม่ให้กับหนังสือที่จะเพิ่ม

1. รับข้อมูลจากผู้ใช้ (ISBN, ชื่อหนังสือ, ผู้แต่ง, จำนวน)

isbn = รหัสหนังสือ (อาจเป็นเลข ISBN)

title = ชื่อหนังสือ

author = ผู้แต่ง

quantity = จำนวนเล่ม (ต้องแปลงเป็น int)

1. ใช้ struct.pack() แปลงข้อมูลเป็น binary ตามรูปแบบ BOOK\_FORMAT

ใช้ struct.pack แปลงข้อมูลจาก Python → Binary ตามรูปแบบที่กำหนดในBOOK\_FORMAT

STATUS\_ACTIVE = flag บอกว่ายังใช้งานได้ (ไม่ถูกลบ)

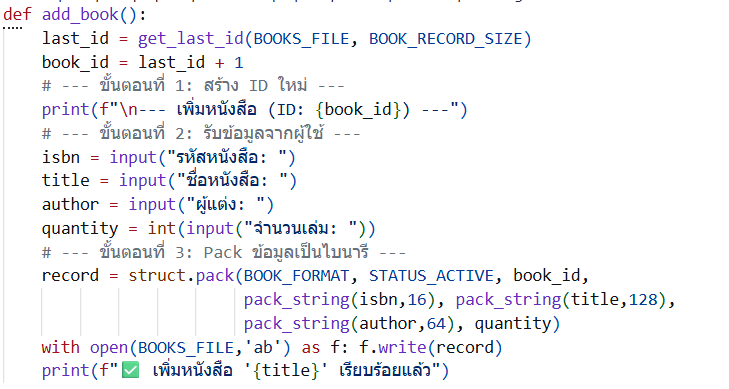
book\_id = รหัสหนังสือใหม่

pack\_string(..., size) = ฟังก์ชันเสริมที่ใช้จัดการ string → fixed size (เช่น ISBN 16 byte, ชื่อ 128 byte, ผู้แต่ง 64 byte)

quantity = จำนวนเล่ม

1. เขียนข้อมูลต่อท้ายไฟล์ด้วย mode 'ab' (append binary)

* เปิดไฟล์ BOOKS\_FILE ด้วยโหมด ab (append binary)
* เขียน record ที่เพิ่งสร้างลงไปต่อท้ายไฟล์
* ใช้ with เพื่อให้ปิดไฟล์ให้อัตโนมัติเมื่อเสร็จ



รูปภาพที่ **4‑1** **ฟังก์ชั่น** add\_book

* + - 1. ฟังก์ชัน update\_book()

ฟังก์ชันถูกแบ่งการทำงานออกเป็น 3 ขั้นตอนหลัก ๆ ดังนี้

1. การสร้าง ID ใหม่ (last\_id = get\_last\_id(...))

* last\_id = get\_last\_id(BOOKS\_FILE, BOOK\_RECORD\_SIZE): ฟังก์ชันนี้จะเรียกใช้ฟังก์ชันภายนอกชื่อ get\_last\_id เพื่อดึง ID ล่าสุด ของหนังสือที่มีอยู่ในไฟล์ฐานข้อมูล

BOOKS\_FILE: คือตัวแปรที่เก็บชื่อหรือพาธของไฟล์ฐานข้อมูลหนังสือ

* BOOK\_RECORD\_SIZE: คือขนาดของเรคคอร์ด (record) หนังสือแต่ละรายการในหน่วยไบต์ ซึ่งใช้ในการคำนวณตำแหน่งของเรคคอร์ดสุดท้าย

book\_id = last\_id + 1: นำ ID ล่าสุดที่ได้มาบวกด้วย 1 เพื่อกำหนดให้เป็น ID ใหม่ สำหรับหนังสือเล่มที่จะเพิ่ม

2. การรับข้อมูลจากผู้ใช้ (Input)

ส่วนนี้เป็นการรับรายละเอียดของหนังสือเล่มใหม่จากผู้ใช้ผ่านทางคอนโซล/เทอร์มินัล โดยใช้ฟังก์ชัน input():

isbn = input("รหัสหนังสือ: "): รับ รหัสหนังสือ (ISBN)

title = input("ชื่อหนังสือ: "): รับ ชื่อหนังสือ

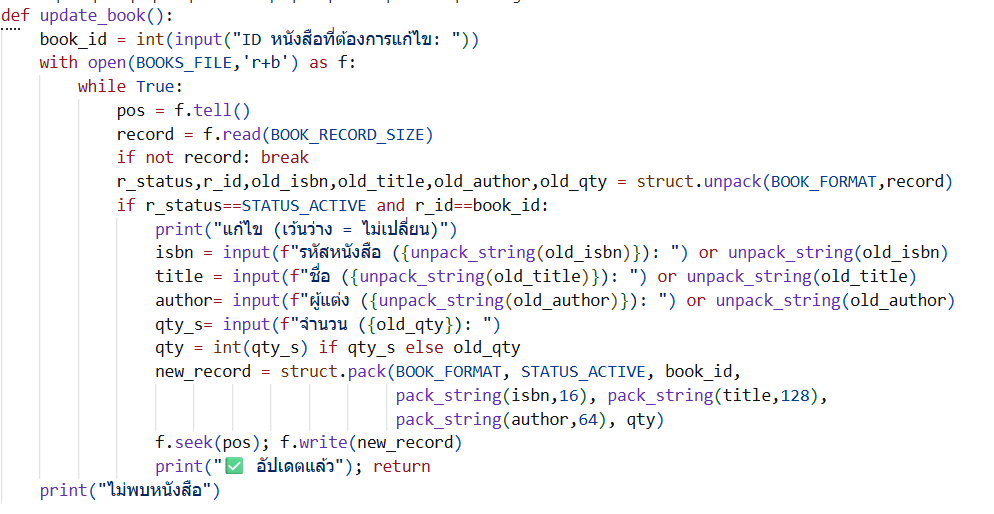
author = input("ผู้แต่ง: "): รับ ชื่อผู้แต่ง

quantity = int(input("จำนวนเล่ม: ")): รับ จำนวนเล่ม และใช้ฟังก์ชัน int() ครอบเพื่อ แปลงค่าที่รับเข้ามาให้เป็นจำนวนเต็ม (Integer) เนื่องจากข้อมูลจำนวนเล่มจะต้องเป็นตัวเลข

3. การ Pack ข้อมูลเป็นไบนารี (Data Structuring and Packing)

ขั้นตอนนี้คือการเตรียมข้อมูลให้อยู่ในรูปแบบที่สามารถบันทึกในไฟล์ไบนารีได้อย่างมีโครงสร้างและมีขนาดคงที่ (Fixed-width record structure) โดยใช้โมดูล struct ของ Python:

* record = struct.pack(BOOK\_FORMAT, ...): ฟังก์ชัน struct.pack() จะทำการแปลง (pack) ข้อมูลที่รับเข้ามาให้อยู่ในรูปของ ไบนารีสติง (bytes object) ตามรูปแบบ (format string) ที่กำหนดไว้ในตัวแปร BOOK\_FORMAT
* ข้อมูลที่ถูก Pack เรียงตามลำดับคือ:
  1. STATUS\_ACTIVE: สถานะของเรคคอร์ด (สันนิษฐานว่าเป็นค่าคงที่ที่บ่งบอกว่าหนังสือนี้เป็นข้อมูลที่ใช้งานอยู่)
  2. book\_id: ID ของหนังสือที่สร้างขึ้นใหม่
  3. pack\_string(isbn, 16): แปลงรหัส ISBN ให้อยู่ในรูปแบบไบนารีที่มีความยาว 16 ไบต์ โดยเรียกใช้ฟังก์ชันภายนอก pack\_string (การทำ padding หรือ truncation เพื่อให้ความยาวคงที่)
  4. pack\_string(title, 128): แปลงชื่อหนังสือให้อยู่ในรูปแบบไบนารีที่มีความยาว 128 ไบต์
  5. pack\_string(author, 64): แปลงชื่อผู้แต่งให้อยู่ในรูปแบบไบนารีที่มีความยาว 64 ไบต์
  6. quantity: จำนวนเล่ม (ตัวเลข Integer)

ผลลัพธ์ที่ได้คือตัวแปร record ซึ่งเป็นข้อมูลหนังสือในรูปแบบไบนารีที่พร้อมสำหรับการบันทึก

รูปภาพที่ **4‑2** **ฟังก์ชัน** update\_book

4.2.1.3 ฟังก์ชัน delete\_book()

วัตถุประสงค์: ลบหนังสือ (Soft Delete)

ขั้นตอนการทำงาน:

1. รับ book\_id จากผู้ใช้

รับค่า ID ของหนังสือที่ต้องการลบจากผู้ใช้

แปลงให้เป็น int ก่อนใช้งาน

1. เปิดไฟล์ฐานข้อมูล

เปิดไฟล์ฐานข้อมูล (BOOKS\_FILE) ในโหมด r+b

r = read

+ = read/write

b = binary

เราต้องใช้โหมดนี้เพราะจะ อ่านข้อมูลและเขียนข้อมูลทับบางส่วน ในไฟล์เดียวกัน

1. วนอ่านทีละ record

วนลูปอ่านข้อมูลทีละ “record” (ขนาดคงที่ BOOK\_RECORD\_SIZE)

pos = f.tell() → จำตำแหน่งก่อนอ่านไว้ เพราะถ้าพบ record ที่ต้องลบ เราจะต้อง “seek กลับมาเขียนทับ” ที่ตำแหน่งเดิม

ถ้า record ว่าง (EOF) แปลว่าอ่านจนจบไฟล์ → หยุดลูป

1. แปลงข้อมูลจาก Binary → ตัวแปร

struct.unpack จะถอดข้อมูล binary ออกมาเป็นค่าปกติ

r\_status → สถานะของหนังสือ (STATUS\_ACTIVE = ใช้งาน STATUS\_DELETED = ลบแล้ว)

r\_id → รหัสหนังสือ

isbn, title, author → ข้อมูล string (แต่ยังอยู่ในรูปแบบ bytes)

qty → จำนวนเล่ม

1. ตรวจสอบว่าใช่หนังสือที่ต้องการลบไหม

เช็กว่า record นี้ยัง “active” อยู่ และ r\_id ตรงกับ ID ที่ผู้ใช้กรอก

1. ยืนยันก่อนลบ

unpack\_string(title) → แปลงข้อมูล bytes เป็น string ปกติ

ถามผู้ใช้ก่อนว่าต้องการลบจริงหรือไม่

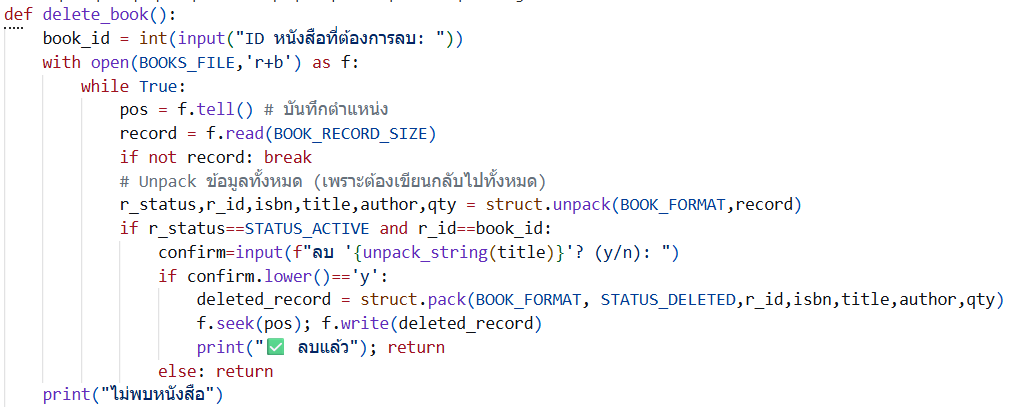
1. ลบ (เปลี่ยนสถานะ)

ถ้าผู้ใช้พิมพ์ y → ทำการ “ลบ” โดย:

สร้าง record ใหม่ โดยเปลี่ยน STATUS\_ACTIVE เป็น STATUS\_DELETED

f.seek(pos) → ย้าย pointer กลับไปตำแหน่งเดิมของ record

เขียนข้อมูลใหม่ทับ record เดิม

จากนั้นแสดงข้อความ “ลบแล้ว” แล้ว return ออกจากฟังก์ชัน

รูปภาพที่ **4‑3 ฟังก์ชัน** delete\_book

### 4.2.2 โมดูล members.py

4.2.2.1 ฟังก์ชัน add\_member()

วัตถุประสงค์: เพิ่มสมาชิกใหม่เข้าระบบ

ขั้นตอนการทำงาน:

1. หาค่า ID ล่าสุด แล้วสร้าง ID ใหม่

get\_last\_id() จะอ่านไฟล์สมาชิก (MEMBERS\_FILE) แล้วคำนวณว่า ID สมาชิกล่าสุดคืออะไร

จากนั้นเพิ่ม 1 เพื่อสร้าง member\_id ใหม่ให้กับสมาชิกที่จะเพิ่ม

1. แสดงข้อความบนหน้าจอให้ผู้ใช้เห็นว่าเรากำลังเพิ่มสมาชิกใหม่ พร้อม ID ที่ถูกกำหนดให้
2. รับข้อมูลสมาชิกจากผู้ใช้

name = ชื่อ-นามสกุลของสมาชิก

phone = หมายเลขโทรศัพท์

1. Pack ข้อมูลเป็น Binary Record

ใช้ struct.pack() เพื่อแปลงข้อมูลจาก Python เป็น Binary ตามรูปแบบ (MEMBER\_FORMAT) ที่กำหนดไว้

ข้อมูลที่ถูก pack จะประกอบด้วย:

STATUS\_ACTIVE → สถานะสมาชิก (active = ยังใช้งานได้)

member\_id → รหัสสมาชิก

pack\_string(name, 64) → แปลงชื่อเป็น bytes ความยาว 64 byte

pack\_string(phone, 16) → แปลงเบอร์โทรเป็น bytes ความยาว 16 byt

1. บันทึกข้อมูลลงไฟล์

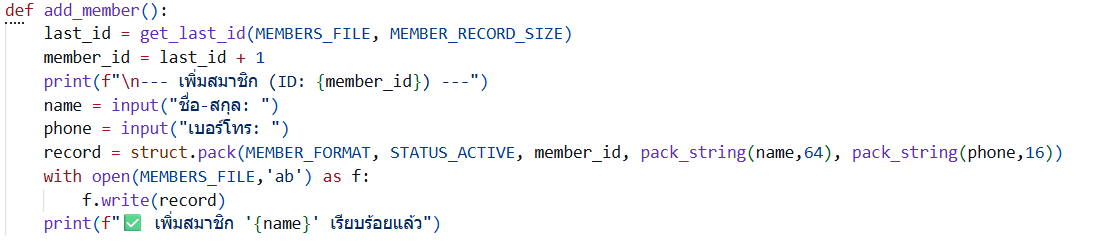
เปิดไฟล์สมาชิก (MEMBERS\_FILE) ในโหมด ab:

a = append (เพิ่มข้อมูลต่อท้าย)

b = binary (อ่าน/เขียนแบบไบนารี)

เขียนข้อมูล record ลงไปที่ท้ายไฟล์

1. แจ้งผลให้ผู้ใช้ทราบว่าสมาชิกถูกเพิ่มเรียบร้อยแล้ว

รูปภาพที่ **4‑4** **ฟังก์ชัน** add\_member

4.2.2.2 ฟังก์ชัน view\_all\_members()

วัตถุประสงค์: แสดงรายชื่อสมาชิกทั้งหมดที่มีสถานะ Active

ขั้นตอนการทำงาน:

1. ตรวจสอบว่าไฟล์ members.dat มีอยู่และไม่ว่างเปล่า

os.path.exists(MEMBERS\_FILE) → ตรวจสอบว่าไฟล์สมาชิกมีอยู่จริงหรือไม่

os.path.getsize(MEMBERS\_FILE)==0 → ตรวจสอบว่าไฟล์ว่างเปล่าหรือไม่ (ขนาด = 0 byte

1. เปิดไฟล์แบบ read binary ('rb')

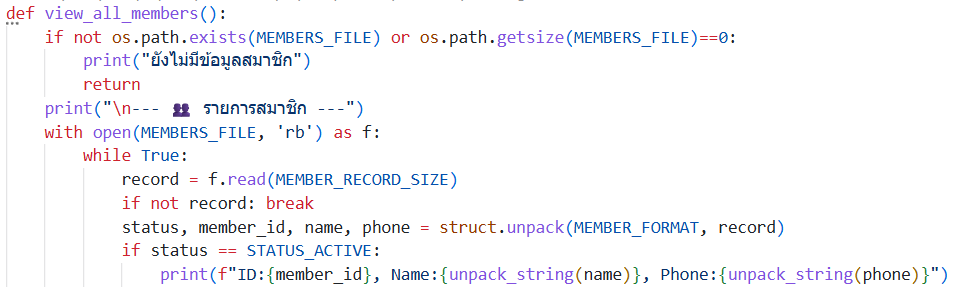
เปิดไฟล์ในโหมด 'rb' (read binary) เพราะข้อมูลถูกจัดเก็บในรูปแบบ **ไบนารี**

ใช้ with เพื่อให้ไฟล์ปิดเองอัตโนมัติหลังใช้งาน

1. วนลูปอ่านทีละ MEMBER\_RECORD\_SIZE bytes จนจบไฟล์

ใช้ while True เพื่อวนอ่านข้อมูลทีละ record โดยขนาดของแต่ละ record คือ MEMBER\_RECORD\_SIZE (ค่าคงที่ที่ระบุจำนวน byte ของ 1 record)

ถ้า record ว่าง (อ่านไม่เจอข้อมูล) → หมายถึงถึงท้ายไฟล์แล้ว → break ออกจากลูป

1. ใช้ struct.unpack() แยกข้อมูลออกมาเป็น status, member\_id, name, phone
2. ตรวจสอบว่า status == STATUS\_ACTIVE (เฉพาะสมาชิกที่ยังใช้งาน)
3. ใช้ unpack\_string() แปลง bytes กลับเป็น string
4. แสดงข้อมูลในรูปแบบ: ID:{member\_id}, Name:{name}, Phone:{phone}

รูปภาพที่ ‑ **ฟังก์ชัน** view\_all\_members

4.2.2.3 ฟังก์ชัน update\_member()

วัตถุประสงค์: แก้ใขข้อมูลสมาชิกที่บันทึกไว้แล้ว

ขั้นตอนการทำงาน:

1. รับ member\_id ที่ต้องการแก้ไขจากผู้ใช้เปิดไฟล์แบบ read binary ('rb')
2. ตรวจสอบว่ามีไฟล์ members.dat หรือไม่
3. เปิดไฟล์แบบ read+write binary ('r+b')ตรวจสอบว่า status == STATUS\_ACTIVE (เฉพาะสมาชิกที่ยังใช้งาน)
4. วนลูปค้นหา record ที่ตรงกับ member\_id ใช้ f.tell() บันทึกตำแหน่งปัจจุบันก่อนอ่าน อ่าน record และ unpack ข้อมูล ตรวจสอบว่า status == ACTIVE และ ID ตรงกัน
5. เมื่อเจอ record ที่ต้องการ แสดงข้อมูลเดิมให้เห็น รับข้อมูลใหม่ (ถ้าผู้ใช้กด Enter = ไม่เปลี่ยน ใช้ค่าเดิม)
6. Pack ข้อมูลใหม่เป็น binary
7. ใช้ f.seek(pos) กลับไปตำแหน่งเดิม
8. ใช้ f.write() เขียนทับ record เดิม
9. แสดงข้อความ "อัปเดตสมาชิกเรียบร้อย"

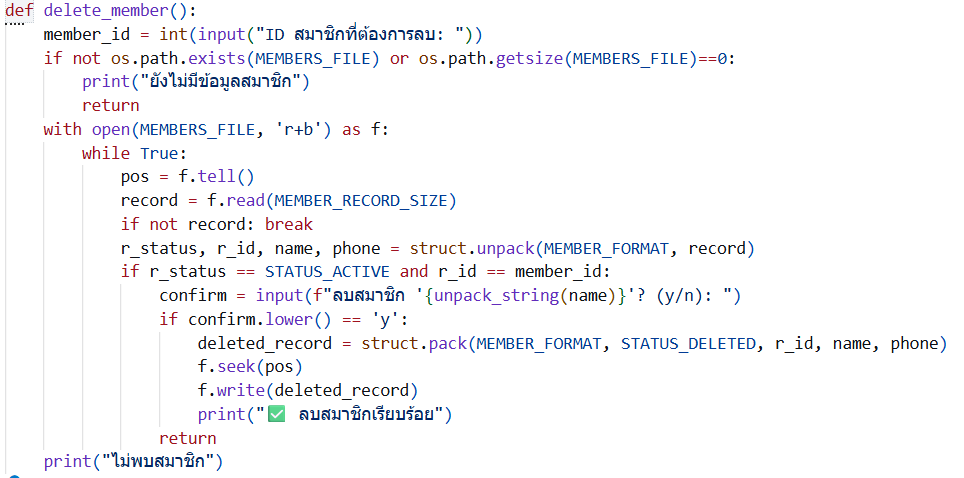
รูปภาพที่ **4‑6 ฟังก์ชัน** view\_all\_members

4.2.2.4 ฟังก์ชัน delete\_member()

วัตถุประสงค์: ลบสมาชิกออกจากระบบ (Soft Delete)

ขั้นตอนการทำงาน:

1. รับ member\_id ที่ต้องการลบ
2. ตรวจสอบว่ามีไฟล์หรือไม่
3. เปิดไฟล์แบบ 'r+b'
4. วนลูปค้นหา record ที่ตรงกับ ID: บันทึกตำแหน่งด้วย f.tell() อ่านและ unpack ข้อมูล
5. เมื่อเจอ record ที่ต้องการ แสดงชื่อสมาชิกและขอยืนยัน
6. ถ้าผู้ใช้ยืนยัน (กด 'y') เปลี่ยนสถานะจาก STATUS\_ACTIVE เป็น STATUS\_DELETED Pack ข้อมูลใหม่ที่มีสถานะเปลี่ยน
7. กลับไปตำแหน่งเดิมและเขียนทับ
8. แสดงข้อความ "✅ ลบสมาชิกเรียบร้อย"

 4.2.3 โมดูล lendings.py

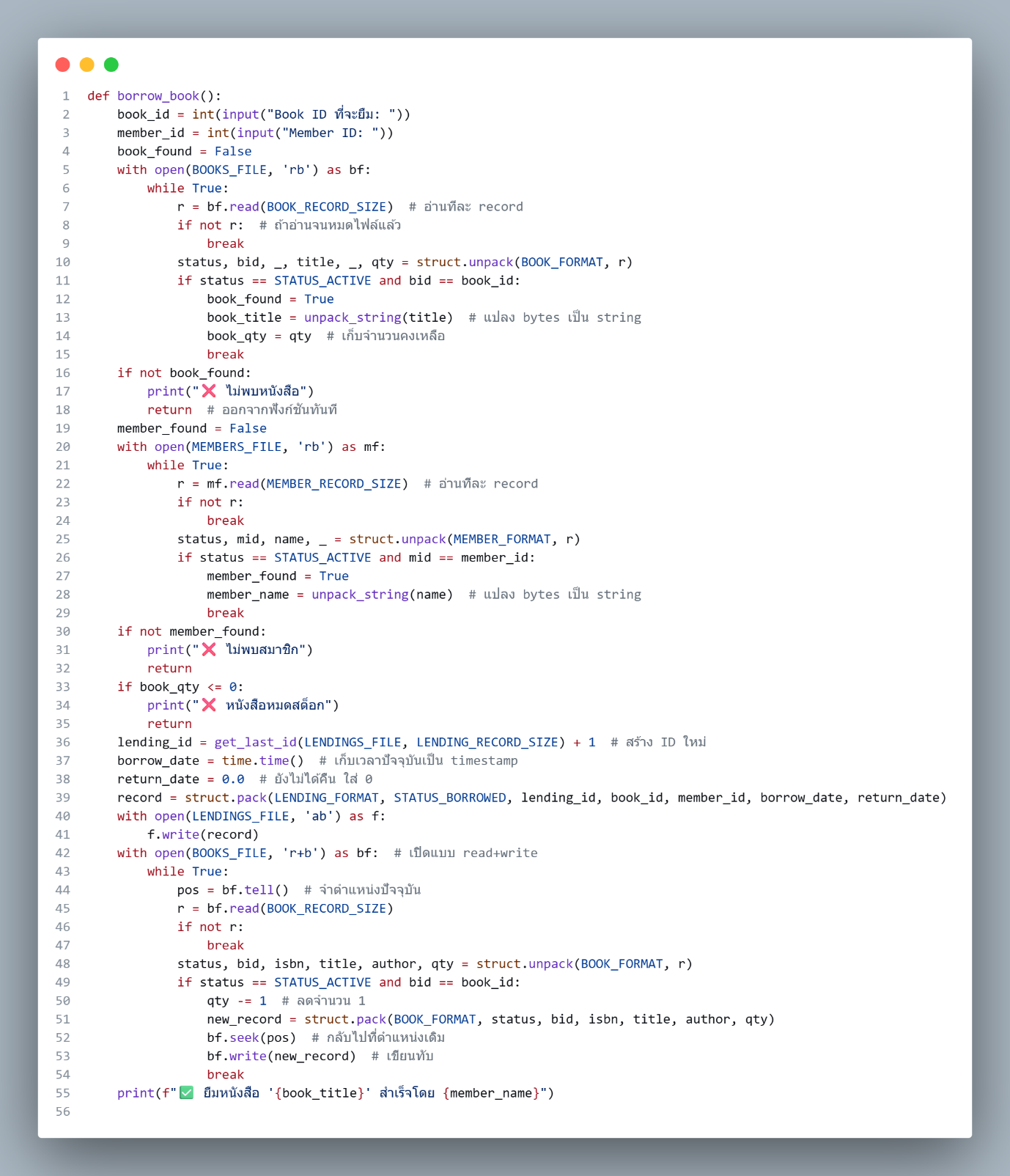
รูปภาพที่ **4‑7 ฟังก์ชัน** delete\_member

4.2.3.1 ฟังก์ชัน borrow\_book()

วัตถุประสงค์: บันทึกการยืมหนังสือ

ขั้นตอนการทำงาน:

1. ตรวจสอบหนังสือ: อ่านไฟล์ books.dat หา book\_id ที่ต้องการยืม
2. ตรวจสอบสมาชิก: อ่านไฟล์ members.dat หา member\_id
3. ตรวจสอบสต็อก: ถ้า quantity <= 0 แสดงข้อความ "หนังสือหมดสต็อก"
4. บันทึกการยืม สร้าง lending\_id ใหม่ ใช้ time.time() บันทึกเวลาปัจจุบันเป็น timestamp บันทึกลงไฟล์ lendings.dat
5. ลดสต็อก: อัปเดตจำนวนหนังสือใน books.dat ลดลง 1

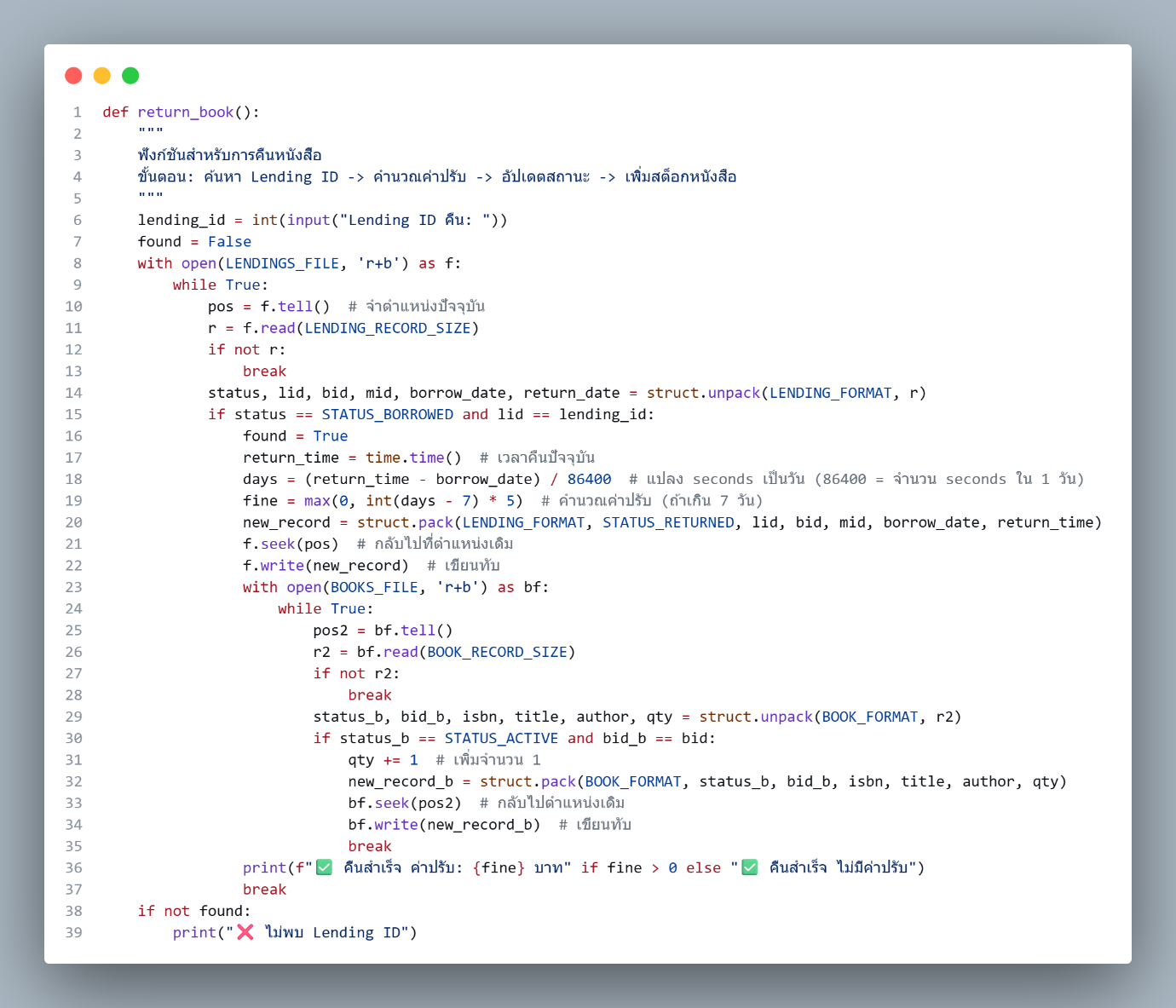


รูปภาพที่ **4‑8 ฟังก์ชัน** borrow\_book

4.2.3.2 ฟังก์ชัน return\_book()

วัตถุประสงค์: บันทึกการคืนหนังสือและคำนวณค่าปรับ

ขั้นตอนการทำงาน:

1. รับ Lending ID
2. ค้นหา record ที่มีสถานะ STATUS\_BORROWED
3. คำนวณค่าปรับ หาจำนวนวันที่ยืม: (return\_time - borrow\_date) / 86400 ถ้ายืมเกิน 7 วัน: ปรับวันละ 5 บาท
4. อัปเดตสถานะ: เปลี่ยนเป็น STATUS\_RETURNED และบันทึก return\_time
5. เพิ่มสต็อก: อัปเดตจำนวนหนังสือใน books.dat เพิ่ม 1

รูปภาพที่ **4‑9 ฟังก์ชัน** return\_book

4.2.4 โมดูล report.py

4.2.4.1 ฟังก์ชัน generate\_report()

วัตถุประสงค์: สร้างรายงานสรุปการยืม-คืนแบบครบถ้วน

ขั้นตอนการทำงาน:

1. เตรียมหัวกระดาษและรูปแบบของรายงาน

สร้าง lines (list ของบรรทัด) เพื่อเก็บข้อความที่จะเขียนลงไฟล์รายงานทีหลัง

"=" \* 120 สร้างเส้นแบ่งความยาว 120 ตัวอักษร (สำหรับความสวยงามของรายงาน)

.center(120) ทำให้หัวข้อย่อยอยู่กึ่งกลางความกว้าง 120 ตัวอักษร

datetime.datetime.now().strftime(...) ใส่เวลาที่สร้างรายงาน (รูปแบบ YYYY-MM-DD HH:MM:SS) — ใช้เวลาจากระบบ (system local time)

header กำหนดคอลัมน์ของตาราง (ชื่อ, โทรศัพท์, หนังสือ, สถานะ) โดยใช้ f-string กับการจัดความกว้าง (:<30 เป็นต้น)

1. สร้าง members\_dict เพื่อแม็ป member\_id → ข้อมูล {name, phone}

เช็คว่ามีไฟล์และไม่ว่างก่อนอ่าน

วนอ่านทีละ MEMBER\_RECORD\_SIZE ไบต์ (fixed-size records) และ struct.unpack(MEMBER\_FORMAT, r) แปลงไบนารีเป็นค่าปกติ

ถ้า status == MEMBER\_ACTIVE (สมาชิกยัง active) → เก็บข้อมูลลง members\_dict โดยใช้ member\_id เป็น key

unpack\_string() คาดว่าเป็นฟังก์ชันที่แปลงจาก bytes → string และตัดpadding/NULL ออก

1. อ่านข้อมูลหนังสือและเก็บลง books\_dict

ทำงานคล้าย members\_dict แต่เก็บ book\_id → title

ใน struct.unpack โค้ดใช้ \_ เพื่อข้ามฟิลด์ที่ไม่ต้องการ (เช่น ISBN หรือ author) แล้วเลือกเอา title เท่านั้น

เก็บเฉพาะหนังสือที่ status == STATUS\_ACTIVE

1. เตรียมตัวนับสถิติและโครงสร้างเก็บการยืมของสมาชิก

สร้างตัวนับ: จำนวนการยืมทั้งหมด, จำนวนที่ยืมยังไม่คืน, จำนวนที่คืนแล้ว

สร้าง member\_lendings เพื่อเก็บข้อมูลการยืมแยกตามสมาชิก — โค้ดนี้ใช้ member\_name เป็น key (จะมีผลตามข้อสังเกตด้านล่าง)

1. อ่านไฟล์การยืม (LENDINGS\_FILE) แล้วรวมข้อมูล

อ่านทีละ record ของการยืมและ unpack เป็น fields (รูปแบบขึ้นกับ LENDING\_FORMAT)

status, \_, bid, mid, \_, \_ — สมมติว่า bid = book\_id, mid = member\_id

เพิ่ม counters (total, borrowed, returned) ตาม status ของแต่ละ record

หา member\_info จาก members\_dict โดยใช้ mid (member\_id); หากไม่เจอใช้ default 'ไม่ระบุ'

เก็บชื่อหนังสือจาก books\_dict ตาม bid (book\_id) — ถ้าไม่พบให้ใส่ข้อความ "ไม่พบชื่อหนังสือ"

เพิ่มชื่อหนังสือและสถานะลงใน list ของสมาชิกคนนั้น (books และ statuses)

1. สร้างบรรทัดของรายงานจาก member\_lendings

สำหรับแต่ละสมาชิก: รวมชื่อหนังสือที่ยืมเป็นสตริงเดียวด้วยคอมมา

กำหนด status\_text แบบเรียบง่าย: ถ้ามีอย่างน้อยหนึ่งสถานะ STATUS\_BORROWED ให้แสดงว่า “ยืมอยู่” มิฉะนั้นแสดงว่า “คืนแล้ว”

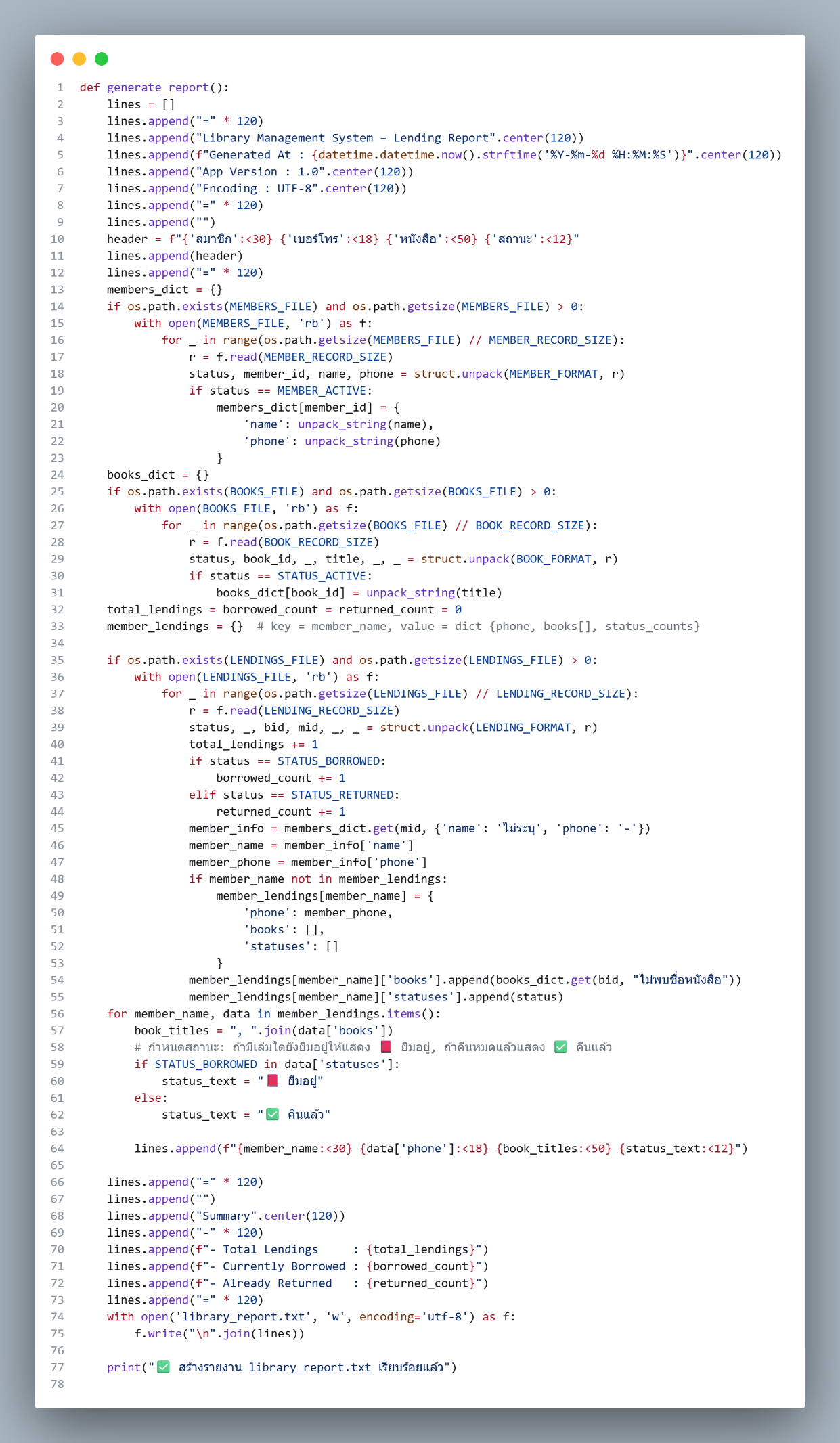
ใส่บรรทัดเข้ากับ lines โดยจัดคอลัมน์ตามความกว้างที่กำหนด (:<30, :<18, :<50, :<12)

1. สรุปท้ายรายงานและเขียนไฟล์

เติมส่วน Summary (Total, Currently Borrowed, Already Returned) ใน lines

เปิดไฟล์ 'library\_report.txt' ด้วย encoding='utf-8' แล้วเขียน "\n".join(lines) — คือเอา list ของบรรทัดมาต่อกันเป็นไฟล์ข้อความ

พิมพ์ข้อความยืนยันเมื่อเขียนเสร็จ



รูปภาพที่ **4‑10 ฟังก์ชัน** generate\_report

4.2.5 โมดูล main.py

4.2.5.1 ฟังก์ชัน main()

วัตถุประสงค์: ควบคุมเมนูหลักของโปรแกรม

ขั้นตอนการทำงาน:

1. แสดงเมนูหลัก 5 ตัวเลือก
2. รับคำสั่งจากผู้ใช้
3. เรียกใช้ฟังก์ชันเมนูย่อย

'1' → books\_menu()

'2' → members\_menu()

'3' → lendings\_menu()

'4' → generate\_report()

'0' → ออกจากโปรแกรม

1. ใช้ os.system('cls') ล้างหน้าจอเพื่อความสะอาด



รูปภาพที่ **4‑11 ฟังก์ชัน** main

# สรุปผลการดำเนินงานและข้อเสนอแนะ

## สรุปผลการดำเนินงาน

ระบบจัดการห้องสมุด PyLibMan ได้รับการพัฒนาสำเร็จครบถ้วนตามวัตถุประสงค์ที่กำหนดไว้ โดยมีผลการดำเนินงานดังนี้

5.1.1 ฟังก์ชันการทำงานหลัก

ระบบสามารถทำงานได้ครบทั้ง 4 ส่วนหลัก ได้แก่:

* การจัดการหนังสือ - สามารถเพิ่ม แสดง แก้ไข และลบข้อมูลหนังสือได้อย่างสมบูรณ์
* การจัดการสมาชิก - จัดเก็บข้อมูลสมาชิกพร้อมรองรับการแก้ไขและลบแบบ Soft Delete
* การจัดการยืม-คืน - มีระบบคำนวณค่าปรับอัตโนมัติและจัดการสต็อกหนังสืออย่างถูกต้อง
* การสร้างรายงาน - สร้างรายงานสรุปข้อมูลการยืม-คืนได้อย่างครบถ้วนและอ่านง่าย

5.1.2 เทคโนโลยีที่ใช้

* ใช้โมดูล struct ในการจัดการ Binary File ทำให้ข้อมูลมีขนาดเล็กและเข้าถึงได้เร็ว
* ใช้ Timestamp ในการบันทึกวันเวลา ทำให้คำนวณค่าปรับได้แม่นยำ
* ออกแบบโครงสร้างโค้ดแบบแยกโมดูล (Modular Design) ทำให้ง่ายต่อการดูแลและพัฒนาเพิ่มเติม

5.1.3 จุดเด่นของระบบ

* Auto ID Generation - ระบบสร้าง ID อัตโนมัติจาก record สุดท้าย ไม่ต้องกังวลเรื่องการซ้ำกัน
* Soft Delete - ข้อมูลที่ลบยังคงอยู่ในไฟล์ สามารถกู้คืนได้ในภายหลัง
* การจัดการสต็อกอัตโนมัติ - ระบบปรับจำนวนหนังสือเมื่อมีการยืม-คืนโดยอัตโนมัติ
* รายงานแบบรวมกลุ่ม - จัดกลุ่มหนังสือหลายเล่มของสมาชิกเดียวกันไว้ด้วยกัน อ่านเข้าใจง่าย

## ปัญหาและอุปสรรคในการดำเนินงาน

### 5.2.1 ปัญหาด้านการเข้ารหัส (Encoding)

**ปัญหา:** การจัดการข้อความภาษาไทยใน Binary File มีปัญหาเรื่องการเข้ารหัส UTF-8 ที่ใช้ 3 bytes ต่อตัวอักษร

**แนวทางแก้ไข:** กำหนดขนาด field ให้ใหญ่พอ (เช่น ชื่อหนังสือ 128 bytes) เพื่อรองรับอักษรไทย และใช้ฟังก์ชัน pack\_string() ที่ตัดข้อความให้พอดีกับขนาดที่กำหนด

### 5.2.2 ปัญหาการจัดการตำแหน่งในไฟล์

**ปัญหา:** การ seek และ write ในตำแหน่งที่ไม่ถูกต้องอาจทำให้ข้อมูลเสียหายหรือเขียนทับผิด record

**แนวทางแก้ไข:** ใช้ f.tell() บันทึกตำแหน่งก่อนอ่านทุกครั้ง และใช้ f.seek(pos) กลับไปตำแหน่งเดิมก่อนเขียนข้อมูล

### 5.2.3 ปัญหาการคำนวณขนาด Record

**ปัญหา:** การคำนวณขนาด record ผิดพลาดอาจทำให้อ่านข้อมูลไม่ครบหรือเกิน

**แนวทางแก้ไข:** ใช้ struct.calcsize() คำนวณขนาดอัตโนมัติจาก format string แทนการคำนวณเอง

### 5.2.4 ปัญหาการแสดงผลในรายงาน

**ปัญหา:** รายงานที่มีหนังสือหลายเล่มต่อคนทำให้ตารางยาวและอ่านยาก

**แนวทางแก้ไข:** ใช้วิธีรวมชื่อหนังสือหลายเล่มเป็น string เดียวด้วย ", ".join() และกำหนดสถานะรวมจากทุกเล่มที่ยืม

## ข้อเสนอแนะ

5.3.1 เพิ่มระบบค้นหา ค้นหาหนังสือตามชื่อหรือผู้แต่ง ค้นหาสมาชิกตามชื่อหรือเบอร์โทร

5.3.2 เพิ่มระบบจัดการผู้ใช้ แยกระดับผู้ใช้เป็น Admin และ Staff ระบบ Login ด้วย Username และ Password

## สิ่งที่ผู้จัดทำได้รับในการพัฒนาโครงงาน

5.4.1 ความรู้และทักษะที่ได้รับ

5.4.2 เข้าใจการทำงานของ struct.pack() และ struct.unpack()

5.4.3 เข้าใจการทำงานของ struct.pack() และ struct.unpack()