**การจัดการคลังสินค้า**

**Warehouse management**

**นายวิทวัส แก้ววิเศษ รหัส 6706022511014**

**นายอนันต์ยศ สายวงษ์ รหัส 6706022510379**

**นายธีรยุทธ ปาแก้ว รหัส 6706022510395**

**นางสาวพิยดา ช้างน้อย รหัส 6706022510361**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต**

**สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศ**

**คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

**ปีการศึกษา 2567**

**ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ**

# สารบัญ

[สารบัญ ข](#_Toc179831742)

[สารบัญภาพ ค](#_Toc179831743)

[บทที่ 1 บทนำ 1](#_Toc179831744)

[1.1 วัตถุประสงค์ของโครงงาน 1](#_Toc179831745)

[1.2. ขอบเขตของโครงงาน 1](#_Toc179831746)

[1.3. ประโยชน์ที่ได้รับ 1](#_Toc179831747)

[1.4. เครื่องมือที่คาดว่าจะต้องใช้ 1](#_Toc179831748)

[บทที่ 2 ระบบการจัดการคลังสินค้า 2](#_Toc179831749)

[2.1. ฟิลด์ในระบบจัดการคลังสินค้า 2](#_Toc179831750)

[2.2 สรุปฟิลด์ในระบบจัดการคลังสินค้า 2](#_Toc179831751)

[บทที่3 การใช้โปรแกรม การจัดการคลังสินค้า 3](#_Toc179831752)

[3.1 สำหรับผู้ใช้งานโปรแกรม 3](#_Toc179831753)

[3.2 การใช้โปรแกรม การแสดงข้อมูลทั้งหมด 4](#_Toc179831754)

[3.3 การใช้โปรแกรม การค้นหาข้อมูล 5](#_Toc179831755)

[3.4 การใช้โปรแกรม การอัปเดตข้อมูล 6](#_Toc179831756)

[3.5. การใช้โปรแกรม การลบข้อมูล 7](#_Toc179831757)

[3.6 การใช้โปรแกรม สร้างรายงาน 8](#_Toc179831758)

[3.7 การใช้โปรแกรม ออกจากโปรแกรม 9](#_Toc179831759)

[บทที่4 อธิบายการทำงานของ code 10](#_Toc179831760)

[การกำหนดรูปแบบข้อมูล 10](#_Toc179831761)

[4.1 ฟังก์ชันสำหรับเพิ่มข้อมูล 11](#_Toc179831762)

[4.2 ฟังก์ชันสำหรับแสดงข้อมูลทั้งหมด 11](#_Toc179831763)

[4.3 ฟังก์ชันสำหรับค้นหาข้อมูลตามรหัสสินค้า 12](#_Toc179831764)

[4.4 ฟังก์ชันสำหรับอัปเดตข้อมูลสินค้า 13](#_Toc179831765)

[4.5 ฟังก์ชันสำหรับลบข้อมูลสินค้า 14](#_Toc179831766)

[4.6 ฟังก์ชันสำหรับเขียนรายงาน 15](#_Toc179831767)

[4.7 จัดเก็บข้อมูลสินค้า 17](#_Toc179831768)

[4.8 เมนูสำหรับเลือกฟังก์ชัน 18](#_Toc179831769)

[4.9 เพิ่มข้อมูลสินค้า 19](#_Toc179831770)

# สารบัญภาพ

[รูปที่ 3.1.1 การเพิ่มข้อมูล 3](#_Toc179829826)

[รูปที่ 3.1.2 การเลือกฟังก์ชัน 4](#_Toc179829827)

[รูปที่ 3.2.1 การแสดงข้อมูลทั้งหมด 4](#_Toc179829828)

[รูปที่ 3.2.2 แสดงข้อมูลสินค้าทั้งหมด 4](#_Toc179829829)

[รูปที่ 3.3.1 การค้นหาข้อมูล 5](#_Toc179829830)

[รูปที่ 3.3.2 การค้นหาข้อมูล 5](#_Toc179829831)

[รูปที่3.4.1 การอัปเดตข้อมูล 6](#_Toc179829832)

[รูปที่3.4.2 การอัปเดตข้อมูล 6](#_Toc179829833)

[รูปที่ 3.5.1 การลบข้อมูล 7](#_Toc179829834)

[รูปที่ 3.5.2 การลบข้อมูลรายการสินค้า 7](#_Toc179829835)

[รูปที่ 3.6.1 การสร้างรายงาน 8](#_Toc179829836)

[รูปที่ 3.6.2 รายงานแยกประเภทสินค้า 8](#_Toc179829837)

[รูปที่ 3.7.1 จบโปรแกรม 9](#_Toc179829838)

[รูปที่ 4 การกำหนดรูปแบบข้อมูล 10](#_Toc179829839)

[รูปที่ 4.1 ฟังก์ชันสำหรับเพิ่มข้อมูล 11](#_Toc179829840)

[รูปที่ 4.2 ฟังก์ชันสำหรับแสดงข้อมูลทั้งหมด 12](#_Toc179829841)

[รูปที่ 4.3 ฟังก์ชันสำหรับค้นหาข้อมูลตามรหัสสินค้า 13](#_Toc179829842)

[รูปที่ 4.4 ฟังก์ชันสำหรับอัปเดตข้อมูลสินค้า 14](#_Toc179829843)

[รูปที่ 4.5 ฟังก์ชันสำหรับลบข้อมูลสินค้า 15](#_Toc179829844)

[รูปที่ 4.6.1 ขั้นตอนการทำงานของฟังก์ชัน 15](#_Toc179829845)

[รูปที่ 4.6.2 เปิดไฟล์สินค้าในโหมดอ่านไบนารี 16](#_Toc179829846)

[รูปที่ 4.6.3 แยกข้อมูลแต่ละรายการออกมา 16](#_Toc179829847)

[รูปที่ 4.7 จัดเก็บข้อมูลสินค้า 17](#_Toc179829848)

[รูปที่ 4.7.1 สร้างเนื้อหารายงาน 18](#_Toc179829849)

[รูปที่ 4.7.2 บันทึกรายงานลงไฟล์ report.txt 18](#_Toc179829850)

[รูปที่ 4.8 เมนูสำหรับเลือกฟังก์ชัน 19](#_Toc179829851)

[รูปที่ 4.9 เพิ่มข้อมูลสินค้า 20](#_Toc179829852)

**คำนำ**

การจัดทำโครงงานในชื่อเรื่อง “การจัดการคลังสินค้า” นี้เป็นส่วนหนึ่งของวิชา COMPUTER PROGRAMMIMG ของหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศและเครือข่าย ภาควิชาเทคโนโลยีสารสนเทศคณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยี พระจอมเกล้าพระนครเหนือ เพื่อให้นักศึกษาได้นำความรู้ที่เรียนมาทั้งหมดมาประยุกต์ใช้ในการ พัฒนาโปรแกรมที่สามารถทำงานได้จริง โดยเน้นการออกแบบและเขียนโปรแกรมในภาษา Python ซึ่งเป็นภาษาที่เรียนมาในวิชา COMPUTER PROGRAMMING โดยโครงงานนี้จะช่วย การคิดวิเคราะห์ และแก้ปัญหาทางเทคนิค เพื่อเตรียมความพร้อมในการประกอบอาชีพด้านวิศวกรรมสารสนเทศและ เครือข่ายในอนาคต หากมีข้อผิดพลาดประการใด คณะผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ด้วย

# บทที่ 1 บทนำ

## 1.1 วัตถุประสงค์ของโครงงาน

1.1.1 เพื่อพัฒนาระบบที่สามารถจัดการหนังสือได้อย่างมีประสิทธิภาพ

1.1.2 เพื่อฝึกฝนทักษะการเขียนโปรแกรมด้วย Python

1.1.3 เพื่อเรียนรู้การจัดการข้อมูลและไฟล์

1.1.4 เพื่อเรียนรู้การทำงานร่วมกันเป็นทีม

## 1.2. ขอบเขตของโครงงาน

1.2.1ระบบจัดการข้อมูลหนังสือจะมีฟังก์ชันพื้นฐาน 7 ฟังก์ชัน 1.เพิ่มข้อมูล 2. แสดงข้อมูลทั้งหมด 3.ค้นหาข้อมูล 4.อัพเดตข้อมูล 5.ลบข้อมูล 6.สร้างรายงาน 7.ออกจากโปรแกรม

1.2.2ระบบการจัดเรียงสินค้าเก็บข้อมูลสินค้าไว้ใย text file ชื่อ report.txt ซึ่งมีจำนวนประเภทสินค้าทั้งหมด ประเภทสินค้า รายการสินค้าทั้งหมด เลขIDสินค้า ชื่อสินค้า ราคาสินค้า จำนวนสินค้า ผลรวมราคาสินค้า ผลรวมราคาสินค้า

## 1.3. ประโยชน์ที่ได้รับ

1.3.1 พัฒนาระบบที่สามารถจัดการหนังสือได้อย่างมีประสิทธิภาพ

1.3.2 พัฒนาทักษะการเขียนโปรแกรม

1.3.3 เรียนรู้การจัดการข้อมูลและไฟล์

1.3.4 เรียนรู้การทำงานร่วมกันเป็นทีม

## 1.4. เครื่องมือที่คาดว่าจะต้องใช้

1.4.1 ภาษา Python

1.4.2 Microsoft office

# บทที่ 2 ระบบการจัดการคลังสินค้า

## 2.1. ฟิลด์ในระบบจัดการคลังสินค้า

การจัดการข้อมูลหนังสือในระบบของคุณประกอบด้วย 7 ฟิลด์หลัก ซึ่งแต่ละฟิลด์มีรายละเอียด และความสำคัญดังนี้

**2.1.1 เพิ่มข้อมูล**

การเพิ่มข้อมูล คือ การเพิ่มข้อมูลสินค้าเข้าไปใหม่ในในคลังสินค้า

**2.1.2 แสดงข้อมูลทั้งหมด**

การแสดงข้อมูลทั้งหมดคือการแสดงข้อมูลสินค้าทั้งหมดไม่ว่าจะเป็นราคาหรือจำนวน

สินค้า ขนิดของสินค้าต่างๆ รายชื่อสินค้า เลขIDสินค้า

**2.1.3 ค้นหาข้อมูล**

ค้นหาคือการค้นหาข้อที่มีอยู่ในคลังสินค้า

**2.1.4 อัพเดตข้อมูล**

การอัพเดตข้อมูลคือการเพิ่มข้อมูลสินค้าทีเราต้องการจะเพิ่มเข้าไปไม่ว่าจะเป็นจำนวนสินค้าหรือราคาสินค้านะช่วงเวลานั้นหรือหรือชื่อสินค้า

**2.1.5. ลบข้อมูล**

การลบข้อมูลคือการลบข้อมูลที่เราไม่ต้องการออกไปเนื่องจากเช่นในสถานการสินค้าหมด

**2.1.6. สร้างรายงาน**

การสร้างรายงานคือการร้ายงานสินค้าจะมีราคารวมของสินค้าและบอกราคาแต่ละชนิดอย่างชัดเจน

**2.1.7 ออกจากโปรแกรม**

การจบการทำงานของโปรแกรม

## 2.2 สรุปฟิลด์ในระบบจัดการคลังสินค้า

ฟิลด์ทั้ง 7 นี้ทำให้ระบบจัดการข้อมูลหนังสือสามารถทำงานได้อย่างมีประสิทธิภาพ โดย ช่วยให้การจัดเก็บ การค้นหา การกรองข้อมูล และการแสดงผลข้อมูลเป็นไปได้อย่างรวดเร็วและมีประสิทธิภาพ นอกจากนี้ยังช่วยให้ผู้ใช้สามารถเข้าใจและเข้าถึงข้อมูลการจัดการคลังสินค้าได้ง่าย

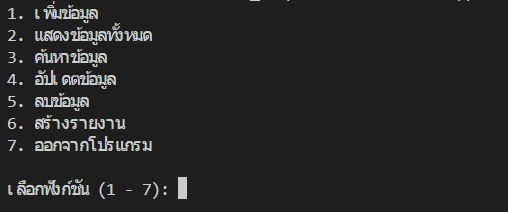
# บทที่3 การใช้โปรแกรม การจัดการคลังสินค้า

โปรแกรมการจัดการสินค้าคือการช่วยจำแนกประเภทสินค้าให้ดูสะดวกง่ายขึ้นและยังช่วยจัดประเภทของสินค้าให้ดูง่ายขึ้นโดยช่วยทำรายงานไปเก็บไว้ยังไฟล์ text

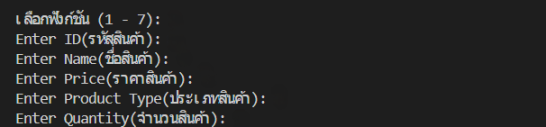
โปรแกรมจัดการสินค้าประกอบไปด้วย การเพิ่มข้อมูลที่จะเก็บข้อมูล ID,NAME,ราคา,ประเภท,จำนวน แสดงข้อมูลสินค้าทั้งหมดในโปรแกรม ค้นหาข้อมูลโดยการใช้เลข ID เพื่อค้นหาสะดวกต่อการใช้งาน อัพเดตข้อมูลสามารถเปลี่ยนแปลงข้อมูลได้แต่ถ้าเราไม่ต้องการเปลี่ยนข้อมูลนั้นให้เราเว้นว่างไว้ ลบข้อมูลโดยการใช้เลขเลขIDเพื่อลบข้อมูลทั้งหมดของเลข ID นั้น สร้างรายงานเพื่อทำสรุปของสินค้าแต่ละประเภทและจำนวณในแต่ละประเภทนั้นมีจำนวณเงินและจำนำนวณทั้งหมดของสินค้า จบการทำงานของโปรแกรม

## 3.1 สำหรับผู้ใช้งานโปรแกรม

การใช้โปรแกรม การเพิ่มข้อมูล

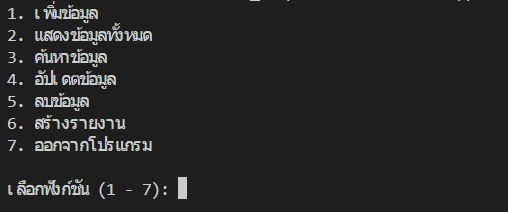
 3.1.1 กรอกหมายเลข 1.ตามวงสีแดงที่ละบุเพื่อเรียกฟังก์ชันเพิ่มข้อมูลที่ประกอบไปด้วย ID,NAME,ราคาสินค้า,ประเภทสินค้า,จำนวนสินค้า

รูปที่ 3.1.1 การเพิ่มข้อมูล

****3.1.2. เมื่อเมนูฟังก์ชันขึ้นมาแล้วจากนั้นก็สามารถระบุตามที่ต้องการได้

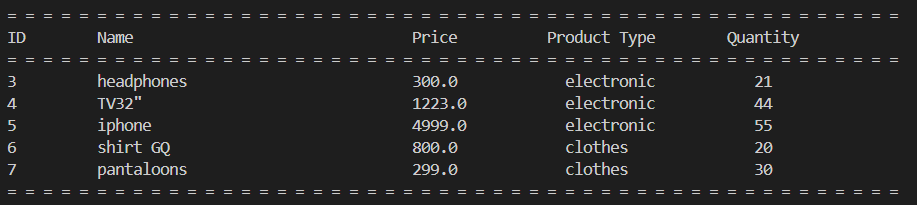
รูปที่ 3.1.2 การเลือกฟังก์ชัน

## 3.2 การใช้โปรแกรม การแสดงข้อมูลทั้งหมด

****3.2.1. กรอกหมายเลขที่2เมื่อแสดงข้อมูลทั้งหมดของสิ้นค้า

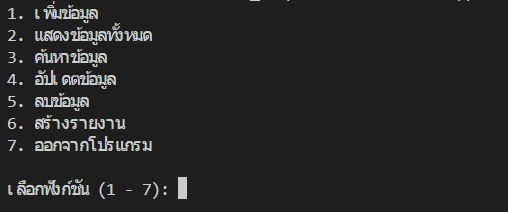
รูปที่ 3.2.1 การแสดงข้อมูลทั้งหมด

3.2.2. หลังจากกดหมายเลข2ข้อที่แล้วก็จะแสดงรายการสินค้ามาตามภาพที่ 2.2

****

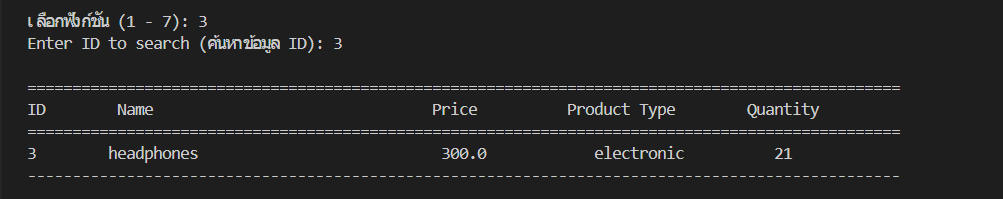
รูปที่ 3.2.2 แสดงข้อมูลสินค้าทั้งหมด

## 3.3 การใช้โปรแกรม การค้นหาข้อมูล

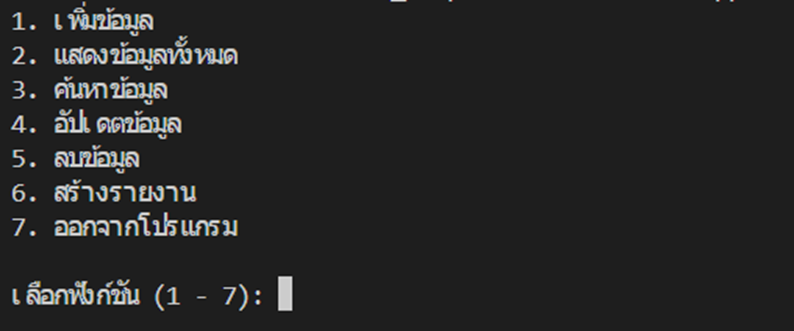
****3.3.1.กรอกหมายเลขที่3เพื่อเข้าฟังก์ชันในค้นหาข้อมูล ID ดูเลข ID ได้ตามรูปที่2.2

รูปที่ 3.3.1 การค้นหาข้อมูล

3.3.2.กรอกหมายเลข ID เพื่อทำการค้นหาข้อมูล ในตัวอย่างนี้เราจะทำการกรอกหมายเลข3เพื่อเป็นตัวอย่างในการค้นหาข้อมูล

รูปที่ 3.3.2 การค้นหาข้อมูล

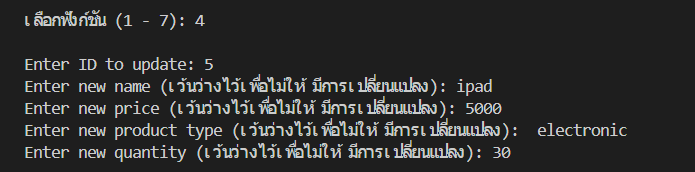
## 3.4 การใช้โปรแกรม การอัปเดตข้อมูล

3.4.1.กรอกหมายเลขที่4เพื่อทำการเลือกฟังก์ชันอัปเดตข้อมูลลงไปในคลังสินค้า

****

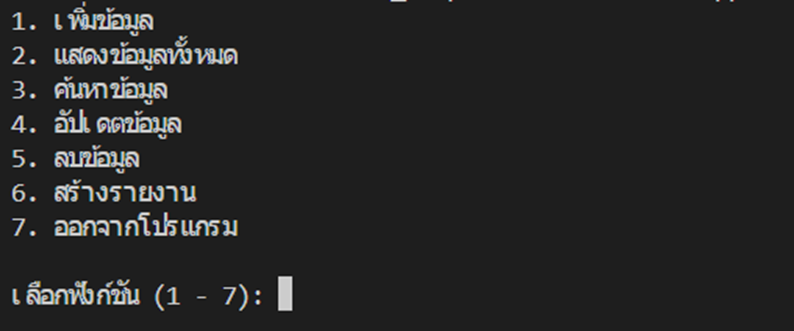
****

รูปที่3.4.1 การอัปเดตข้อมูล

****3.4.2.กรอก ID ที่ต้องการจะเปลี่ยน ในตัวอย่างนี้กรอกหมายเลข5เพื่อเปลี่ยนข้อมูลจาก iphone เป็น ipad เปลี่ยน price และ quantity แต่ product type ยังคงเดิม โดยสามารถดูข้อมูลสินค้าที่ยังไม่อัปเดตรูปที่2.2

รูปที่3.4.2 การอัปเดตข้อมูล

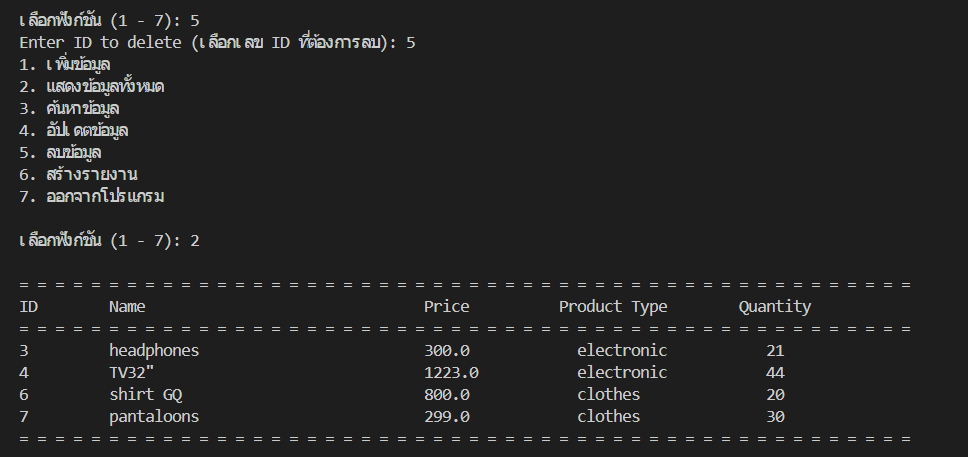
## 3.5. การใช้โปรแกรม การลบข้อมูล

 3.5.1.กรอกหมาย 5 เพื่อทำการลบข้อมูลในรายการข้อมูล

****

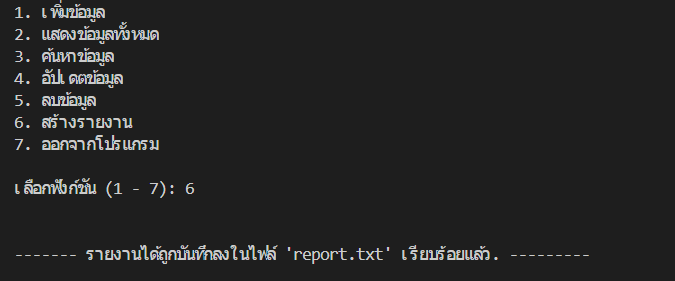
****

รูปที่ 3.5.1 การลบข้อมูล

**** 3.5.2.กรอก ID ที่ต้องการจะลบ โดยรูปที่ 5.2 กรอกหมายเลข 5 เพื่อเป็นตัวอย่างโดยสามารถดูข้อมูลสินค้าที่ยังไม่ได้ลบดังรูปที่ 2.2

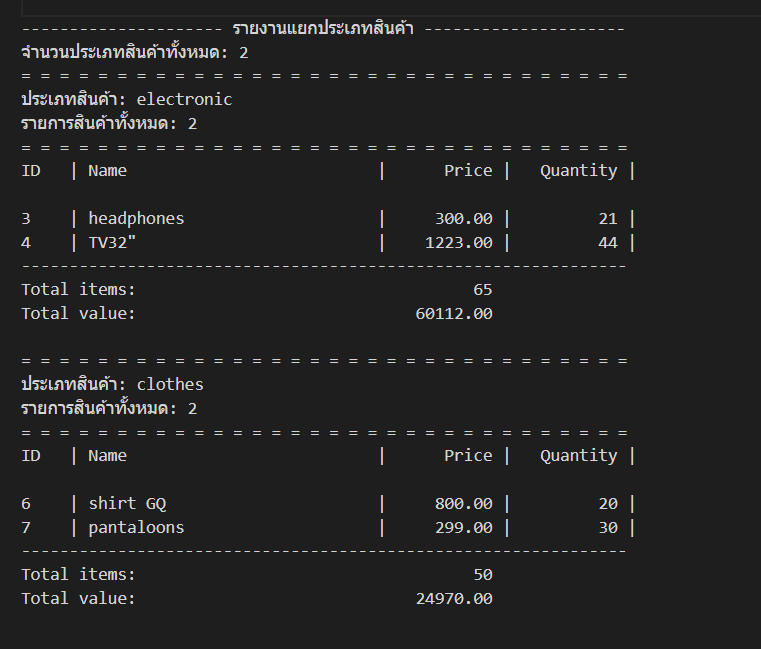
รูปที่ 3.5.2 การลบข้อมูลรายการสินค้า

## 3.6 การใช้โปรแกรม สร้างรายงาน

**** 3.6.1 กรอกหมาย 6 เพื่อสร้างรายงาน

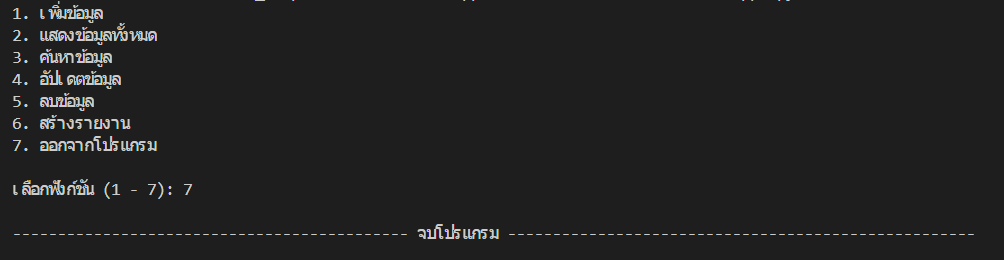
รูปที่ 3.6.1 การสร้างรายงาน

3.6.2 รายงานแยกประเภทสินค้า

****

รูปที่ 3.6.2 รายงานแยกประเภทสินค้า

## 3.7 การใช้โปรแกรม ออกจากโปรแกรม

**** 3.7.1 กรอกหมายเลข 7 เพื่อออกจากโปรแกรม

รูปที่ 3.7.1 จบโปรแกรม

# บทที่4 อธิบายการทำงานของ code

โค้ดนี้เป็นโปรแกรมที่ทำงานกับไฟล์ไบนารีสำหรับจัดการข้อมูลสินค้า โดยใช้โมดูล สำหรับการแปลงข้อมูลเป็นไบนารี และมีฟังก์ชันต่างๆ สำหรับการเพิ่ม, แสดง, ค้นหา, อัปเดต, ลบข้อมูลสินค้า และสร้างรายงานแยกตามประเภทสินค้า

## การกำหนดรูปแบบข้อมูล

record\_format: กำหนดรูปแบบของข้อมูลแต่ละรายการสินค้าในไฟล์ ซึ่งประกอบด้วย:

I: ข้อมูลชนิด unsigned int (ID ของสินค้า) ขนาด 4 bytes

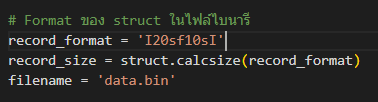
20s: ชื่อสินค้า เป็นสตริงความยาว 20 ตัวอักษร (20 bytes)

f: ราคาสินค้า เป็นชนิด float (4 bytes)

10s: ประเภทสินค้า เป็นสตริงความยาว 10 ตัวอักษร (10 bytes)

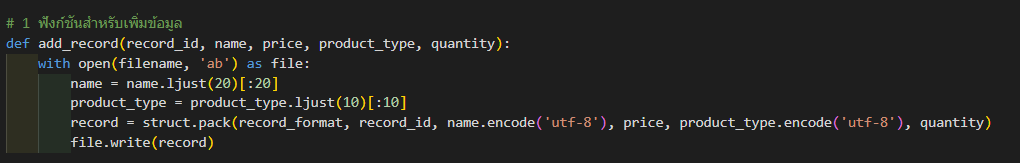
I: จำนวนสินค้า เป็นชนิด unsigned int (4 bytes)

record\_size: ใช้ฟังก์ชัน struct.calcsize เพื่อคำนวณขนาดของข้อมูลแต่ละรายการตามรูปแบบ record\_format ซึ่งจะเป็น 42 bytes ต่อ 1 รายการ filename: กำหนดชื่อไฟล์ที่จะใช้เก็บข้อมูลเป็น ‘data.bin’

****

รูปที่ 4 การกำหนดรูปแบบข้อมูล

## 4.1 ฟังก์ชันสำหรับเพิ่มข้อมูล

ฟังก์ชันนี้ใช้สำหรับเพิ่มข้อมูลสินค้าใหม่ลงในไฟล์ใช้ ljust() เพื่อตัดหรือเพิ่มช่องว่างให้ชื่อสินค้า (name) และประเภทสินค้า (product\_type) มีความยาวตามที่กำหนด (20 ตัวอักษรและ 10 ตัวอักษรตามลำดับ)ใช้ struct.pack() เพื่อแปลงข้อมูลสินค้าให้อยู่ในรูปแบบไบนารีตาม record\_formatใช้โหมด ab (append binary) เพื่อเปิดไฟล์และเขียนข้อมูลใหม่ต่อท้ายไฟล์เดิม23

รูปที่ 4.1 ฟังก์ชันสำหรับเพิ่มข้อมูล

## 4.2 ฟังก์ชันสำหรับแสดงข้อมูลทั้งหมด

ฟังก์ชันนี้ใช้แสดงข้อมูลสินค้าทั้งหมดจากไฟล์ใช้ struct.unpack() เพื่อแปลงข้อมูลไบนารีกลับเป็นข้อมูลแบบเดิม แล้วนำมาแสดงผลบนหน้าจอใช้ rb (read binary) เพื่อเปิดไฟล์ในโหมดอ่านไบนารี

print() : แสดงบรรทัดว่างเพื่อความสวยงาม

print("= "\* 50) : แสดงเส้นคั่นตารางโดยการพิมพ์เครื่องหมาย = จำนวน 50 ครั้ง (คูณด้วยช่องว่างเพื่อแสดงเป็นเส้นยาว)

print(f"{'ID':<10}…): แสดงหัวตารางด้วยการจัดรูปแบบข้อมูล แต่ละฟิลด์ (ID, Name, Price, Product Type, Quantity) มีความกว้างที่กำหนดไว้

print("= "\* 50) : แสดงเส้นคั่นตารางหลังหัวตาราง

with open(filename, 'rb') as file : เปิดไฟล์ filename ในโหมดอ่านแบบไบนารี (rb)

with ทำหน้าที่ปิดไฟล์อัตโนมัติเมื่อจบการทำงาน

while chunk := file.read(record\_size) : อ่านไฟล์ทีละบล็อก (ขนาดเท่ากับrecord\_size) จนกว่าจะหมดไฟล์ ตัวแปร chunk จะเก็บข้อมูลในรูปแบบไบนารีที่อ่านมาในแต่ละรอบ

record = struct.unpack(record\_format, chunk) : ใช้ struct.unpack เพื่อแปลงข้อมูลไบนารีใน chunk ตามรูปแบบที่กำหนดใน record\_format ซึ่งผลลัพธ์จะเก็บในตัวแปร record

print(f"{record : แสดงข้อมูลในแต่ละบรรทัดตามลำดับฟิลด์ที่อ่านได้จากตัวแปร record โดยใช้การจัดรูปแบบข้อมูลให้เรียงตามความกว้างที่กำหนด:

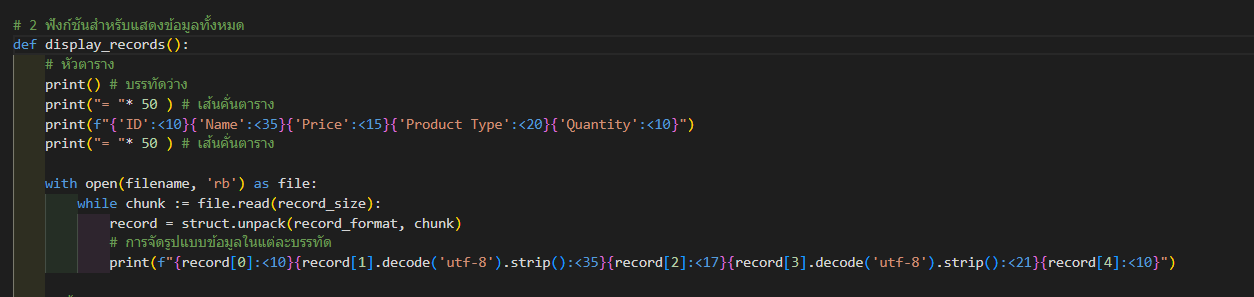
record[0]:<10: เก็บข้อมูล ID ให้อยู่ใน record[0]

record[1].decode('utf-8').strip():<35: เก็บชื่อสินค้า ให้อยู่ใน record[1] ซึ่งถูกแปลงจากไบนารีเป็นข้อความด้วย decode('utf-8') และตัดช่องว่างหน้า-หลังด้วย strip()

record[2]:<17: เก็บราคา ให้อยู่ในrecord[2]

record[3].decode('utf-8').strip():<21: เก็บประเภทสินค้า ให้อยู่ใน record[3] ที่ถูกแปลงเป็นข้อความและจัดรูปแบบให้ชิดซ้าย

record[4]:<10: เก็บจำนวนสินค้า ให้อยู่ใน record[4]



รูปที่ 4.2 ฟังก์ชันสำหรับแสดงข้อมูลทั้งหมด

## 4.3 ฟังก์ชันสำหรับค้นหาข้อมูลตามรหัสสินค้า

ฟังก์ชันนี้ใช้ค้นหาข้อมูลสินค้าโดยใช้ ID ของสินค้าในการอ่านข้อมูลจากไฟล์และตรวจสอบว่ารหัสสินค้า (record[0]) ตรงกับ search\_id ที่ค้นหาหรือไม่ หากพบจะคืนค่าข้อมูลสินค้านั้น

with open(filename, 'rb') as file : เปิดไฟล์ที่ชื่อ filename ในโหมดไบนารี (rb) ซึ่งเป็นโหมดที่เปิดไฟล์เพื่ออ่านข้อมูลในรูปแบบไบนารี การใช้ with จะช่วยจัดการปิดไฟล์อัตโนมัติเมื่อสิ้นสุดการทำงานภายในบล็อกนี้

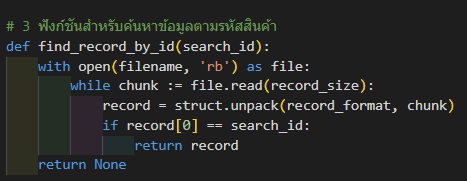
while chunk := file.read(record\_size) : วนลูปอ่านไฟล์ โดยขนาดของข้อมูลที่อ่านจะมีขนาดเท่ากับ record\_size ตัวแปร chunk จะเก็บข้อมูลที่อ่านมา ถ้าอ่านจนไม่มีข้อมูลเหลือแล้ว (สิ้นสุดไฟล์) ลูปจะหยุดทำงาน

record = struct.unpack(record\_format, chunk) : เป็นฟังก์ชันที่ใช้แปลงข้อมูลไบนารีในรูปแบบ chunk(อ่านข้อมูลออกเป็นช่วง ๆ ) ให้เป็นข้อมูลที่สามารถนำมาใช้ได้ตามรูปแบบที่กำหนดโดย record\_format ซึ่งเป็นโครงสร้างข้อมูล (เช่น integer, string, float) ที่เก็บไว้ใน record นั้น ๆ

ผลลัพธ์ที่ได้จะถูกเก็บไว้ในตัวแปร record

if record[0] == search\_id : ตรวจสอบว่าค่าของ record[0]หรือ ID เท่ากับหมายเลขระบุข้อมูลว่าตรงกับ search\_id ที่ต้องการค้นหาหรือไม่

return record ถ้าพบ record ที่ record[0] ตรงกับ search\_id จะคืนค่าข้อมูลของ record กลับมา

 return None ถ้าไม่พบ record ที่ตรงกับ search\_id หลังจากวนลูปจนหมดไฟล์แล้ว ฟังก์ชันจะคืนค่า None

รูปที่ 4.3 ฟังก์ชันสำหรับค้นหาข้อมูลตามรหัสสินค้า

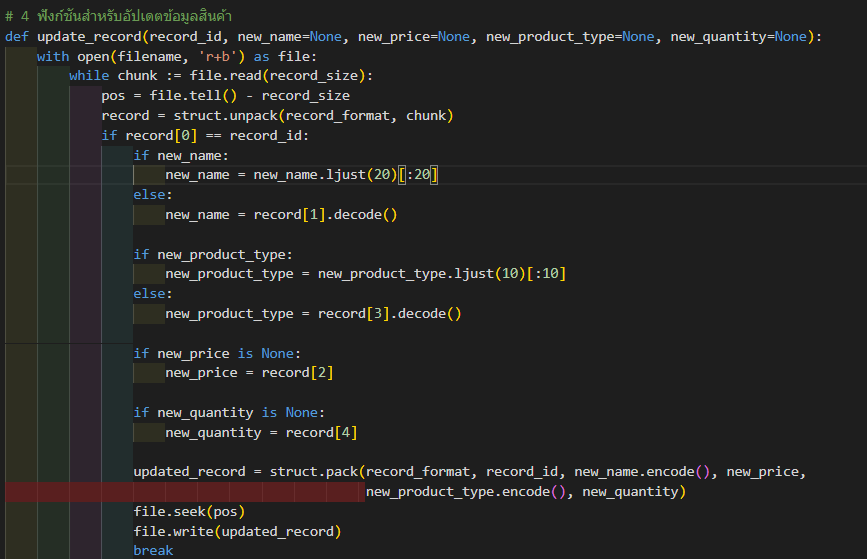
## 4.4 ฟังก์ชันสำหรับอัปเดตข้อมูลสินค้า

ฟังก์ชัน update\_record() ทำหน้าที่ในการอัปเดตข้อมูลสินค้าที่อยู่ในไฟล์ไบนารี โดยสามารถอัปเดตชื่อสินค้า (new\_name), ราคาสินค้า (new\_price), ประเภทสินค้า (new\_product\_type), และจำนวนสินค้า (new\_quantity) ได้ตามที่ต้องการ โครงสร้างการทำงาน

เปิดไฟล์ข้อมูลต้นฉบับ (filename) ในโหมดอ่านไบนารี ('rb+') ซึ่งหมายถึงสามารถอ่านและเขียนไฟล์ได้ใช้ลูป while เพื่ออ่านข้อมูลทีละชุดจากไฟล์ โดยแต่ละชุดข้อมูลจะถูกอ่านตามขนาดที่กำหนดในตัวแปร record\_sizeสำหรับแต่ละชุดข้อมูล จะบันทึกตำแหน่งปัจจุบันของไฟล์ (ใช้ file.tell() เพื่อบันทึกตำแหน่ง) และทำการถอดรหัสข้อมูลไบนารีโดยใช้ struct.unpack() ให้ได้ข้อมูลในรูปแบบที่สามารถอ่านได้ตามฟอร์แมตที่กำหนดใน record\_formatตรวจสอบว่า record[0] (ซึ่งคือ record\_id ของข้อมูลที่อ่านมา) ตรงกับ record\_id ที่ต้องการอัปเดตหรือไม่ถ้าตรงกับ record\_id ที่ต้องการอัปเดต จะเริ่มกระบวนการอัปเดตข้อมูลแต่ละส่วนตามที่ส่งเข้ามาใน

ถ้ามีค่า new\_name จะปรับชื่อสินค้าโดยจัดรูปแบบให้มีความยาว 20 ตัวอักษร

ถ้าไม่มีค่า new\_name จะใช้ชื่อเดิมจากข้อมูลที่อ่านมาเช่นเดียวกันกับ new\_product\_type, new\_price และ new\_quantity ถ้ามีค่าที่ส่งมาใหม่ก็จะอัปเดต แต่ถ้าไม่มีค่าก็จะใช้ข้อมูลเดิม

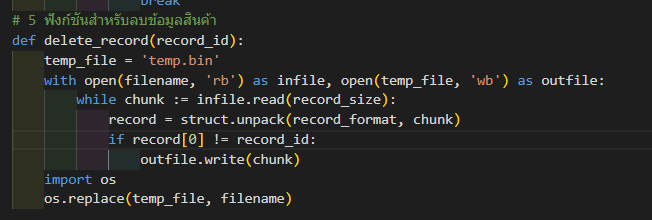
เมื่ออัปเดตข้อมูลเรียบร้อย จะทำการแพ็คข้อมูลกลับเป็นไบนารีด้วย struct.pack() แล้วเขียนข้อมูลนั้นกลับไปที่ตำแหน่งเดิมในไฟล์โดยใช้ file.seek() เพื่อเลื่อนไปยังตำแหน่งเดิมในไฟล์ แล้วใช้ file.write() เพื่อเขียนข้อมูลที่อัปเดตใหม่ลงไปหลังจากเขียนข้อมูลที่อัปเดตแล้วจะออกจากลูปโดยใช้ break เพื่อหยุดการทำงาน

รูปที่ 4.4 ฟังก์ชันสำหรับอัปเดตข้อมูลสินค้า

## 4.5 ฟังก์ชันสำหรับลบข้อมูลสินค้า

เป็นฟังก์ชันที่ทำหน้าที่ในการลบข้อมูลที่มี id ตรงกับค่าที่ที่มีในไฟล์ข้อมูล โดยวิธีการทำงานสามารถกำหนดไฟล์ชั่วคราวชื่อ temp\_file เพื่อใช้เก็บข้อมูลที่เหลือหลังจากการลบเปิดไฟล์ข้อมูลต้นฉบับ ด้วยโหมดอ่านแบบไบนารี ('rb') และเปิดไฟล์ชั่วคราวด้วยโหมดเขียนแบบไบนารี ('wb')

ใช้ลูปวนเพื่ออ่านข้อมูลจากไฟล์ต้นฉบับทีละส่วนและใช้ struct.unpack() เพื่อถอดรหัสข้อมูลไบนารีให้กลับมาเป็นข้อมูลแบบที่อ่านได้ โดย record\_format คือรูปแบบของข้อมูลที่ต้องการถอดรหัส

ตรวจสอบว่า record[0] ซึ่งเป็น id ของข้อมูลนั้น ตรงกับ id ที่เราต้องการลบหรือไม่ถ้าตรง ก็จะเขียนข้อมูลลงไฟล์ชั่วคราวเมื่อเสร็จสิ้นกระบวนการอ่านและเขียนแล้ว จะใช้ os.replace() เพื่อแทนที่ไฟล์ temp\_fileเข้าไปบันทึกในไฟล้ต้นฉบับ

รูปที่ 4.5 ฟังก์ชันสำหรับลบข้อมูลสินค้า

## 4.6 ฟังก์ชันสำหรับเขียนรายงาน

หน้าที่ของฟังก์ชัน generate\_report\_by\_product\_type คือการสร้างรายงานที่จัดหมวดหมู่สินค้าแต่ละประเภท โดยจะแสดงรายละเอียดของสินค้าที่อยู่ในประเภทนั้น ๆ รวมถึงสรุปจำนวนสินค้าทั้งหมดและมูลค่ารวมของสินค้าในประเภทนั้น แล้วบันทึกข้อมูลในรูปแบบไฟล์ .txt

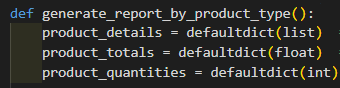
**4.6.1 ขั้นตอนการทำงานของฟังก์ชัน**

ใช้ defaultdict ในการเก็บข้อมูลตามประเภทสินค้า โดย:

product\_details: เก็บรายละเอียดสินค้าตามแต่ประเภทในรูปแบบของรายการ (list)

product\_totals: เก็บผลรวมของราคาสินค้าตามประเภทนั้น ๆ

product\_quantities: เก็บจำนวนสินค้ารวมของแต่ละประเภท

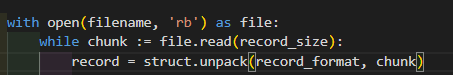
****

รูปที่ 4.6.1 ขั้นตอนการทำงานของฟังก์ชัน

**4.6.2 เปิดไฟล์สินค้าในโหมดอ่านไบนารี**

with open(filename, 'rb') as file :เปิดไฟล์ที่ชื่อว่า filename ในโหมดอ่านแบบไบนารี ('rb') with ทำให้เมื่อการทำงานภายในบล็อกเสร็จสิ้น ไฟล์จะถูกปิดอัตโนมัติ

ใช้ฟังก์ชัน struct.unpack() เพื่อแปลงข้อมูลไบนารี (chunk) ให้อยู่ในรูปแบบของข้อมูลตามที่ระบุใน record\_format



รูปที่ 4.6.2 เปิดไฟล์สินค้าในโหมดอ่านไบนารี

**4.6.3 แยกข้อมูลแต่ละรายการออกมา**

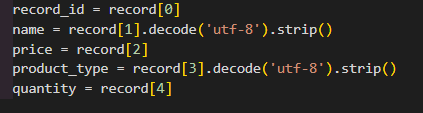
กำหนดค่าให้กับตัวแปร record\_id โดยนำค่าแรกในรายการ record[0] มาเก็บไว้ record\_id

นำค่าที่สองในรายการ record[1] มาเก็บในตัวแปร name โดยใช้ decode('tuff-8') เพื่อแปลงข้อมูลที่อยู่ในรูปแบบไบต์ (bytes) ให้เป็นสตริง (string) ที่เข้ารหัสด้วย UTF-8 และใช้ strip() เพื่อลบช่องว่าง (whitespace)

กำหนดค่าให้กับตัวแปร price โดยนำข้อมูลมาจาก record[2]

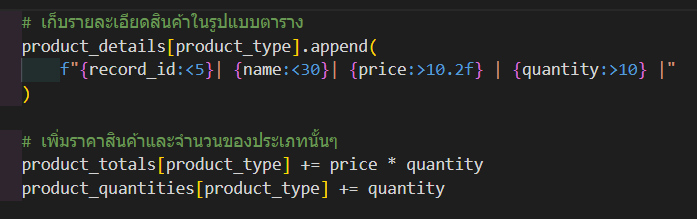
กำหนดตัวแปรชื่อ product type มาเก็บข้อมูลจาก record[3] โดยใช้ decode('tuff-8') เพื่อแปลงข้อมูลที่อยู่ในรูปแบบไบต์ (bytes) ให้เป็นสตริง (string) ที่เข้ารหัสด้วย UTF-8 และใช้ strip() เพื่อลบช่องว่างของข้อความ

กำหนดนตัวแปรชื่อ quantity โดยกำหนดนข้อมูลจาก record[4]



รูปที่ 4.6.3 แยกข้อมูลแต่ละรายการออกมา

## 4.7 จัดเก็บข้อมูลสินค้า

จากนั้นเราเพิ่มข้อมูลสินค้าลงในรายการ product\_type โดยจัดรูปแบบเป็นตารางแสดงรหัสสินค้า ชื่อสินค้า ราคา และจำนวน จากนั้นคำนวณผลรวมของราคาสินค้าในประเภทนั้นโดยนำราคาสินค้า (price) คูณด้วยจำนวน (quantity) แล้วบวกเพิ่มใน product\_type พร้อมทั้งบวกจำนวนสินค้าที่เพิ่มเข้ามาใน product\_type เพื่อสะสมจำนวนสินค้าในประเภทนั้นๆ

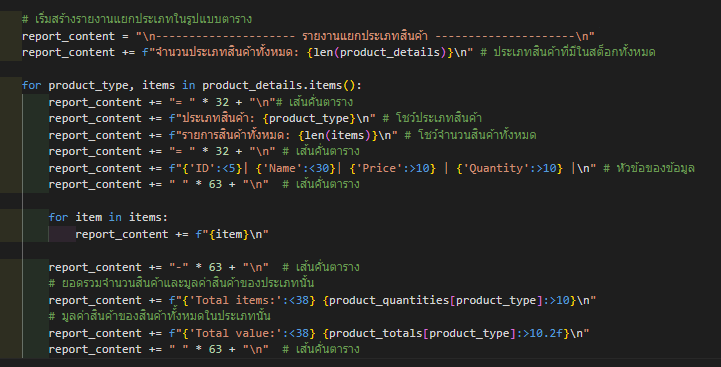
รูปที่ 4.7 จัดเก็บข้อมูลสินค้า

**4.7.1 สร้างเนื้อหารายงาน**

การสร้างรายงานในรูปแบบตารางเพื่อนำเสนอข้อมูลสินค้าแยกตามประเภท โดยขั้นตอนและการทำงานจะเริ่มต้นจากการสร้างหัวรายงาน report\_content ซึ่งประกอบด้วยจำนวนประเภทสินค้าที่มีทั้งหมดในระบบ ต่อมาจะทำการวนลูปเพื่อดึงข้อมูลในแต่ละประเภทสินค้า และจัดทำรายงานตามประเภทนั้นๆ โดยภายในแต่ละประเภทสินค้า จะเพิ่มบรรทัดที่บอกชื่อประเภทสินค้า และจำนวนรายการสินค้าที่อยู่ในประเภทนั้น

หลังจากนั้นจะสร้างหัวตารางที่ประกอบไปด้วย รหัสสินค้า ID, ชื่อสินค้า Name, ราคา Price และจำนวน Quantity

จากนั้นจะทำลูปที่วนเข้าไปดูข้อมูลแล้วนำมาแสดงรายการสินค้าทั้งหมดในประเภทนั้นๆ โดยแต่ละรายการจะถูกเพิ่มเข้าไปใน report\_content ด้วยข้อมูลของสินค้าจริง ที่มาจากข้อมูลที่เราเพิ่มเข้ามาให้ระบบ

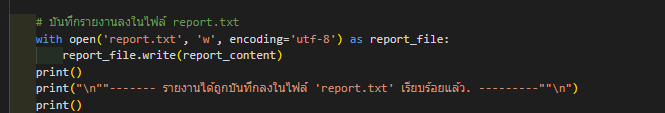
 เมื่อรายการสินค้าถูกแสดงทั้งหมดแล้ว และทำการรวมจำนวนของสินค้าทั้งหมดจากนั้นเราจะทำการเพิ่มราคารวมของสินค้ารวมของสินค้าในแต่ละประเภทนั้น

รูปที่ 4.7.1 สร้างเนื้อหารายงาน

**4.7.2 บันทึกรายงานลงไฟล์ report.txt**

เราจะใช้ทำสั่งสร้างไฟล์ report.txt โดยการเปิดไฟล์ในโหมดเขียน และ เช้ารหัสเป็นป็น

utf-8 เพื่อรองรับการใช้ภาษาไทยหรืออักขระพิเศษอื่นๆ เขียนเนื้อหาของรายงานแล้วเก็บไว้ในตัวแปร report\_content ลงในไฟล์ จากนั้นปิดไฟล์เมื่อการเขียนเสร็จสมบูรณ์ จัดการปิดไฟล์ให้อัตโนมัติเมื่อใช้ with open() เพื่อป้องกันการตั้งชื่อไฟล์ผิด หลังจากบันทึกจะแสดงข้อความว่า รายงานได้ถูกบันถึง ลงไฟล์ report.txt แล้ว

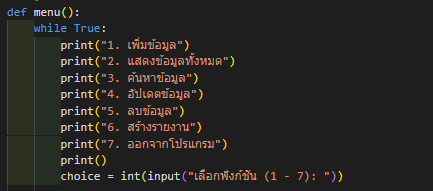


รูปที่ 4.7.2 บันทึกรายงานลงไฟล์ report.txt

## 4.8 เมนูสำหรับเลือกฟังก์ชัน

ฟังก์ชัน menu ทำหน้าที่เป็นเมนูหลักของโปรแกรมที่ให้ผู้ใช้สามารถเลือกฟังก์ชันที่ต้องการดำเนินการ เช่น เพิ่มข้อมูล แสดงข้อมูล ค้นหาข้อมูล อัปเดต ลบ หรือสร้างรายงาน ออกจากโปรแกรม

เลือกฟังก์ชัน (1 - 7): รับอินพุตจากผู้ใช้ โดยให้ผู้ใช้เลือกตัวเลือกที่ต้องการ



รูปที่ 4.8 เมนูสำหรับเลือกฟังก์ชัน

## 4.9 เพิ่มข้อมูลสินค้า

ถ้าได้รับอินพุต 1:

จะให้ผู้ใช้ใส่ข้อมูล ID, Name, Price, Product Type, Quantity แล้วส่งค่าไปที่ฟังช์ add\_record

ถ้าได้รับอินพุต 2:

จะเรียกใช้ฟังก์ชัน display\_records() เพื่อแสดงรายการสินค้าทั้งหมดที่มีอยู่ในระบบ

ถ้าได้รับอินพุต 3: ค้นหาข้อมูล

จะให้ผู้ใช้ป้อน id เพื่อค้นหาสินค้าเฉพาะตัวตามรหัสสินค้านั้น ถ้าพบสินค้า จะแสดงรายละเอียดของสินค้านั้นโดยแสดง ID, Name, Price, Product Type, และ Quantity

ถ้าได้รับอินพุต 4:

จะให้ผู้ใช้ป้อน id ของสินค้าที่ต้องการแก้ไข แล้วจะรับข้อมูลใหม่จากผู้ใช้ป้อน ชื่อใหม่ , ราคาที่แก้ไข , ประเภทสินค้าใหม่ , และจำนวนใหม่ หากไม่ต้องการเปลื่ยนให้เว้นว่างไว หลังจากนั้นจะเรียกฟังก์ชัน update\_record() เพื่ออัปเดตข้อมูลของสินค้านั้นในระบบ

ถ้าได้รับอินพุต 5:

จะให้ผู้ใช้ป้อน id ของสินค้าที่ต้องการลบจากระบบ และเรียกใช้ฟังก์ชัน delete\_record() เพื่อทำการลบสินค้านั้นออกจากระบบ

ถ้าได้รับอินพุต 6:

จะเรียกใช้ฟังก์ชัน generate\_report\_by\_product\_type() เพื่อสร้างรายงานที่แบ่งตามประเภทสินค้า และอาจแสดงหรือบันทึกรายงานนี้ในรูปแบบไฟล์

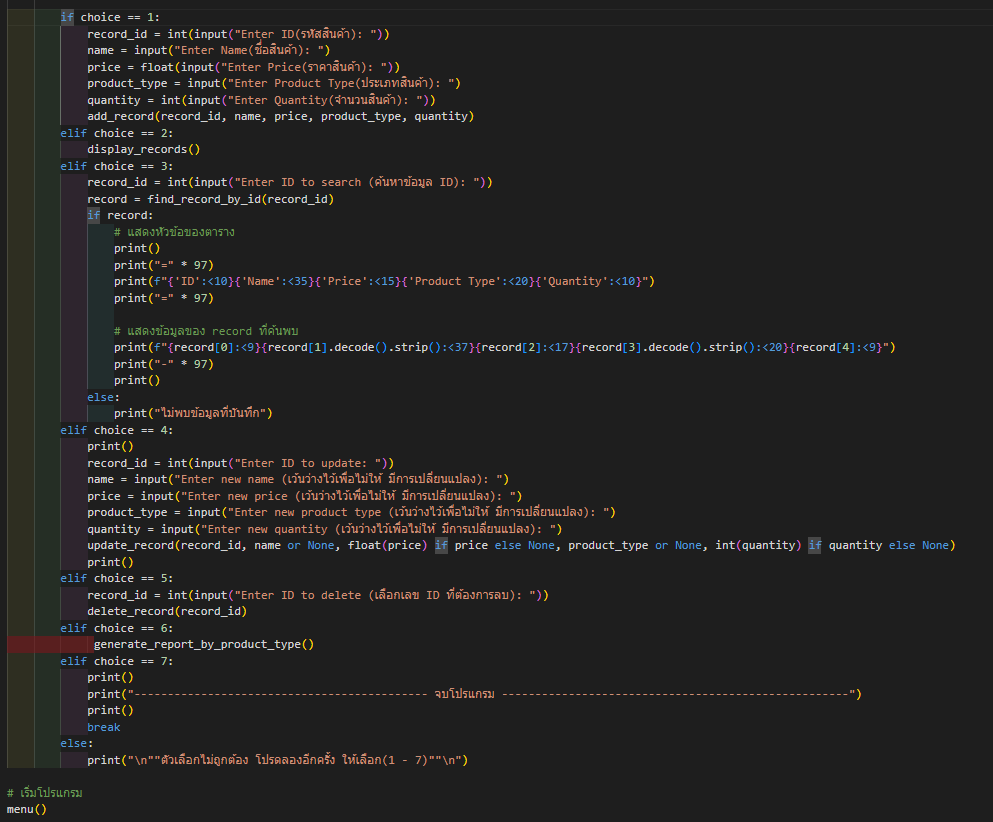
ถ้าได้รับอินพุต 7:

จะ จบการทำงานของโปรแกรม และทำงานออกจากลูป

ถ้าไม่มีเงื่อนไขใดตรง:

โค้ดจะแสดงข้อความเตือนว่า "ตัวเลือกไม่ถูกต้อง โปรดลองอีกครั้ง

ซึ่งผู้ใช้ป้อน ตัวเลือกที่กำหนด 1 – 7

****

รูปที่ 4.9 เพิ่มข้อมูลสินค้า