

**โครงงานฝึกทักษะการเขียนโปรแกรม ครั้งที่ 2**

**โปรแกรมสำหรับจัดเก็บข้อมูลบัตรนักศึกษา**

**ด้วยเครื่องมือ Double Linked List**

**นายจิรเมธ พั่วพันธ์**

**รหัสนิสิต 60160157**

**เสนอ**

**ผู้ช่วยศาสตราจารย์นวลศรี เด่นวัฒนา**

**อาจารย์พจน์สพร แซ่ลิ้ม**

**โครงงานนี้เป็นส่วนหนึ่งของ**

**รายวิชา 88823359 โครงสร้างข้อมูลและขั้นตอนวิธีสำหรับวิศวกรรมซอฟต์แวร์**

**ภาคการศึกษาที่ 1 ปีการศึกษา 2561 สาขาวิชาวิศวกรรมซอฟต์แวร์**

**คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา**

# สารบัญ

เรื่อง หน้า

[สารบัญ ก](#_Toc527055881)

[สารบัญรูปภาพ ข](#_Toc527055882)

[สารบัญตาราง ค](#_Toc527055883)

[ส่วนที่ 1 การวิเคราะห์โจทย์ปัญหา 4](#_Toc527055884)

[1.1 โจทย์ปัญหา 4](#_Toc527055885)

[1.2 วิเคราะห์แนวทางแก้ไขปัญหา 4](#_Toc527055886)

[1.3 การออกแบบ 4](#_Toc527055887)

[1.4 รหัสเทียม 8](#_Toc527055888)

[ส่วนที่ 2 รายละเอียดของผลลัพธ์ 14](#_Toc527055889)

[2.1 หน้าจอระบบ 14](#_Toc527055890)

[ส่วนที่ 3 การวิเคราะห์และประเมินผลลัพธ์ 17](#_Toc527055891)

[3.1 ประสิทธิภาพของโปรแกรม 17](#_Toc527055892)

[3.2 สรุปผลลัพธ์ 24](#_Toc527055893)

# สารบัญรูปภาพ

ภาพที่ หน้า

[ภาพที่ 1‑1 แผนภาพคลาสของโครงงาน 5](#_Toc527056795)

[ภาพที่ 2‑1 ส่วนการแสดงผลของเมนูระบบ 14](#_Toc527056796)

[ภาพที่ 2‑2 การเพิ่มข้อมูลนักศึกษา 15](#_Toc527056797)

[ภาพที่ 2‑3 การค้นหาข้อมูลนักศึกษา 15](#_Toc527056798)

[ภาพที่ 2‑4 การยืนยันว่าต้องการแก้ไขข้อมูลหรือไม่ 15](#_Toc527056799)

[ภาพที่ 2‑5 ส่วนการแก้ไขข้อมูลนักศึกษา 16](#_Toc527056800)

[ภาพที่ 2‑6 ส่วนการลบข้อมูลนักศึกษา 16](#_Toc527056801)

[ภาพที่ 2‑7 ส่วนการแสดงข้อมูลนักศึกษาทั้งหมด 16](#_Toc527056802)

[ภาพที่ 3‑1 โปรแกรมส่วนของมอดูลการแสดงผลเมนูระบบ 17](#_Toc527056803)

[ภาพที่ 3‑2 โปรแกรมส่วนของมอดูลค้นหาข้อมูล 18](#_Toc527056804)

[ภาพที่ 3‑3 โปรแกรมส่วนของมอดูลแสดงข้อมูล 19](#_Toc527056805)

[ภาพที่ 3‑4 โปรแกรมส่วนของมอดูลแก้ไขข้อมูล 20](#_Toc527056806)

[ภาพที่ 3‑5 โปรแกรมส่วนของมอดูลอ่านไฟล์ 21](#_Toc527056807)

[ภาพที่ 3‑6 โปรแกรมส่วนของมอดูลบันทึกไฟล์ 22](#_Toc527056808)

[ภาพที่ 3‑7 โปรแกรมส่วนของมอดูลลบข้อมูล 23](#_Toc527056809)

# สารบัญตาราง

ตารางที่ หน้า

[ตารางที่ 1‑1 แสดงส่วนประกอบของคลาส DBLinkedList 5](#_Toc527056810)

[ตารางที่ 1‑2 แสดงส่วนประกอบของคลาส DBLinkedList (ต่อ) 6](#_Toc527056811)

[ตารางที่ 1‑3 แสดงส่วนประกอบของคลาส node 6](#_Toc527056812)

[ตารางที่ 1‑4 แสดงส่วนประกอบของคลาส Student 6](#_Toc527056813)

[ตารางที่ 1‑5 แสดงส่วนประกอบของคลาส Student (ต่อ) 7](#_Toc527056814)

[ตารางที่ 3‑1 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลการแสดงผลของเมนูระบบ 17](#_Toc527056815)

[ตารางที่ 3‑2 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลค้นหาข้อมูล 18](#_Toc527056816)

[ตารางที่ 3‑3 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลแสดงข้อมูล 19](#_Toc527056817)

[ตารางที่ 3‑4 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลแก้ไขข้อมูล 20](#_Toc527056818)

[ตารางที่ 3‑5 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลอ่านไฟล์ 21](#_Toc527056819)

[ตารางที่ 3‑6 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลบันทึกไฟล์ 22](#_Toc527056820)

[ตารางที่ 3‑7 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลลบข้อมูล 23](#_Toc527056821)

# การวิเคราะห์โจทย์ปัญหา

## โจทย์ปัญหา

โปรแกรมสำหรับจัดเก็บข้อมูลบัตรนักศึกษา ด้วยเครื่องมือ Double Linked List

**ความสามารถของโปรแกรม**

1. ผู้ใช้งานสามารถเพิ่มหรือจัดเก็บข้อมูลส่วนตัวของนักศึกษาแต่ละคน โดยมีข้อมูลที่ต้องจัดเก็บ ดังนี้

1.1 รหัสนักศึกษา

1.2 คำนำหน้า

1.3 ชื่อ – นามสกุล

1.4 วันที่ออกบัตร

1.5 วันที่หมดอายุบัตร

1.6 คณะที่กำลังศึกษา

หมายเหตุ : ในส่วนการจัดเก็บข้อมูลนักศึกษานี้ ต้องดำเนินการแบบเชิงวัตถุ (Class) เท่านั้น

2. ผู้ใช้งานสามารถค้นหาข้อมูลบัตรของนักศึกษาได้

3. ผู้ใช้งานสามารถแก้ไขข้อมูลบัตรของนักศึกษาได้

4. ผู้ใช้งานสามารถลบข้อมูลบัตรของนักศึกษาได้

5. ผู้ใช้งานสามารถเรียกดูข้อมูลของบัตรนักศึกษาทั้งหมดที่มีอยู่ได้

6. กรณีออกจากโปรแกรมทำการจัดเก็บข้อมูลบัตรนักศึกษาลงไฟล์ std\_card.txt

7. กรณีเริ่มต้นโปรแกรมทำการดึงข้อมูลบัตรนักศึกษาที่เคยจัดเก็บจากไฟล์ std\_card.txt

## วิเคราะห์แนวทางแก้ไขปัญหา

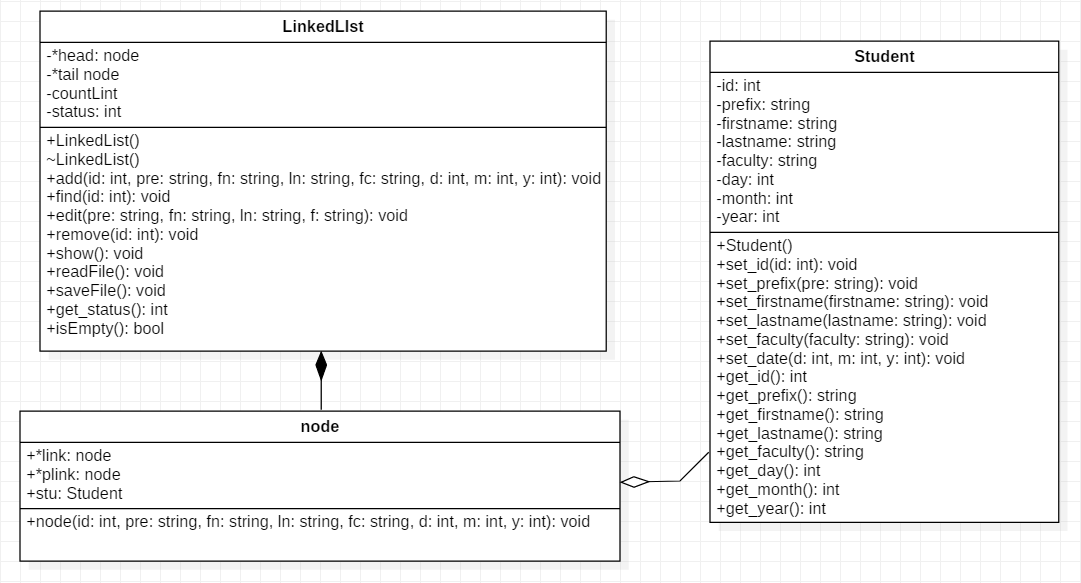
จากโจทย์วิเคราะห์ได้ว่าควรจะมีการการสร้างคลาสจำนวน 3 คลาส ในคลาสที่หนึ่ง เป็นคลาส Studentใช้จัดเก็บข้อมูลส่วนตัวของนักศึกษา ควรมี Attribute (ตัวแปร) ที่เป็นชนิด String เพื่อเก็บข้อมูลที่เป็นอักขระ และชนิด Int เพื่อเก็บข้อมูลที่เป็นจำนวนเต็ม ในคลาสที่สอง เป็นคลาส DBLinkedList ใช้เป็นโครงสร้างของข้อมูลเก็บไว้ใน node เป็นลำดับ โดยทำหน้าที่สำหรับเก็บเพิ่ม ลบ และการค้นหาข้อมูล และในคลาสที่สาม เป็นคลาส node จะประกอบไปด้วย ข้อมูลของคลาส Student และpointer 2ตัว ใช้เป็นตัวชี้ไปยัง node ถัดไปและอีกหนึ่งตัวชี้ไปยังnodeก่อนหน้า

## การออกแบบ

จากการวิเคราะห์นำมาสร้างแผนภาพ ทำให้ได้แผนภาพคลาส ดังหัวข้อที่ 1.3.1

### แผนภาพคลาส (Class Diagram)

แผนภาพคลาสแสดงได้ดังภาพที่ 1-1 แผนภาพคลาสของโครงงาน รวมถึงการแสดงค่าต่างๆ ในคลาสดังตารางที่ 1-1 ตารางที่ 1-2 ตารางที่ 1-3 และตารางที่ 1-4 แสดงส่วนประกอบของคลาส LinkedList Student และ node



ภาพที่ 1‑1 แผนภาพคลาสของโครงงาน

ตารางที่ 1‑1 แสดงส่วนประกอบของคลาส DBLinkedList

|  |  |
| --- | --- |
| ข้อมูล | คำอธิบาย |
| Attribute | |
| \* head : node | ตัวชี้สำหรับชี้ที่อยู่ของข้อมูลลำดับแรก |
| \* tail : node | ตัวชี้สำหรับชี้ที่อยู่ของข้อมูลลำดับท้าย |
| status : int | ตัวแปรสำหรับการเช็คเงื่อนไงใน main |
| int : count | ตัวแปรใช้นับจำนวนข้อมูลที่เก็บไว้ทั้งหมด |
| Method | |
| LinkedList () | ค่า Constructor |
| ~ LinkedList () | ค่า Deconstructor |
| add(id:int,pre:string,fn:string,ln:string,  fc:string,d:int,m:int,y:int) : void | เป็น Method สำหรับเพิ่มข้อมูล |
| find(id:int) : void | เป็น Method สำหรับค้นหาข้อมูล |
| edit(pre:string,fn:string,ln:string,f:string) : void | เป็น Method สำหรับแก้ไขข้อมูล |
| show() : void | เป็น Method สำหรับแสดงข้อมูล |
| get\_status() : int | เป็น Method สำหรับคืนค่าไปยัง main |
| isEmpty () : bool | เป็น Method สำหรับเช็คค่าว่าง |
| remove(id:int) : void | เป็น Method สำหรับลบข้อมูล |

ตารางที่ 1‑2 แสดงส่วนประกอบของคลาส DBLinkedList (ต่อ)

|  |  |
| --- | --- |
| ข้อมูล | คำอธิบาย |
| readFile() : void | เป็น Method สำหรับอ่านข้อมูลบนไฟล์ |
| saveFile() : void | เป็น Method สำหรับเขียนข้อมูลลงบนไฟล์ |

ตารางที่ 1‑3 แสดงส่วนประกอบของคลาส node

|  |  |
| --- | --- |
| ข้อมูล | คำอธิบาย |
| Attribute | |
| stu : Student | Object สำหรับเก็บข้อมูลนักศึกษา |
| node \* link; | ตัวชี้เก็บข้อมูลที่อยู่ของ node ถัดจากนี้ |
| node \* plink; | ตัวชี้เก็บข้อมูลที่อยู่ของ node ก่อนหน้านี้ |
| Method | |
| node(id:int,pre:string,fn:string,ln:string,  fc:string,d:int,m:int,y:int) | ค่า Constructor |

ตารางที่ 1‑4 แสดงส่วนประกอบของคลาส Student

|  |  |
| --- | --- |
| ข้อมูล | คำอธิบาย |
| Attribute | |
| Id : int | ตัวแปรสำหรับการเก็บข้อมูลรหัสนักศึกษา |
| prefix : string | ตัวแปรสำหรับการเก็บข้อมูลคำนำหน้าชื่อ |
| firstname : string | ตัวแปรสำหรับการเก็บข้อมูลชื่อจริง |
| lastname : string | ตัวแปรสำหรับการเก็บข้อมูลนามสกุล |
| Faculty : string | ตัวแปรสำหรับการเก็บข้อมูลคณะ |
| day : int | ตัวแปรสำหรับการเก็บข้อมูลวันที่ออกบัตร |
| month : int | ตัวแปรสำหรับการเก็บข้อมูลเดือนที่ออกบัตร |
| year : int | ตัวแปรสำหรับการเก็บข้อมูลปีที่ออกบัตร |
| Method | |
| Student() | ค่า Constructor |
| set\_id(id:int) : void | เป็น Method สำหรับกำหนดค่าให้ตัวแปร id |

ตารางที่ 1‑5 แสดงส่วนประกอบของคลาส Student (ต่อ)

|  |  |
| --- | --- |
| ข้อมูล | คำอธิบาย |
| set\_prefix(pre:string) : void | เป็น Method สำหรับกำหนดค่าให้ตัวแปร prefix |
| set\_firstname(firstname:string) : void | เป็น Method สำหรับกำหนดค่าให้ตัวแปร firstname |
| set\_lastname(lastname:string) : void | เป็น Method สำหรับกำหนดค่าให้ตัวแปร lastname |
| set\_faculty(faculty:string) : void | เป็น Method สำหรับกำหนดค่าให้ตัวแปร faculty |
| set\_date(d:int,m:int,y:int) : void | เป็น Method สำหรับกำหนดวันที่ออกบัตร |
| get\_id() : int | เป็น Method สำหรับคืนค่าตัวแปร id |
| get\_prefix() : string | เป็น Method สำหรับคืนค่าตัวแปร prefix |
| get\_firstname() : string | เป็น Method สำหรับคืนค่าตัวแปร firstname |
| get\_lastname() : string | เป็น Method สำหรับคืนค่าตัวแปร lastname |
| get\_faculty() : string | เป็น Method สำหรับคืนค่าตัวแปร faculty |
| get\_day(): int | เป็น Method สำหรับคืนค่าตัวแปร day |
| get\_month(): int | เป็น Method สำหรับคืนค่าตัวแปร month |
| get\_year(): int | เป็น Method สำหรับคืนค่าตัวแปร year |

## รหัสเทียม

รหัสเทียมจะแสดงเขียนแยกตาม Method หรือ Module ของโปรแกรม

### มอดูลการแสดงข้อมูลเมนู

|  |  |
| --- | --- |
| อัลกอริทึม | มอดูลการแสดงเมนูของโปรแกรม |
| ข้อมูลนำเข้า | ไม่มี |
| ผลลัพธ์ | แสดงเมนู |
| 1. ประกาศตัวแปรชื่อ choice | |
| 1. ลูป | |
| * 1. แสดงเมนู | |
| * 1. เลือกเมนูในช่วงที่กำหนด   2. ถ้าเลือกเมนูในช่วงที่กำหนด จบลูป   3. ถ้าไม่ กลับไปที่ 2.1 | |
| 1. ทำงานตามเมนูที่ระบุ 2. จบการทำงาน  มอดูการเพิ่มข้อมูลนักศึกษา อัลกอริทึม มอดูลการเพิ่มข้อมูลนักศึกษา  ข้อมูลนำเข้า รับค่ารหัสนิสิต คำนำหน้าชื่อ ชื่อ-นามสกุล คณะ และวันที่จากแป้นพิมพ์  ผลลัพธ์ จัดเก็บข้อมูลไว้ในระบบ  1. ประกาศตัวแปรชื่อ id, day, month, year เป็นชนิดจำนวนเต็ม  2. ประกาศตัวแปรชื่อ prefix, firstname , lastname, faculty เป็นชนิดอักขระ  3. กำหนดค่าให้ตัวแปรผ่านแป้นพิมพ์  3.1 กำหนดค่าให้ตัวแปร id  3.2 กำหนดค่าให้ตัวแปร prefix  3.3 กำหนดค่าให้ตัวแปร firstname  3.4 กำหนดค่าให้ตัวแปร lastname  3.5 กำหนดค่าให้ตัวแปร faculty  3.6 กำหนดค่าให้ตัวแปร day  3.7 กำหนดค่าให้ตัวแปร month  3.8 กำหนดค่าให้ตัวแปร year  4. เรียกใช้ Method add จากคลาส LinkedList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์  5. การทำงานภายใน Method add  5.1 ประกาศตัวแปรชื่อ newNode ให้ชี้ไปยังที่อยู่ใหม่ของคลาส node  5.2 เข้าเงื่อนไข และเรียกใช้ Method isEmpty เพื่อเช็คจำนวน count ว่างว่างหรือไม่ | |
| ให้ตัวแปร  5.2.1 กำหนดให้ตัวแปร head เก็บค่าที่อยู่ของ newNode  5.2.2 กำหนดให้ตัวแปร tail เก็บค่าที่อยู่ของ newNode  5.3 เข้าเงื่อนไข else  5.3.1 กำหนดให้ตัวแปร newNode ให้ชี้ไปยังที่อยู่ plink เก็บค่าที่อยู่ของ tail  5.3.2 กำหนดให้ตัวแปร tail ให้ชี้ไปยังที่อยู่ link เก็บค่าที่อยู่ของ newNode  5.3.3 กำหนดให้ตัวแปร tail เก็บค่าที่อยู่ของ newNode  6. เพิ่มค่าของตัวแปร count ขึ้น 1 จำนวน  7. จบการทำงาน มอดูลการค้นหาข้อมูลนักศึกษา | |
| อัลกอริทึม มอดูลการค้นหาข้อมูลนักศึกษา  ข้อมูลนำเข้า รับค่ารหัสนิสิตจากแป้นพิมพ์  ผลลัพธ์ แสดงข้อมูลนิสิตที่ค้นหา  1. ประกาศตัวแปรชื่อ id เป็นชนิดจำนวนเต็ม  2. กำหนดค่าให้ตัวแปร id ผ่านแป้นพิมพ์  3. เรียกใช้ Method find จากคลาส LinkedList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์  4. การทำงานภายใน Method find  4.1 ประกาศตัวแปรชื่อ check เป็นชนิดจำนวนเต็ม และมีค่าเป็น -1  4.2ประกาศตัวแปรชื่อ i เป็นชนิดจำนวนเต็ม และมีค่าเป็น 1  4.3 กำหนดให้ตัวแปร status มีค่าเป็น -1  4.2 เข้าลูป วน LinkedList  4.2.1 เข้าเงื่อนไข ถ้า id เท่ากับ ตัวแปร temp ชี้ไเรียกใช้ Method get\_id ของ Obiect stu จาก  คลาส Student  4.2.1.1 กำหนดค่า check ให้เท่ากับค่า i  4.2.1.2 แสดงข้อมูลส่วนตัวของนักศึกษาทั้งหมด โดยให้ temp เรียกใช้ Method get ของ  Object stu จากคลาส Student  4.3 เข้าเงื่อนไข ถ้า check เท่ากับ -1  4.3.1 แสดงข้อความผ่านทางหน้าจอว่า “-> No Data!!!”  5. จบการทำงาน | |
|  | |

### มอดูลการแก้ไขข้อมูลนักศึกษา

อัลกอริทึม มอดูลการแก้ไขข้อมูลนักศึกษา

ข้อมูลนำเข้า รับค่ารหัสนิสิต คำนำหน้าชื่อ ชื่อจริง นามสกุล และ คณะจากแป้นพิมพ์

ผลลัพธ์ จัดเก็บข้อมูลนิสิตที่แก้ไข

1. ประกาศตัวแปรชื่อ id และ status เป็นชนิดจำนวนเต็ม

2. กำหนดค่าให้ตัวแปร id ผ่านแป้นพิมพ์

3. เรียกใช้ Method find จากคลาส LinkedList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์

4. การทำงานภายใน Method find

4.1 ประกาศตัวแปรชื่อ check เป็นชนิดจำนวนเต็ม และมีค่าเป็น -1

4.2ประกาศตัวแปรชื่อ i เป็นชนิดจำนวนเต็ม และมีค่าเป็น 1

4.3 กำหนดให้ตัวแปร status มีค่าเป็น -1

4.2 เข้าลูป วน LinkedList

4.2.1 เข้าเงื่อนไข ถ้า id เท่ากับ ตัวแปร temp ชี้ไปเรียกใช้ Method get\_id ของ Obiect stu จาก

คลาส Student

4.2.1.1 กำหนดค่า check ให้เท่ากับค่า i

4.2.1.2 แสดงข้อมูลส่วนตัวของนักศึกษาทั้งหมด โดยให้ temp ไปเรียกใช้ Method get ของ

Object stu จากคลาส Student

4.3 เข้าเงื่อนไข ถ้า check เท่ากับ -1

4.3.1 แสดงคำว่า “-> No Data!!!” ออกทางหน้าจอ

4.4 เข้าเงื่อนไข ถ้า check ไม่เท่ากับ -1

4.4.2 แสดงข้อความผ่านทางหน้าจอว่า "Do you want to edit the information? [Y|N]: "

4.4.3 ประกาศตัวแปร yes เป็นชนิดอักขระ

4.4.4 กำหนดค่าให้ตัวแปร yes ผ่านแป้นพิมพ์

4.4.5 เข้าเงื่อนไข ถ้า yes มีค่าเท่ากับ “Y”

4.4.5.1 กำหนดค่าให้ status มีค่าเท่ากับ check

5. เข้าสู่การทำงานของ main

6. เข้าเงื่อนไข ถ้า -1 ไม่เท่ากับ Method get\_status จากคลาส LinkedList

6.1 ประกาศตัวแปรชื่อ prefix, firstname , lastname, faculty เป็นชนิดอักขระ

6.1.1 กำหนดค่าให้ตัวแปร prefix

6.1.2 กำหนดค่าให้ตัวแปร firstname

6.1.3 กำหนดค่าให้ตัวแปร lastname

6.1.4 กำหนดค่าให้ตัวแปร faculty

7. เรียกใช้ Method edit จากคลาส LinkedList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์

8. การทำงานภายใน Method edit

8.1 สร้างตัวแปร temp ชี้ไปที่อยู่ของ head มีชนิดเป็นคลาส node

8.2 เข้าลูป เงื่อนไข “วนจนกว่าค่า i จะมากกว่า status”

8.2.1 กำหนดให้ temp มีค่าเท่ากับ temp ชี้ไปยังที่อยู่ของ link

8.1 ให้ temp เรียกใช้ Method set\_prefix ของ Object stu จากคลาส Student โดยมีพารามิเตอร์

เป็น prefix

8.2 ให้ temp เรียกใช้ Method set\_firstname ของ Object stu จากคลาส Student โดยมีพารามิเตอร์

เป็น firstname

8.3 ให้ temp เรียกใช้ Method set\_lastname ของ Object stu จากคลาส Student โดยมีพารามิเตอร์

เป็น lastname

8.4 ให้ temp เรียกใช้ Method set\_faculty ของ Object stu จากคลาส Student โดยมีพารามิเตอร์

เป็น faculty

8.5 กำหนดค่า status ให้มีค่าเท่ากับ -1

8.6 แสดงข้อความผ่านทางหน้าจอ "-> Edit Complete!!!"

9. จบการทำงาน

### มอดูลการแสดงข้อมูลนักศึกษาทั้งหมด

อัลกอริทึม มอดูลการแสดงข้อมูลนักศึกษา

ข้อมูลนำเข้า ไม่มี

ผลลัพธ์ แสดงข้อมูลส่วนตัวนักศึกษา

1. เรียกใช้ Method show จากคลาส LinkedList

2. การทำงานภายใน Method show

2.1 เข้าลูป LinkedList

2.1.1 ให้ temp เรียกใช้ Method get\_id ของ Object stu และแสดงออกจอ

2.1.2 ให้ temp เรียกใช้ Method get\_prefix ของ Object stu และแสดงออกจอ

2.1.3 ให้ temp เรียกใช้ Method get\_firstname ของ Object stu และแสดงออกจอ

2.1.4 ให้ temp เรียกใช้ Method get\_lastname ของ Object stu และแสดงออกจอ

2.1.5 ให้ temp เรียกใช้ Method get\_faculty ของ Object stu และแสดงออกจอ

2.1.6 ให้ temp เรียกใช้ Method get\_day ของ Object stu และแสดงออกจอ

2.1.7 ให้ temp เรียกใช้ Method get\_month ของ Object stu และแสดงออกจอ

2.1.8 ให้ temp เรียกใช้ Method get\_year ของ Object stu และแสดงออกจอ

3. จบการทำงาน

### มอดูลการลบข้อมูลนักศึกษาทั้งหมด

อัลกอริทึม มอดูลการลบข้อมูลนักศึกษา

ข้อมูลนำเข้า รับค่ารหัสนิสิตจากแป้นพิมพ์

ผลลัพธ์ ลบข้อมูลส่วนตัวนักศึกษา

1. ประกาศตัวแปรชื่อ id เป็นชนิดจำนวนเต็ม

2. กำหนดค่าให้ตัวแปร id ผ่านแป้นพิมพ์

3. เรียกใช้ Method remove จากคลาส LinkedList โดยมีพารามิเตอร์ เป็นตัวแปรที่รับผ่านแป้นพิมพ์

4. การทำงานภายใน Method remove

4.1 ประกาศตัวแปรชื่อ round เป็นชนิดจำนวนเต็ม และมีค่าเป็น 0

4.2 ประกาศตัวแปรชื่อ check เป็นชนิดจำนวนเต็ม และมีค่าเป็น 0

4.3 เข้าลูป LinkedList

4.4 เช็คเงื่อนไข round เท่ากับ 0

4.4.1 เช็คเงื่อนไข temp เรียกใช้ Method get\_id ของ Object stu เท่ากับ id

4.4.1.1 เช็คเงื่อนไข temp เท่ากับ head และ temp เท่ากับ tail

• กำหนดให้ head มีค่าเป็น NULL

• กำหนดให้ tail มีค่าเป็น NULL

4.4.1.2 เช็คเงื่อนไข temp เท่ากับ tail

• กำหนดให้ head มีค่าเป็น head ชี้ไปที่ link

• กำหนดให้ head ชี้ไปที่ plink มีค่าเป็น NULL

4.4.1.3 ลบ temp

4.4.1.4 ลดจำนวน count ลงอีก 1

4.4.1.5 กำหนดให้ check เท่ากับ 2

4.4.1.6 หยุดการทำงานของลูป

4.5 เช็คเงื่อนไข round ไม่เท่ากับ 0

4.5.1 เช็คเงื่อนไข temp เรียกใช้ Method get\_id ของ Object stu เท่ากับ id

4.5.1.1 เช็คเงื่อนไข temp เท่ากับ tail

• กำหนดให้ tail มีค่าเป็น tail ชี้ไปที่ plink

• กำหนดให้ tail ชี้ไปที่ link มีค่าเป็น NULL

4.5.1.2 เข้า else

• กำหนดให้ temp ชี้ไปที่ plink ชี้ไปที่ link มีค่าเป็น temp ชี้ไปที่ link

• กำหนดให้ temp ชี้ไปที่ link ชี้ไปที่ plink มีค่าเป็น temp ชี้ไปที่ plink

4.5.1.3 ลบ temp

4.5.1.4 ลดจำนวน count ลงอีก 1

4.5.1.5 กำหนดให้ check เท่ากับ 2

4.5.1.6 หยุดการทำงานของลูป

4.5.2 เข้า else

4.5.2.1 กำหนดให้ check เท่ากับ 1

4.6 เพิ่มจำนวน round ขึ้นอีก 1

4.7 จบลูป LinkedList

4.8 เข้าเงื่อนไข check เท่ากับ 1

4.8.1 แสดงข้อความผ่านทางหน้าจอ “-> No Data!!!”

4.9 เข้า else

4.8.1 แสดงข้อความผ่านทางหน้าจอ "-> Remove ID “<< id << " Complete...”

5. จบการทำงาน

# รายละเอียดของผลลัพธ์

## หน้าจอระบบ

### การแสดงผลของเมนูระบบ

เมื่อเริ่มเปิดโปรแกรม โปรแกรมจะแสดงข้อมูลเมนูระบบ ดังภาพที่ 2 – 1 โดย การใช้งานคือ ให้ระบุ เลขตัวเลือก 0 1 2 3 และ 4 โดย แต่ละ ตัวเลขที่การทำงานที่แตกต่างการ ดังนี้

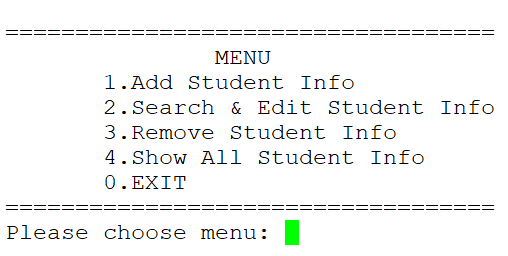
0. Exit คือ ออกจากโปรแกรม

1. Add Student Info คือ การเพิ่มข้อมูลนักศึกษา

2. Search & Edit Student Info คือ การค้นหาและแก้ไขข้อมูลนักศึกษา

3. Remove Student Info คือ การลบข้อมูลนักศึกษา

4. Show All Student Info คือ การแสดงข้อมูลนักศึกษาที่มีทั้งหมด



ภาพที่ 2‑1 ส่วนการแสดงผลของเมนูระบบ

### การเพิ่มข้อมูลนักศึกษา

เมื่อเข้าสู่เมนูที่ 1 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 2 โดย การใช้งานคือ ให้ระบุข้อมูลดังนี้

1. รหัสนิสิต

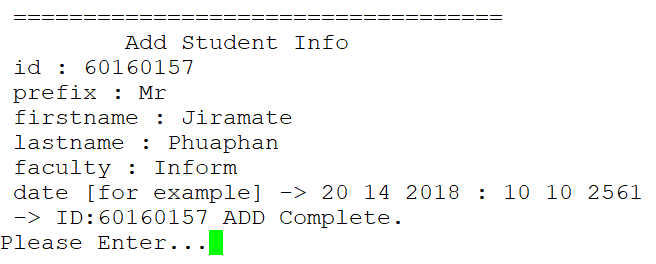
2. คำนำหน้า

3. ชื่อจริง

4. นามสกุล

5. คณะ

6. วันที่ปัจจุบัน

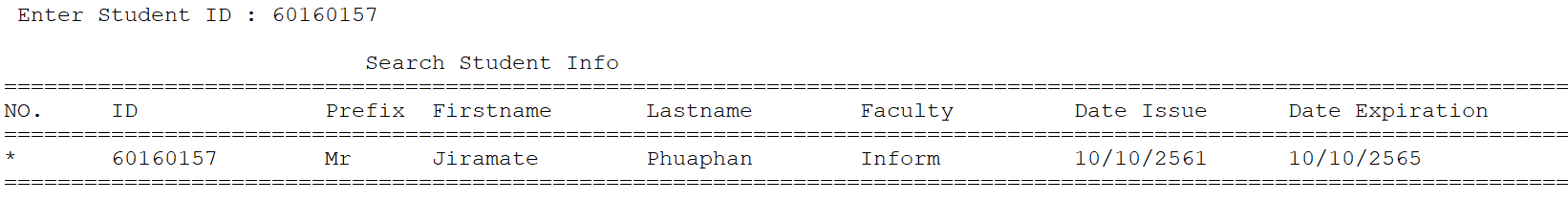


ภาพที่ 2‑2 การเพิ่มข้อมูลนักศึกษา

### การค้นหาข้อมูลนักศึกษา

เมื่อเข้าสู่เมนูที่ 2 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 3 โดย การใช้งานคือ ให้ระบุข้อมูลรหัสนักศึกษาที่มีอยู่ในระบบ เพื่อทำการค้นหา

หากค้นหาไม่เจอข้อมูลรหัสนักศึกษา โปรแกรมจะกลับไปที่หน้าเมนู ดังภาพที่ 2 - 1



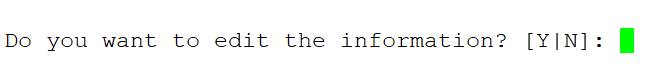
ภาพที่ 2‑3 การค้นหาข้อมูลนักศึกษา

### การแก้ไขข้อมูลนักศึกษา

เมื่อได้ผลลัพธ์ ดังภาพที่ 2 – 3 จะมีข้อความให้ระบุ เพื่อยืนยันว่าต้องการแก้ไข ดังภาพที่ 2 – 4 โดย วิธีใช้งานคือ ให้ระบุข้อมูลที่เป็นตัวอักษรดังนี้

Y คือ ยืนยันเพื่อไปหน้าแก้ไขข้อมูล

N คือ กลับหน้าเมนู



ภาพที่ 2‑4 การยืนยันว่าต้องการแก้ไขข้อมูลหรือไม่

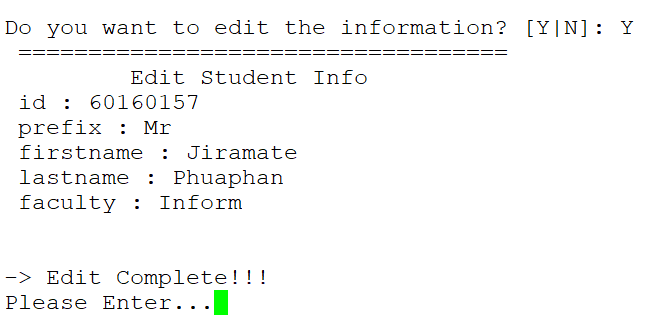
เมื่อทำการระบุ Y จะได้ผลลัพธ์ ดังภาพที่ 2 – 5 โดย การใช้งานคือ ให้แก้ไขข้อมูลดังนี้

1. คำนำหน้า

2. ชื่อจริง

3. นามสกุล

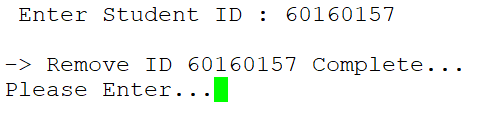
4. คณะ



ภาพที่ 2‑5 ส่วนการแก้ไขข้อมูลนักศึกษา

### การลบข้อมูลนักศึกษา

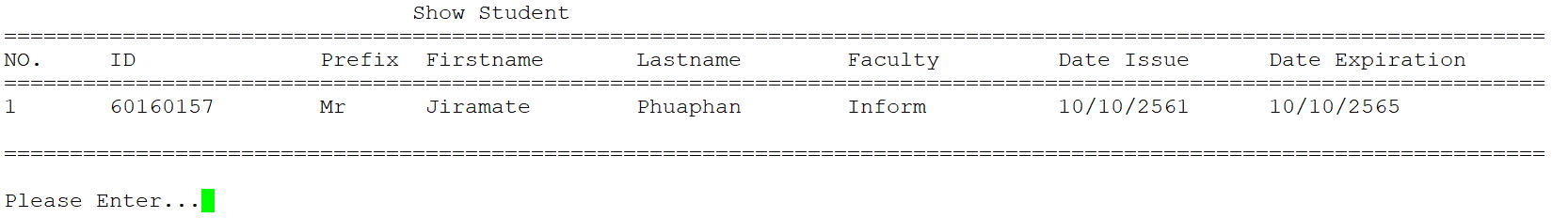
เมื่อเข้าสู่เมนูที่ 4 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 6 การใช้งานคือ ให้ระบุข้อมูลรหัสนักศึกษาที่มีอยู่ในระบบ เพื่อทำการลบข้อมูล



ภาพที่ 2‑6 ส่วนการลบข้อมูลนักศึกษา

### การแสดงข้อมูลนักศึกษาทั้งหมด

เมื่อเข้าสู่เมนูที่ 3 โปรแกรมจะแสดงผล ดังภาพที่ 2 – 7



ภาพที่ 2‑7 ส่วนการแสดงข้อมูลนักศึกษาทั้งหมด

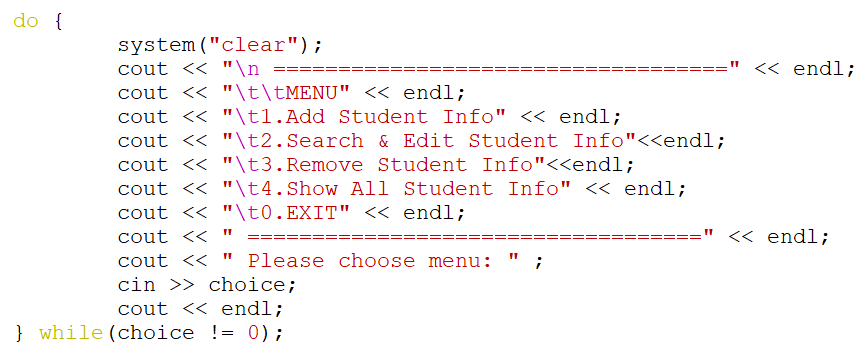
# การวิเคราะห์และประเมินผลลัพธ์

## ประสิทธิภาพของโปรแกรม

เป็นส่วนการวิเคราะห์ข้อมูล ที่ได้จากผลลัพธ์ของโปรแกรม ซึ่งประกอบไปด้วยมอดูล การแสดงผลเมนูของระบบ การค้นหาข้อมูล และการแสดงข้อมูล โดยมีการวิเคราะห์ในแต่ละส่วนได้ดังนี้

### มอดูลการแสดงผลของเมนูระบบ

ในส่วนมอดูลการแสดงผลของเมนูระบบมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3-1 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 1



ภาพที่ 3‑1 โปรแกรมส่วนของมอดูลการแสดงผลเมนูระบบ

ตารางที่ 3‑1 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลการแสดงผลของเมนูระบบ

|  |  |  |
| --- | --- | --- |
| ค่า choice | ตรวจสอบเงื่อนไข choice != 0 | แสดงผลข้อมูลภายใน loop |
| 1 | / | / |
| 2 | / | / |
| 3 | / | / |
| 0 | / | X |
| จำนวนครั้งที่ทำ | 4 | 3 |
| ถ้า n ครั้ง | n | n -1 |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n) + (n – 1) = 2n – 1

= BigO(n)

### มอดูลค้นหาข้อมูล

ในส่วนมอดูลการค้นหาข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 2 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 2



ภาพที่ 3‑2 โปรแกรมส่วนของมอดูลค้นหาข้อมูล

กำหนดให้ค่า count เท่ากับ 5 (วนทั้งหมด 5 ครั้ง ถึงจะเจอค่า NULL)

ตารางที่ 3‑2 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลค้นหาข้อมูล

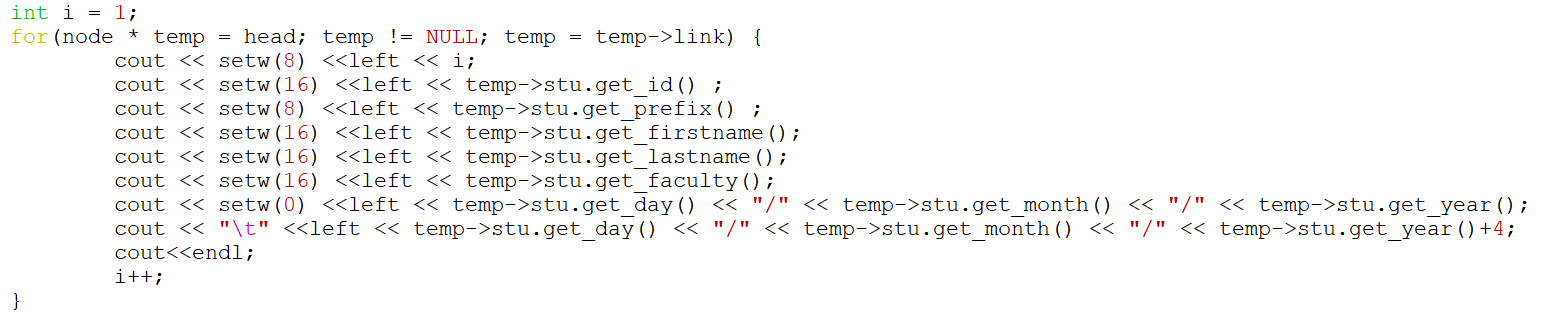
|  |  |  |
| --- | --- | --- |
| ค่า temp | ตรวจสอบเงื่อนไข temp != NULL | แสดงผลข้อมูลภายใน loop |
| 0x...1 | / | / |
| 0x…2 | / | / |
| 0x...3 | / | / |
| 0x...4 | / | / |
| 0x...5 | / | / |
| NULL | / | X |
| จำนวนครั้งที่ทำ | 6 | 5 |
| ถ้า n ครั้ง | n + 1 | n |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n + 1) + (n) = 2n + 1

= BigO(n)

### มอดูลแสดงข้อมูล

ในส่วนมอดูลการแสดงข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 3 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 3



ภาพที่ 3‑3 โปรแกรมส่วนของมอดูลแสดงข้อมูล

กำหนดให้ค่า count เท่ากับ 5 (วนทั้งหมด 5 ครั้ง ถึงจะเจอค่า NULL)

ตารางที่ 3‑3 แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลแสดงข้อมูล

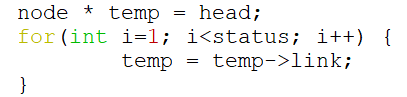
|  |  |  |
| --- | --- | --- |
| ค่า temp | ตรวจสอบเงื่อนไข temp != NULL | แสดงผลข้อมูลภายใน loop |
| 0x...1 | / | / |
| 0x…2 | / | / |
| 0x...3 | / | / |
| 0x...4 | / | / |
| 0x...5 | / | / |
| NULL | / | X |
| จำนวนครั้งที่ทำ | 6 | 5 |
| ถ้า n ครั้ง | n + 1 | n |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n + 1) + (n) = 2n + 1

= BigO(n)

### มอดูลแก้ไขข้อมูล

ในส่วนมอดูลการแสดงข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 4 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 4



ภาพที่ ‑ โปรแกรมส่วนของมอดูลแก้ไขข้อมูล

กำหนดให้ค่า status เท่ากับ 5

ตารางที่ ‑ แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลแก้ไขข้อมูล

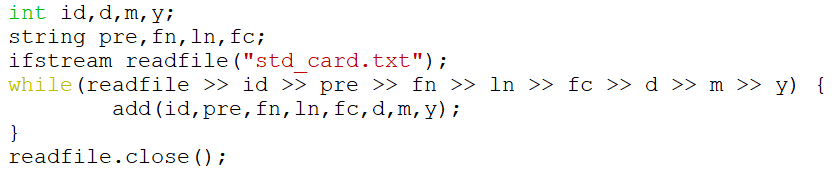
|  |  |  |
| --- | --- | --- |
| ค่า i | ตรวจสอบเงื่อนไข I < status | แสดงผลข้อมูลภายใน loop |
| 1 | / | / |
| 2 | / | / |
| 3 | / | / |
| 4 | / | / |
| 5 | / | X |
| จำนวนครั้งที่ทำ | 5 | 4 |
| ถ้า n ครั้ง | n | n - 1 |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n) + (n – 1) = 2n – 1

= BigO(n)

### มอดูลอ่านไฟล์

ในส่วนมอดูลการแสดงข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 5 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 5



ภาพที่ ‑ โปรแกรมส่วนของมอดูลอ่านไฟล์

กำหนดให้ใน std\_card.txt มีข้อความจำนวน 5 บรรทัด

ตารางที่ ‑ แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลอ่านไฟล์

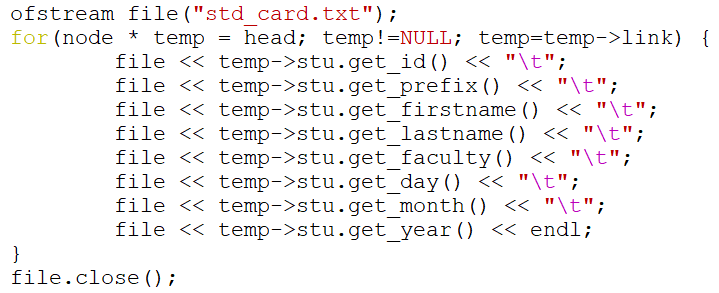
|  |  |  |
| --- | --- | --- |
| readfile | ตรวจสอบเงื่อนไข | แสดงผลข้อมูลภายใน loop |
| อ่านบรรทัดที่ 1 | / | / |
| อ่านบรรทัดที่ 2 | / | / |
| อ่านบรรทัดที่ 3 | / | / |
| อ่านบรรทัดที่ 4 | / | / |
| อ่านบรรทัดที่ 5 | / | / |
| NULL | / | X |
| จำนวนครั้งที่ทำ | 6 |  |
| ถ้า n ครั้ง | n + 1 | n |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n + 1) + (n) = 2n + 1

= BigO(n)

### มอดูลบันทึกไฟล์

ในส่วนมอดูลการแสดงข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 6 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 6



ภาพที่ ‑ โปรแกรมส่วนของมอดูลบันทึกไฟล์

กำหนดให้ค่า count เท่ากับ 5 (วนทั้งหมด 5 ครั้ง ถึงจะเจอค่า NULL)

ตารางที่ ‑ แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลบันทึกไฟล์

|  |  |  |
| --- | --- | --- |
| temp | ตรวจสอบเงื่อนไข temp != NULL | แสดงผลข้อมูลภายใน loop |
| 0x...1 | / | / |
| 0x…2 | / | / |
| 0x...3 | / | / |
| 0x...4 | / | / |
| 0x...5 | / | / |
| NULL | / | X |
| จำนวนครั้งที่ทำ | 6 | 5 |
| ถ้า n ครั้ง | n + 1 | n |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n + 1) + (n) = 2n + 1

= BigO(n)

### มอดูลลบข้อมูล

ในส่วนมอดูลการแสดงข้อมูลมีลักษณะการทำงานตามโปรแกรมที่แสดงอยู่ดังภาพที่ 3 - 7 และทำการวิเคราะห์ได้ดังตารางที่ 3 – 7



ภาพที่ ‑ โปรแกรมส่วนของมอดูลลบข้อมูล

กำหนดให้ค่า count เท่ากับ 5 (วนทั้งหมด 5 ครั้ง ถึงจะเจอค่า NULL)

ตารางที่ ‑ แสดงการวิเคราะห์ประสิทธิภาพของการทำงานมอดูลลบข้อมูล

|  |  |  |
| --- | --- | --- |
| temp | ตรวจสอบเงื่อนไข temp != NULL | แสดงผลข้อมูลภายใน loop |
| 0x...1 | / | / |
| 0x…2 | / | / |
| 0x...3 | / | / |
| 0x...4 | / | / |
| 0x...5 | / | / |
| NULL | / | X |
| จำนวนครั้งที่ทำ | 6 | 5 |
| ถ้า n ครั้ง | n + 1 | n |

สรุปได้ว่า Big-O ของการทำงานมอดูลนี้คือ (n + 1) + (n) = 2n + 1

= BigO(n)

## สรุปผลลัพธ์

จากการวิเคราะห์การทำงานของโปรแกรม พบว่ามอดูลทุกมอดูลมีประสิทธิภาพเท่ากันในโปรแกรม

เนื่องจาก โปรแกรมมีการใช้ loop เพื่อวนเช็คเงื่อนไขหรือแสดงผลเท่านั้น