**A logo with a design on it

Description automatically generated**

**ใบงานที่ 9**

**เรื่อง Page Replacement**

**เสนอ**

**อาจารย์ปิยพล ยืนยงสถาวร**

**จัดทำโดย**

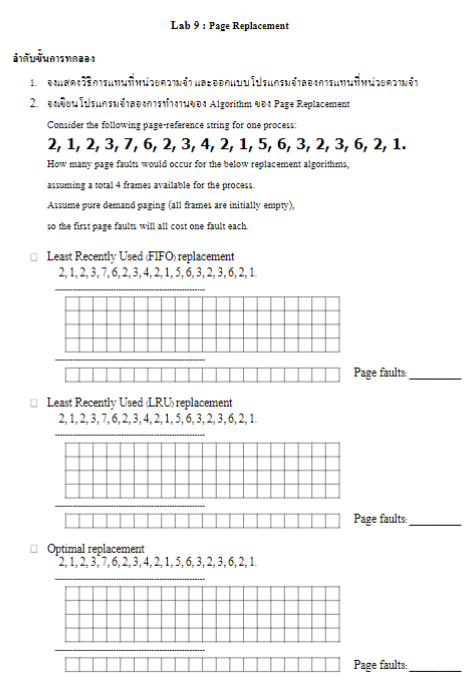
**นาย กวีวัธน์ กาญจน์สุพัฒนากุล 65543206003-7**

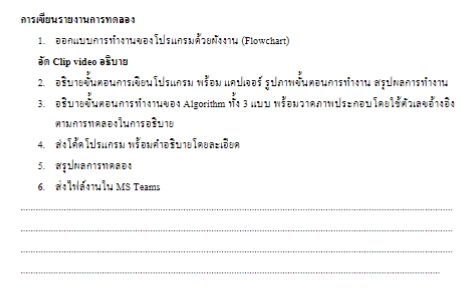
**ใบงานนี้เป็นส่วนหนึ่งของรายวิชา ระบบปฏิบัติการ**

**หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

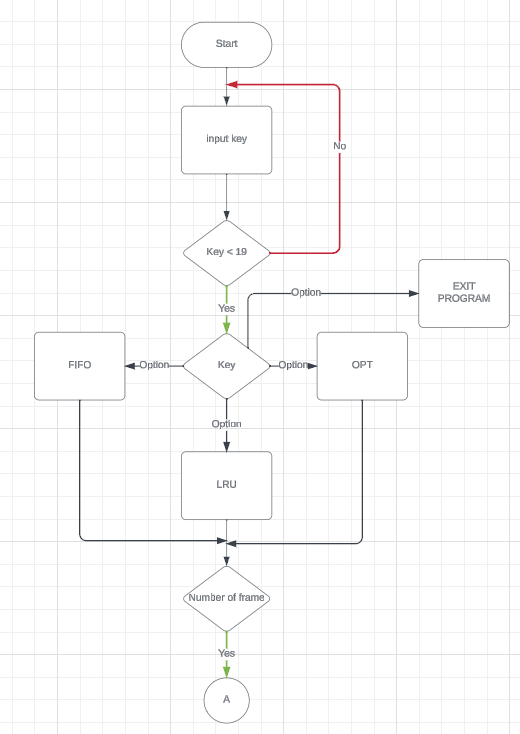
**มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา**

**ประจำภาคที่ 2 ปีการศึกษา 2566**

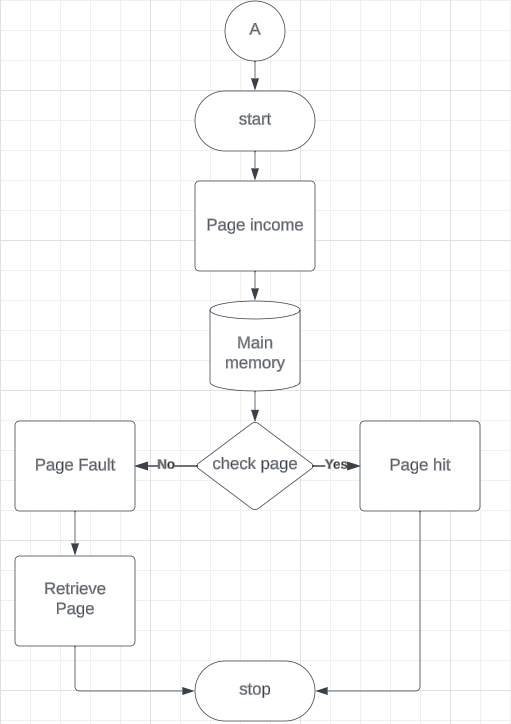


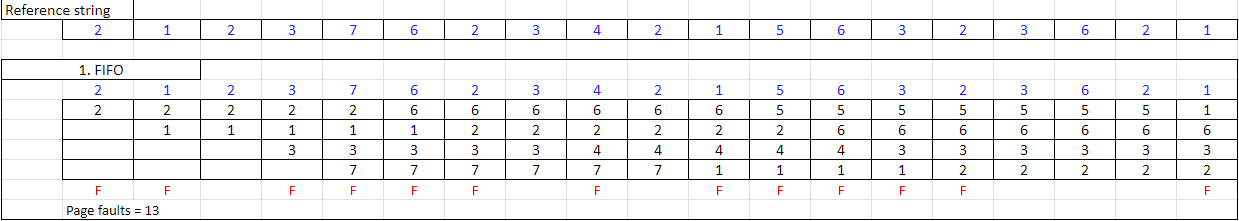


Flowchart Maincode

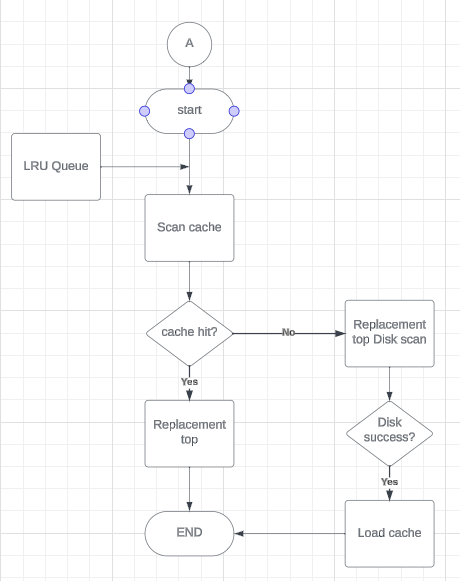


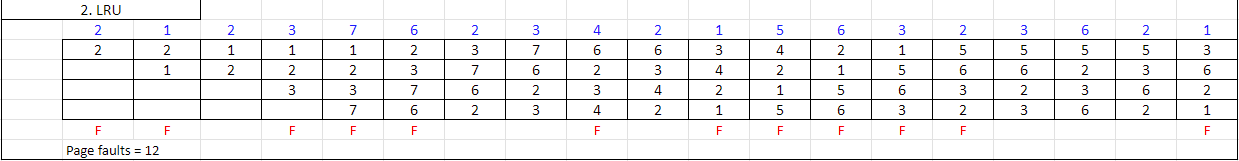
Flowchart FIFO



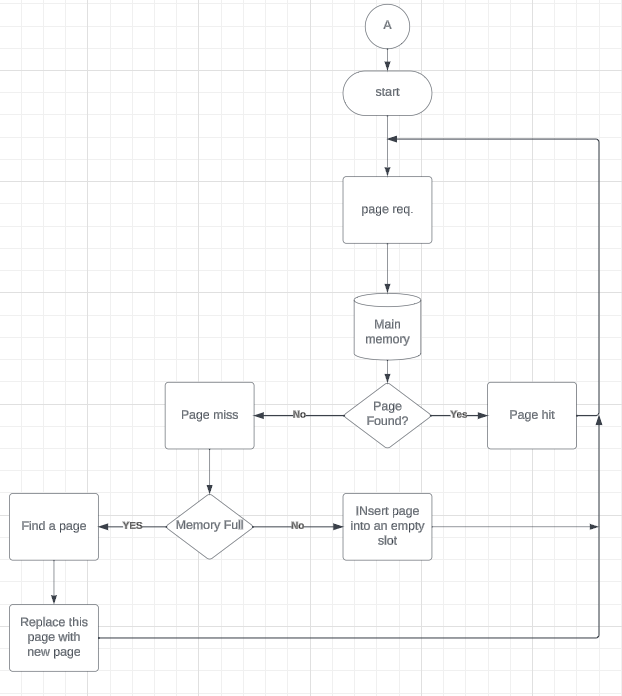


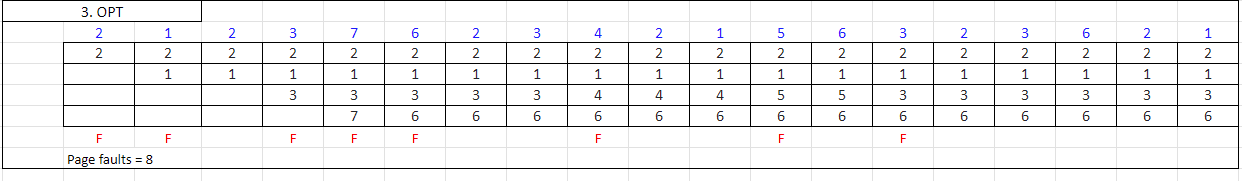
Flowchart LRU



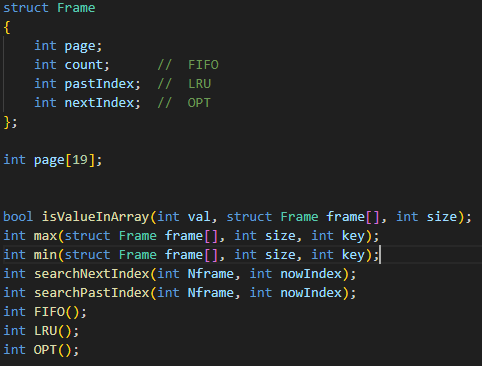


Flowchart OPT





Coding

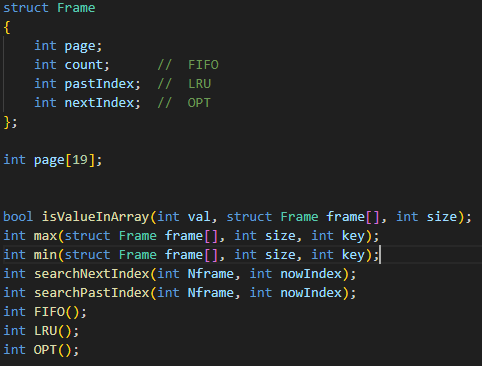


1. **ไฟล์ส่วนของ Header และการนิยาม Struct:**

* โปรแกรมนำเข้าไฟล์ส่วนของภาษา C ที่จำเป็น (stdio.h สำหรับการนำเข้า/ส่งออก และ stdbool.h สำหรับการใช้งาน boolean)
* นิยามโครงสร้างที่ชื่อ Frame เพื่อแทนตัวเฟรมหน้า (page frame) ในหน่วยความจำ มีฟิลด์สำหรับหมายเลขหน้า (page), ตัวนับสำหรับอัลกอริทึม FIFO (count), และดัชนีสำหรับอัลกอริทึม LRU (pastIndex) และ OPT (nextIndex)

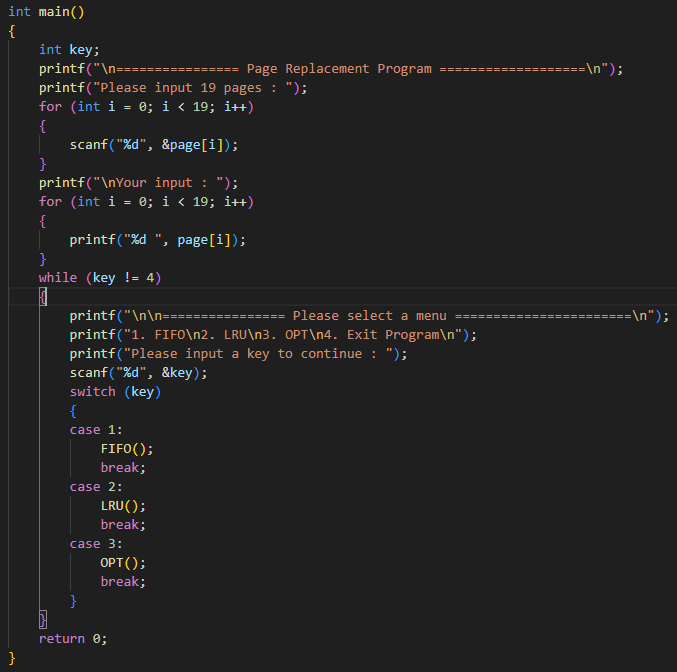
1. **ตัวแปร Global:**

* ประกาศอาร์เรย์ page[19] เพื่อเก็บหน้า 19 หน้าที่จะถูกนำมาใช้เป็นอินพุตสำหรับอัลกอริทึมการแทนหน้า



1. **ฟังก์ชันประการตัวแบบ:**

* ประกาศตัวแบบของฟังก์ชัน เช่น isValueInArray, max, min, searchNextIndex, searchPastIndex, และ 3 อัลกอริทึมหลัก FIFO, LRU, และ OPT



1. **การประกาศตัวแปร:**

* int key ประกาศตัวแปรจำนวนเต็มชื่อ key เพื่อเก็บค่าที่ผู้ใช้เลือกจากเมนู.

1. **การรับข้อมูลจากผู้ใช้:**

* โปรแกรมแสดงข้อความให้ผู้ใช้ป้อนหน้าข้อมูล 19 หน้า โดยใช้ลูป for และจัดเก็บข้อมูลที่ป้อนลงในอาร์เรย์ชื่อ page.

1. **การแสดงผลข้อมูลที่ผู้ใช้ป้อน:**

* โปรแกรมแสดงผลข้อมูลที่ผู้ใช้ป้อนเพื่อยืนยันหน้าข้อมูลที่ป้อน.

1. **เมนูและลูป:**

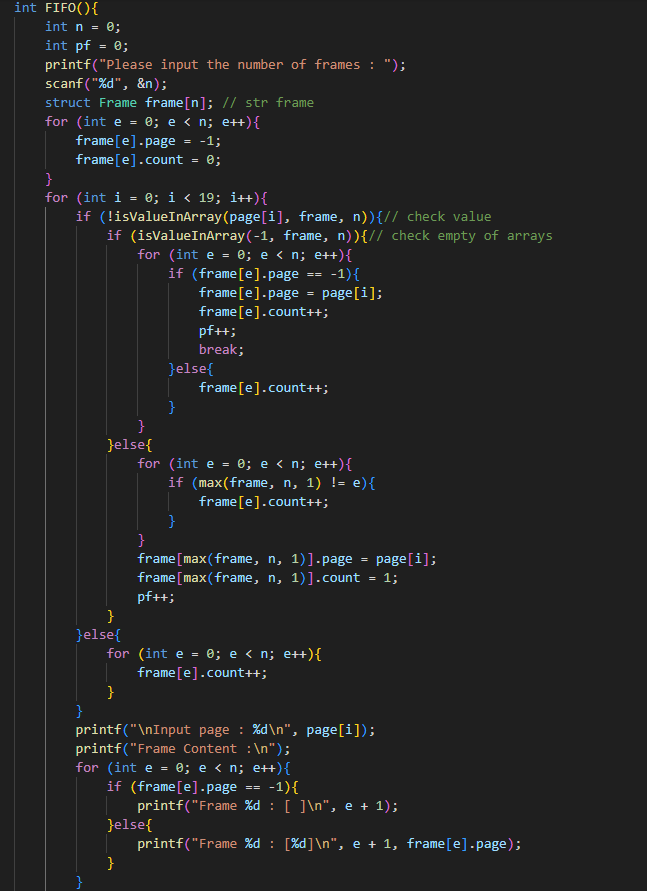
* โปรแกรมเข้าสู่ลูป while ซึ่งจะทำงานไปเรื่อยๆ จนกว่าผู้ใช้จะป้อนคีย์ 4 เพื่อออก.
* ภายในลูป โปรแกรมแสดงเมนูที่ให้ผู้ใช้เลือกอัลกอริทึมการแทนหน้าที่ต่างกัน (FIFO, LRU, OPT, และ ปิดโปรแกรม).
* ผู้ใช้ถูกแจ้งให้ป้อนคีย์และโปรแกรมใช้คำสั่ง switch เพื่อดำเนินการอัลกอริทึมที่ถูกเลือกตามค่าที่ผู้ใช้ป้อน.

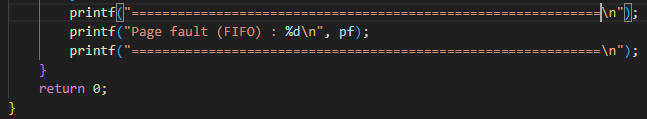
1. **การทำงานของอัลกอริทึม:**

* ถ้าผู้ใช้ป้อน 1 จะเรียกใช้ฟังก์ชัน FIFO()
* ถ้าผู้ใช้ป้อน 2 จะเรียกใช้ฟังก์ชัน LRU()
* ถ้าผู้ใช้ป้อน 3 จะเรียกใช้ฟังก์ชัน OPT()
* ถ้าผู้ใช้ป้อน 4 จะทำการหยุดการทำงานของโปรแกรม

1. **คำสั่ง Return:**

* ฟังก์ชัน main จะส่งค่า 0 เมื่อทำงานเสร็จสมบูรณ์.





1. **การรับค่าจำนวนเฟรม (frames) จากผู้ใช้:**

* โปรแกรมแสดงข้อความให้ผู้ใช้ป้อนจำนวนเฟรมที่ต้องการใช้.

1. **การสร้างและเตรียมข้อมูลเฟรม:**

* โปรแกรมสร้างอาร์เรย์ frame ที่มีขนาดเท่ากับจำนวนเฟรมที่ผู้ใช้ป้อน.
* แต่ละเฟรมมีสมาชิก page และ count โดยเริ่มต้น page ให้มีค่า -1 และ count ให้เป็น 0.

1. **การประมวลผลหน้าข้อมูล:**

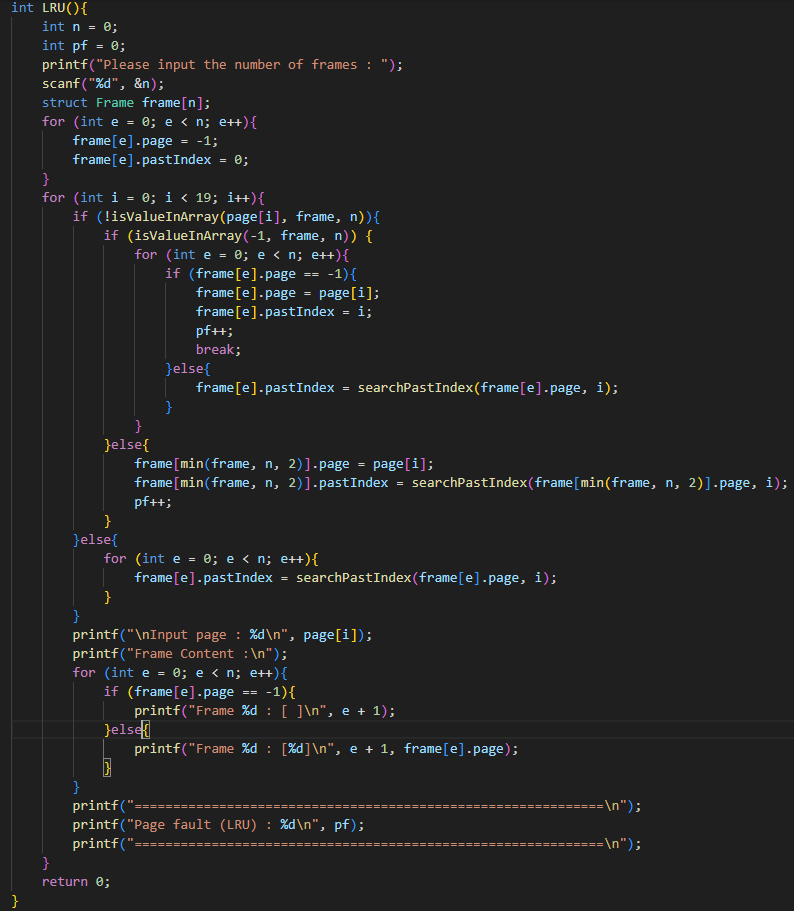
* โปรแกรมทำการตรวจสอบว่าหน้าข้อมูลปัจจุบันอยู่ในเฟรมหรือไม่โดยใช้ฟังก์ชัน isValueInArray.
* ถ้าหน้าข้อมูลยังไม่อยู่ในเฟรม (page fault) โปรแกรมทำการตรวจสอบว่ามีเฟรมที่ว่างหรือไม่.
* ถ้ามีเฟรมว่าง โปรแกรมใส่หน้าข้อมูลลงในเฟรมที่ว่าง.
* ถ้าไม่มีเฟรมว่าง โปรแกรมจะเลือกเฟรมที่มีการใช้น้อยที่สุดในการแทนที่ (FIFO) และใส่หน้าข้อมูลลงไป.
* ถ้าหน้าข้อมูลมีอยู่ในเฟรม (ไม่เกิด page fault) โปรแกรมจะเพิ่มค่า count ในทุกๆ เฟรม.

1. **การแสดงผลลัพธ์:**

* หลังจากการประมวลผลแต่ละหน้าข้อมูล โปรแกรมทำการแสดงผลลัพธ์ที่แสดงหน้าข้อมูลที่ถูกป้อนเข้ามา, เนื้อหาของเฟรม, และจำนวน page fault ที่เกิดขึ้น.

1. **คำสั่ง Return:**

* ฟังก์ชัน FIFO ส่งค่า 0 เมื่อทำงานเสร็จสมบูรณ์.



1. **การรับค่าจำนวนเฟรม (frames) จากผู้ใช้:**

* โปรแกรมแสดงข้อความให้ผู้ใช้ป้อนจำนวนเฟรมที่ต้องการใช้.

1. **การสร้างและเตรียมข้อมูลเฟรม:**

* โปรแกรมสร้างอาร์เรย์ frame ที่มีขนาดเท่ากับจำนวนเฟรมที่ผู้ใช้ป้อน.
* แต่ละเฟรมมีสมาชิก page และ pastIndex โดยเริ่มต้น page ให้มีค่า -1 และ pastIndex ให้เป็น 0.

1. **การประมวลผลหน้าข้อมูล:**

* โปรแกรมทำการตรวจสอบว่าหน้าข้อมูลปัจจุบันอยู่ในเฟรมหรือไม่โดยใช้ฟังก์ชัน isValueInArray.
* ถ้าหน้าข้อมูลยังไม่อยู่ในเฟรม (page fault) โปรแกรมทำการตรวจสอบว่ามีเฟรมที่ว่างหรือไม่.
* ถ้ามีเฟรมว่าง โปรแกรมใส่หน้าข้อมูลลงในเฟรมที่ว่างพร้อมกับการเซ็ต pastIndex ในเฟรมนั้นเป็นค่าปัจจุบัน.
* ถ้าไม่มีเฟรมว่าง โปรแกรมจะเลือกเฟรมที่มีค่า pastIndex น้อยที่สุดในการแทนที่ (LRU) และใส่หน้าข้อมูลลงไปพร้อมกับการเซ็ต pastIndex ในเฟรมใหม่เป็นค่าปัจจุบัน.
* ถ้าหน้าข้อมูลมีอยู่ในเฟรม (ไม่เกิด page fault) โปรแกรมจะอัพเดตค่า pastIndex ในทุกๆ เฟรม.

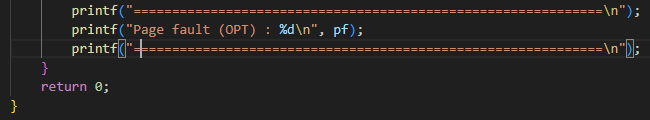
1. **การแสดงผลลัพธ์:**

* หลังจากการประมวลผลแต่ละหน้าข้อมูล โปรแกรมทำการแสดงผลลัพธ์ที่แสดงหน้าข้อมูลที่ถูกป้อนเข้ามา, เนื้อหาของเฟรม, และจำนวน page fault ที่เกิดขึ้น.

1. **คำสั่ง Return:**

* ฟังก์ชัน LRU ส่งค่า 0 เมื่อทำงานเสร็จสมบ





1. **การรับค่าจำนวนเฟรม (frames) จากผู้ใช้:**

* โปรแกรมแสดงข้อความให้ผู้ใช้ป้อนจำนวนเฟรมที่ต้องการใช้.

1. **การสร้างและเตรียมข้อมูลเฟรม:**

* โปรแกรมสร้างอาร์เรย์ frame ที่มีขนาดเท่ากับจำนวนเฟรมที่ผู้ใช้ป้อน.
* แต่ละเฟรมมีสมาชิก page และ nextIndex โดยเริ่มต้น page ให้มีค่า -1 และ nextIndex ให้เป็น 0.

1. **การประมวลผลหน้าข้อมูล:**

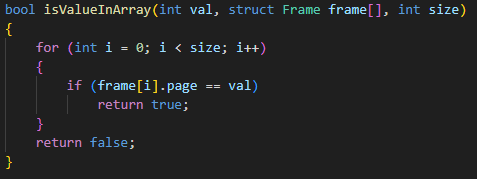
* โปรแกรมทำการตรวจสอบว่าหน้าข้อมูลปัจจุบันอยู่ในเฟรมหรือไม่โดยใช้ฟังก์ชัน isValueInArray.
* ถ้าหน้าข้อมูลยังไม่อยู่ในเฟรม (page fault) โปรแกรมทำการตรวจสอบว่ามีเฟรมที่ว่างหรือไม่.
* ถ้ามีเฟรมว่าง โปรแกรมใส่หน้าข้อมูลลงในเฟรมที่ว่างพร้อมกับการเซ็ต nextIndex ในเฟรมนั้นเป็นค่าที่บ่งบอกถึงการใช้งานต่อไปของหน้าข้อมูลนี้.
* ถ้าไม่มีเฟรมว่าง โปรแกรมจะเลือกเฟรมที่มีค่า nextIndex ที่มีการใช้งานล่าสุดน้อยที่สุด (Optimal) และใส่หน้าข้อมูลลงไปพร้อมกับการเซ็ต nextIndex ในเฟรมใหม่เป็นค่าที่บ่งบอกถึงการใช้งานต่อไปของหน้าข้อมูลนี้.
* ถ้าหน้าข้อมูลมีอยู่ในเฟรม (ไม่เกิด page fault) โปรแกรมจะอัพเดตค่า nextIndex ในทุกๆ เฟรม.

1. **การแสดงผลลัพธ์:**

* หลังจากการประมวลผลแต่ละหน้าข้อมูล โปรแกรมทำการแสดงผลลัพธ์ที่แสดงหน้าข้อมูลที่ถูกป้อนเข้ามา, เนื้อหาของเฟรม, และจำนวน page fault ที่เกิดขึ้น.

1. **คำสั่ง Return:**

* ฟังก์ชัน OPT ส่งค่า 0 เมื่อทำงานเสร็จสมบูรณ์.



1. **การรับค่า:**

* รับค่า val ที่ต้องการตรวจสอบว่ามีอยู่ในอาร์เรย์หรือไม่.
* รับอาร์เรย์ของโครงสร้าง Frame ที่ชื่อ frame.
* รับขนาดของอาร์เรย์ size.

1. **การทำงาน:**

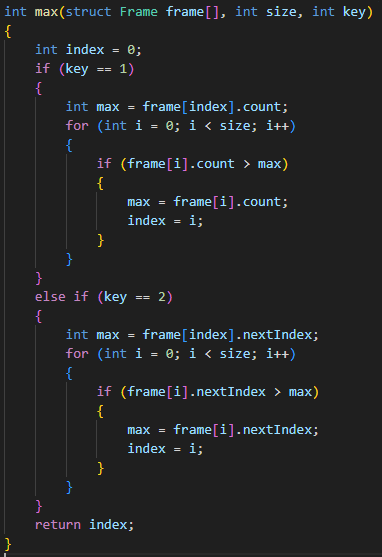
* ใช้ลูป for เพื่อวนลูปทุกรายการในอาร์เรย์ frame.

1. **การเปรียบเทียบค่า:**

* ที่แต่ละรอบของลูป for ฟังก์ชันตรวจสอบว่า frame[i].page เท่ากับค่า val หรือไม่.
* ถ้าเท่ากัน (ค่า val อยู่ในอาร์เรย์) ฟังก์ชันจะคืนค่า true.

1. **คืนค่า:**

* ถ้าไม่เจอค่า val ในทุกรายการในอาร์เรย์ frame ฟังก์ชันจะคืนค่า false.



1. **การรับค่า:**

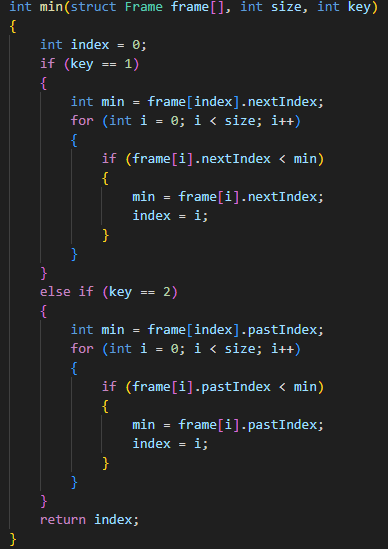
* รับอาร์เรย์ของโครงสร้าง Frame ที่ชื่อ frame.
* รับขนาดของอาร์เรย์ size.
* รับค่า key เพื่อระบุว่าจะใช้ค่า count หรือ nextIndex ในการค้นหา.

1. **การคำนวณ:**

* ตั้งค่าตัวแปร index เป็น 0 เพื่อเก็บดัชนีของค่าสูงสุด.
* ตั้งค่าตัวแปร max ในขอบเขตของลูป เพื่อให้สามารถเปรียบเทียบค่าได้.
* ในลูป for ทำการเปรียบเทียบค่าของ count หรือ nextIndex ของแต่ละเฟรม.
* หากค่าใดมีค่ามากกว่าค่า max ที่เก็บไว้, ก็ทำการอัพเดตค่า max และ index เป็นดัชนีของเฟรมนั้น.

1. **คืนค่า:**

* หลังจากลูปเสร็จสิ้น, ฟังก์ชันจะคืนค่า index ที่เก็บดัชนีของเฟรมที่มีค่าสูงสุด.



1. **การรับค่า:**

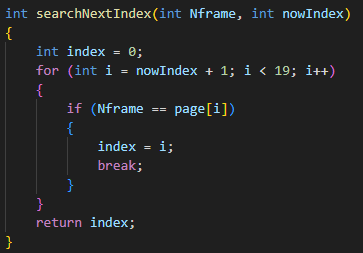
* รับอาร์เรย์ของโครงสร้าง Frame ที่ชื่อ frame.
* รับขนาดของอาร์เรย์ size.
* รับค่า key เพื่อระบุว่าจะใช้ค่า nextIndex หรือ pastIndex ในการค้นหา.

1. **การคำนวณ:**

* ตั้งค่าตัวแปร index เป็น 0 เพื่อเก็บดัชนีของค่าน้อยที่สุด.
* ตั้งค่าตัวแปร min ในขอบเขตของลูป เพื่อให้สามารถเปรียบเทียบค่าได้.
* ในลูป for ทำการเปรียบเทียบค่าของ nextIndex หรือ pastIndex ของแต่ละเฟรม.
* หากค่าใดมีค่าน้อยกว่าค่า min ที่เก็บไว้, ก็ทำการอัพเดตค่า min และ index เป็นดัชนีของเฟรมนั้น.

1. **คืนค่า:**

* หลังจากลูปเสร็จสิ้น, ฟังก์ชันจะคืนค่า index ที่เก็บดัชนีของเฟรมที่มีค่าน้อยที่สุด.



1. **การรับค่า:**

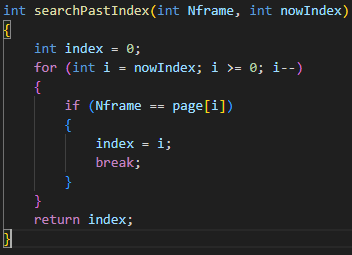
* รับค่า Nframe ซึ่งเป็นหน้าข้อมูลที่ต้องการค้นหาดัชนีของ.
* รับค่า nowIndex ซึ่งเป็นดัชนีปัจจุบันที่ต้องการเริ่มการค้นหา.

1. **การค้นหา:**

* ใช้ลูป for เพื่อวนลูปตั้งแต่ดัชนี nowIndex + 1 ไปจนถึงสุดท้ายของอาร์เรย์ page (จำนวนรอบที่เหลือเท่ากับ 19 - nowIndex - 1).
* ในแต่ละรอบของลูป, ทำการเปรียบเทียบค่าของ Nframe กับ page[i].
* หากค่าที่เรียกค้นหา (Nframe) ตรงกับค่าในอาร์เรย์ที่พบที่ดัชนี i, ก็ทำการกำหนดค่า index เป็น i และจบการทำงานของลูป.

1. **คืนค่า:**

* ฟังก์ชันคืนค่า index ที่เก็บดัชนีของหน้าข้อมูล Nframe ในอาร์เรย์ page ที่พบครั้งแรกหลังจาก nowIndex.



1. **การรับค่า:**

* รับค่า Nframe ซึ่งเป็นหน้าข้อมูลที่ต้องการค้นหาดัชนีของ.
* รับค่า nowIndex ซึ่งเป็นดัชนีปัจจุบันที่ต้องการเริ่มการค้นหา.

1. **การค้นหา:**

* ใช้ลูป for เพื่อวนลูปตั้งแต่ดัชนี nowIndex ไปยัง 0.
* ในแต่ละรอบของลูป, ทำการเปรียบเทียบค่าของ Nframe กับ page[i].
* หากค่าที่เรียกค้นหา (Nframe) ตรงกับค่าในอาร์เรย์ที่พบที่ดัชนี i, ก็ทำการกำหนดค่า index เป็น i และจบการทำงานของลูป.

1. **คืนค่า:**

* ฟังก์ชันคืนค่า index ที่เก็บดัชนีของหน้าข้อมูล Nframe ในอาร์เรย์ page ที่พบครั้งแรกก่อน nowIndex.

**สรุปผลการทดลอง**

เมื่อโปรแกรมทำงานจะให้เรากรอกค่าจำนวนเลขที่เราต้องการที่จะกรอก

และทำการกรอกเลข frame เพื่อนทำการประมวลผล และเรื่องระบบการทำงานว่าจะให้ทำงานแบบไหน