โครงงานการจัดทำโปรแกรม
“เกมเรียงตัวเลข”
(Puzzle number)

นายนำโชค สิงหะชัย
รหัสนิสิต 60160169 กลุ่ม 2

เสนอ
ผู้ช่วยศาสตราจารย์นวลศรี เด่นวัฒนา
อาจารย์ณัฐพร ภักดี
อาจารย์มาโนชญ์ ใจกว้าง
อาจารย์พจน์สพร แซ่ลิ้ม

รายวิชา 88814159 และ 88814259
หลักและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์
ภาคการศึกษาที่ 1 ปีการศึกษา 2560 สาขาวิชาวิศวกรรมซอฟต์แวร์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

บทคัดย่อ

ชื่อเรื่อง	เกมเรียงตัวเลข (Puzzle number)
ชื่อผู้จัดทำ	นายนำโชค สิงหะชัย
สาขาวิชา	วิศวกรรมซอฟต์แวร์
คณะ	วิทยาการสนเทศ
	มหาวิทยาลัยบูรพา
ปีการศึกษา	2560

โครงงานนี้จัดทำขึ้นเพื่อเป็นคะแนนส่วนหนึ่งในรายวิชา 88814159 และ 88814259 เรื่องการทำโครงงานและอีกส่วนหนึ่งเพื่อเป็นการพัฒนาโปรแกรม เกมเรียงตัวเลข (Puzzle number) โดยการศึกษาจากภาษาซี ในโปรแกรมสามารถเลือกระดับความยากได้ และโปรแกรมก็จะทำการสุ่มค่าเลขมาให้เพื่อให้ ผู้เล่นได้เรียง คะแนนจะถูกคำนวณเป็นเวลาและจำนวนของการสไลด์ เป็นต้น

ซึ่งผู้จัดทำโรงงานนี้ได้ออกแบบระบบของโปรแกรมที่นำไปใช้จริงได้ เพื่อให้สะดวกต่อการใช้งานจึงได้แบ่งการทำงานเป็นหมวดต่างๆ ได้แก่ แบ่งรายการการทำงาน มีการเลือกระดับความยาก ป้อนชื่อ สามารถดูคะแนน หรือ เวลาที่เข้ามาเล่นได้ เป็นต้น

กิตติกรรมประกาศ

โครงงานเกมเรียงตัวเลข (Puzzle number) เล่มนี้ สำเร็จรุร่วงไปด้วยความกรุณาให้ความช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์นวลศรี เด่นวัฒนา และคณะอาจารย์ผู้สอนในรายวิชา 88814159 และ 88814259 วิชาหลักและวิธีการทางโปรแกรมสำหรับวิศวกรรรมซอฟต์แวร์ โดยได้ให้แนวคิดและวิธีการต่างๆ รวมถึงคำแนะนำ และคำปรึกษาตลอดจนการแก้ไขข้อบกพร่องของโปรแกรมต่างๆ บอกทุกขั้นตอนการจัดทำโครงงาน ทางผู้จัดทำขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ขอขอบคุณครอบครัว พี่และเพื่อนในสาขาวิชาวิศวกรรมซอฟต์แวร์ที่คอยช่วยให้คำแนะนำ และ เป็นกำลังใจ ตลอดจนการแก้ไขข้อบกพร่องของโปรแกรมและการจัดทำโครงงานเล่มนี้ จนทำให้โครงงานเล่มนี้ประสบความสำเร็จ ทางผู้จัดทำโครงงานเล่มนี้ขอขอบพระคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

นำโชค สิงหะชัย

สารบัญ

หน้า

บทคัดย่อ.....ก

กิตติกรรมประกาศ.....ข

สารบัญ.....ค

สารบัญรูปภาพ.....ฉ

สารบัญตาราง.....ช

บทที่

1.บทนำ.....1

1.1. ที่มาของโครงงาน.....1

1.2. วัตถุประสงค์ของโครงงาน.....1

1. เพื่อพัฒนาระบบเกมเรียงตัวเลข (Puzzle number).....1

2. เพื่อผ่อนคลายจากการเรียน และการทำงาน.....1

3. เพื่อฝึกทักษะการจัดเรียงตัวเลขให้กับนิสิต สาขาวิชาวิศวกรรมซอฟต์แวร์.....1

1.3. ขอบเขตของโครงงาน.....1

1. เมื่อเลือกเมนูที่ 1.เล่น.....1

2. เมื่อเลือกเมนูที่ 2.วิธีการเล่น.....2

3. เมื่อเลือกเมนูที่ 3.ตารางคะแนน.....2

4. เมื่อเลือกเมนูที่ 4.เครดิต.....2

5. เมื่อเลือกเมนูที่ 5.ออกโปรแกรม.....2

1.4. เครื่องมือที่ใช้ในการพัฒนา.....2

1. ทรัพยากรด้านฮาร์ดแวร์.....2

2. ทรัพยากรด้านซอฟต์แวร์.....2

3. ทรัพยากรด้านเครื่องมือภาษาซี.....3

1.5. ขั้นตอนในการดำเนินงาน.....4

สารบัญ (ต่อ)

บทที่ หน้า

1.6. ประโยชน์ที่คาดว่าจะได้รับ 4

2.ทฤษฎีและโครงงานที่เกี่ยวข้อง 5

2.1. หลักการและวิธีใช้ภาษาซี 5

1. ประวัติภาษาซี 5

2. โครงสร้างของโปรแกรมภาษาซี 6

3. ตัวแปรกับชนิดของข้อมูล 6

4. การแสดงผลและการรับข้อมูล (Printf and Scanf) 7

5. ตัวนำเดินการทางคณิตศาตร์ 9

6. คำสั่งเงื่อนไข 9

7. คำสั่งทำซ้ำ 9

8. ฟังก์ชัน (Function) 10

9. ตัวแปรแถวลำดับ (Array) 11

10. การโปรแกรมกับแฟ้มข้อมูล (File) 11

11. ตัวชี้ (Pointer) 12

12. การกำหนดชนิดโครงสร้าง (structure) 12

13. สตริง (String or Array of charactor) 12

2.2. ปริศนา (Puzzle) 13

1. Puzzle มีประโยชน์อย่างไร 13

3.วิธีการดำเนินโครงงาน 14

3.1. วิเคราะห์และออกแบบความสามารถของระบบ 14

1. ความสามารถของระบบสำหรับผู้ดูแลระบบ 14

2. ความสามารถของระบบสำหรับผู้เล่น 14

3.2. รหัสเทียมของโปรแกรม 14

สารบัญ (ต่อ)

บทที่ หน้า

4.ผลการดำเนินงานโครงงาน 31

4.1. การทำงานของโปรแกรม 31

2. เมนู วิธีการเล่น 34

3. เมนู ตารางคะแนน 34

4. เมนู เครดิตผู้จัดทำ 35

5. สิ้นสุดการทำงาน 35

5.สรุปและวิเคราะห์ผลการดำเนินการ 36

5.1. ประโยชน์ที่ได้รับ 36

5.2. ปัญหาและอุปสรรคในการจัดทำโครงงาน 36

5.3. ข้อจำกัดของโปรแกรม 36

บรรณานุกรม 37

ภาคผนวก ก 39

สารบัญรูปภาพ

ภาพที่ หน้า

2-1 การเรียกใช้ PRINTF 7

2-2 รหัสควบคุมรูปแบบ 8

2-3 การเรียกใช้ SCANF 8

2-4 ฟังก์ชัน (FUNCTION) 10

4-1 หน้าแรก 31

4-2 เมนู PLAY 31

4-3 เมนู ใส่ชื่อผู้เล่น 32

4-4 เมนู READY? 32

4-5 แสดงเกมระดับง่าย 32

4-6 แสดงเกมระดับปานกลาง 33

4-7 แสดงเกมระดับยาก 33

4-8 เมื่อชนะเกม 34

4-9 เมนู วิธีการเล่น 34

4-10 เมนู ตารางคะแนน 34

4-11 เมนู เครดิตผู้จัดทำ 35

4-12 สิ้นสุดการทำงาน 35

สารบัญตาราง

ตารางที่ หน้า

1-1 กำหนดการในการจัดทำโครงงาน 4

บทที่ 1

บทนำ

1.1 ที่มาของโครงงาน

เนื่องจากผู้จัดทำโครงงานต้องการที่จะพัฒนาโปรแกรมเกมเรียงตัวเลข (Puzzle number) ซึ่งพัฒนามาจากภาษาซี เพื่อฝึกทักษะการเขียนโปรแกรมและฝึกการจัดเรียงตัวเลขสำหรับนิสิตสาขาวิชาวิศวกรรมซอฟต์แวร์ เพื่อผ่อนคลายละว่างการพักจากการอ่านอ่านหนังสือ หรือทำการบ้าน ซึ่งโปรแกรมเรียงตัวเลขได้สามารถเลือกระดับความยาก ได้ 3 ระดับ เพื่อความสนุกสนานเพิ่มขึ้น โดยสามารถกำหนดคะแนนโดยเวลาและจำนวนครั้งการสไลด์

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อพัฒนาระบบเกมเรียงตัวเลข (Puzzle number)
2. เพื่อผ่อนคลายจากการเรียน และการทำงาน
3. เพื่อฝึกทักษะการจัดเรียงตัวเลขให้กับนิสิต สาขาวิชาวิศวกรรมซอฟต์แวร์

1.3 ขอบเขตของโครงงาน

เมื่อทำการรันโปรแกรมเกมเรียงตัวเลข ก็จะแสดงส่วนของเมนูขึ้นมา ดังนี้

1. เมื่อเลือกเมนูที่ 1.เล่น
    1.1. เมื่อเข้ามาถึงจะมีเมนูให้เลือกระดับความยาก 3 ระดับ คือ 1.ง่าย 2.ปกติ 3.ยาก และกด x เพื่อออกสู่เมนูหลัก
    1.2. เมื่อเลือกระดับ ง่าย จะให้ทำการใส่ชื่อผู้เล่น เมื่อใส่เสร็จจะถามว่าคุณพร้อมหรือไม่ และจะแสดงตารางตัวเลข 3 คูณ 3 และแสดงจำนวณครั้งของการสไลด์ เมื่อทำงานสไลด์ตัวเลขเรียงได้ถูกต้องแล้ว ให้กด Enter แล้วโปรแกรมก็จะแสดงเวลาทั้งหมดที่ใช้ไปและจำนวนครั้งของการสไลด์ กด x เพื่อออกสู่เมนูเลือกระดับความยาก
    1.3. เมื่อเลือกระดับ ปกติ จะให้ทำการใส่ชื่อผู้เล่น เมื่อใส่เสร็จจะถามว่าคุณพร้อมหรือไม่ และ จะแสดงตารางตัวเลข 4 คูณ 4 และแสดงจำนวณครั้งของการสไลด์ เมื่อทำงานสไลด์ตัวเลขเรียงได้ถูกต้องแล้ว ให้กด Enter แล้วโปรแกรมก็จะแสดงเวลาทั้งหมดที่ใช้ไปแล จำนวนครั้งของการสไลด์ กด x เพื่อออกสู่เมนูเลือกระดับความยาก

1.4. เมื่อเลือกระดับ ยาก จะให้ทำการใส่ชื่อผู้เล่น เมื่อใส่เสร็จจะถามว่าคุณพร้อมหรือไม่ และ จะแสดงตารางตัวเลข 5 คูณ 5 และแสดงจำนวณครั้งของการสไลด์ เมื่อทำงานสไลด์ตัวเลขเรียงได้ถูกต้องแล้ว ให้กด Enter แล้วโปรแกรมก็จะแสดงเวลาทั้งหมดที่ใช้ไปและจำนวนครั้งของการสไลด์ กด x เพื่อออกสู่เมนู เลือกระดับความยาก

2. เมื่อเลือกเมนูที่ 2.วิธีการเล่น
   2.1. โปรแกรมก็จะแสดงวิธีการเล่น และกด x เพื่อออกสู่เมนูหลัก
3. เมื่อเลือกเมนูที่ 3.ตารางคะแนน
   3.1. โปรแกรมจะแสดง ระดับความยาก ชื่อผู้เล่น จำนวนครั้งการสไลด์ จำนวนเวลาที่เล่น และวันที่เล่น ตามลำดับการเข้ามาเล่น และกด x เพื่อออกสู่เมนูหลัก
4. เมื่อเลือกเมนูที่ 4.เครดิต
   4.1. โปรแกรมจะแสดง รายละเอียดของผู้จัดทำโครงงาน เช่น เบอร์โทร,อีเมลติดต่อ, เฟสบุ๊ค เป็นต้น และกด x เพื่อออกสู่เมนูหลัก
5. เมื่อเลือกเมนูที่ 5.ออกโปรแกรม

1.4 เครื่องมือที่ใช้ในการพัฒนา

ในการที่จัดทำทำโครงงานพัฒนาเกมเรียงตัวเลข (Puzzle number) ต้องใช้ทรัพยากรดังนี้

1. ทรัพยากรด้านฮาร์ดแวร์
   - Notebook ยี่ห้อ Dell
   - CPU Intet CORE i5 @ 1.70GHz.
   - Memory 4GB DDR3.
   - HDD 500 GBs.
   - Intel® HD GRAPHICS 4000
2. ทรัพยากรด้านซอฟต์แวร์
   2.1. DekdeeBurapha Linux
      - Burapha Linux
      - VI Editor
      - GCC (GUN Compiler Collection)
   2.2. Windows
      - Windows 10 64 bit

- Putty (SSH and telnet client)
- Winscp (graphical SFTP client)
- Dev – C++ version 5.11

3. ทรัพยากรด้านเครื่องมือภาษาซี
    3.1. ข้อความสั่งแสดงผลทางจอภพ (Printf)
    3.2. ข้อความสั่งรับค่าทางแป้นพิมพ์ (Scanf)
    3.3. ตัวนำเดินการทางคณิตศาตร์
    3.4. ข้อความสั่งแบบมีเงื่อนไข (if / else)
    3.5. ข้อความสั่งแบบมีเงื่อนไข (switch case)
    3.6. ข้อความสั่งซ้ำโดยตรวจสอบเงื่อนไข (for)
    3.7. ข้อความสั่งซ้ำโดยตรวจสอบเงื่อนไข (while)
    3.8. ข้อความสั่งซ้ำโดยตรวจสอบเงื่อนไข (do...while)
    3.9. การใช้ ฟังก์ชัน (Function)
    3.10. การใช้ ตัวแปรแถวลำดับ (Array)
    3.11. การใช้ แฟ้มอักขระข้อมูล (File)
    3.12. การใช้ สตริง (String or Array of charactor)
    3.13. การใช้ library time.h
    3.14. การใช้ library stdlib.h
    3.15. การใช้ library string.h
    3.16. การใช้ library unistd.h
    3.17. การใช้ library termios.h

1.5 ขั้นตอนในการดำเนินงาน

ตารางที่ 1-1 กำหนดการในการจัดทำโครงงาน

ลำดับ	แผนปฏิบัติงานโครงงาน								
	กิจกรรม	ตุลาคม				พฤษจิกายน			
		15 – 31				1 - 30			
1.	ศึกษาความเป็นไปได้ที่จะพัฒนาระบบ	x	x						
2.	ศึกษาเครื่องมือ			x	x				
3.	ออกแบบระบบ				x	x	x		
4	ลงมือพัฒนา					x	x	x	
5	ทดสอบระบบ								x

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เพิ่มทักษะการจัดเรียงเลขให้กับนิสิต สาขาวิชาวิศวกรรมซอฟต์แวร์
2. ได้พัฒนาโปรแกรมเกมเรียงตัวเลข (Puzzle number)

บทที่ 2

ทฤษฎีและโครงงานที่เกี่ยวข้อง

การจัดทำโครงงานเกมเรียงตัวเลข (Puzzle number) ครั้งนี้ ผู้จัดทำได้ศึกษาหาความรู้ทางด้าน การเขียนโปรแกรมโดยใช้ภาษาซีในการดำเนินการ ทั้งในชั้นเรียน นอกชั้นเรียน และยังมีการศึกษาเกี่ยวกับ ปริศนา (Puzzle) เพื่อเป็นแนวทางการพัฒนา จากเว็บไซต์ต่างๆ ซึ่งมีเนื้อหาดังนี้

- หลักการและวิธีใช้ภาษา ซี
- ปริศนา (Puzzle)

2.1 หลักการและวิธีใช้ภาษาซี

1. ประวัติภาษาซี

ภาษาซีเป็นภาษาที่ถือว่าเป็นทั้งภาษาระดับสูงและระดับต่ำ ถูกพัฒนาโดยเดนนิส ริดชี (Dennis Ritche) แห่งห้องทดลองเบลล์ (Bell Laboratories) ที่เมอร์รีฮิล มลรัฐนิวเจอร์ซี่ โดยเดนนิสได้ใช้หลักการของภาษา บีซีพีแอล (BCPL : Basic Combine Programming Language) ซึ่งพัฒนาขึ้นโดยเคน ทอมสัน (Ken Tomson) การออกแบบและพัฒนาภาษาซีของเดนนิส ริดชี มีจุดมุ่งหมายให้เป็นภาษาสำหรับใช้เขียนโปรแกรมปฏิบัติการระบบยูนิกซ์ และได้ตั้งชื่อว่า ซี (C) เพราะเห็นว่า ซี (C) เป็นตัวอักษรต่อจากบี (B) ของภาษา BCPL ภาษาซีถือว่าเป็นภาษาระดับสูงและภาษาระดับต่ำ ทั้งนี้เพราะ ภาษาซีมีวิธีใช้ข้อมูลและมีโครงสร้างการควบคุมการทำงานของโปรแกรมเป็นอย่างเดียวกับภาษาของโปรแกรมระดับสูงอื่นๆ จึงถือว่าเป็นภาษาระดับสูง ในด้านที่ถือว่าภาษาซีเป็นภาษาระดับต่ำ เพราะภาษาซีมีวิธีการเข้าถึงในระดับต่ำที่สุดของฮาร์ดแวร์ ความสามารถทั้งสองด้านของภาษานี้เป็นสิ่งที่เกื้อหนุนซึ่งกันและกัน ความสามารถระดับต่ำทำให้ภาษาซีสามารถใช้เฉพาะเครื่องได้ และความสามารถระดับสูง ทำให้ภาษาซีเป็นอิสระจากฮาร์ดแวร์ ภาษาซีสามารถสร้างรหัสภาษาเครื่องซึ่งตรงกับชนิดของข้อมูลนั้นได้เอง ทำให้โปรแกรมที่เขียนด้วยภาษาซีที่เขียนบนเครื่องหนึ่ง สามารถนำไปใช้กับอีกเครื่องหนึ่งได้ ประกอบกับการใช้พอยน์เตอร์ในภาษาซี นับได้ว่าเป็นตัวอย่างที่ดีของการเป็นอิสระจากฮาร์ดแวร์

2. โครงสร้างของโปรแกรมภาษาซี

โปรแกรมในภาษาซีทุกโปรแกรมจะประกอบด้วยฟังก์ชันอย่างน้อย หนึ่งฟังก์ชัน คือ ฟังก์ชัน main โดยโปรแกรมภาษาซีจะเริ่มทำงานที่ฟังก์ชัน main ก่อน ในแต่ละฟังก์ชันจะประกอบด้วย

2.1. Function Header ประกอบด้วยชื่อฟังก์ชัน และอาจมีรายการของ argument (บางคนเรียก parameter) อยู่ในวงเล็บ

2.2. Variable Declaration ส่วนประกาศตัวแปร สำหรับภาษาซี ตัวแปรหรือค่าคงที่ทุกตัวที่ใช้ในโปรแกรมจะต้องมีการประกาศก่อนว่าจะใช้งานอย่างไร จะเก็บค่าในรูปแบบใด เช่น interger หรือ real number

2.3. Compound Statements ส่วนของประโยคคำสั่งต่างๆ ซึ่งแบ่งเป็นประโยคเชิงซ้อน (compound statement) กับ ประโยคนิพจน์ (expression statment) โดยประโยคเชิงซ้อนจะอยู่ภายในวงเล็บปีกกาคู่หนึ่ง {และ} โดยในหนึ่งประโยคเชิงซ้อนจะมีประโยคนิพจน์ที่แยกจากกันด้วยเครื่องหมาย semicolon (;) หลายๆ ประโยครวมกัน และ อาจมีวงเล็บปีกกาใส่ประโยคเชิงซ้อนย่อยเข้าไปอีกได้

3. ตัวแปรกับชนิดของข้อมูล

ในการเขียนโปรแกรมคอมพิวเตอร์กระบวนการสำคัญที่เกิดขึ้น คือ การรับข้อมูล การประมวลผลข้อมูล และการแสดงผลข้อมูล จะเห็นว่าสิ่งที่เป็นส่วนสำคัญที่สุดคือข้อมูล การทำงานของโปรแกรมขณะใดขณะหนึ่ง จะต้องมีการเก็บข้อมูลไว้ในคอมพิวเตอร์ โดยรับข้อมูลจากอุปกรณ์รับข้อมูลไปเก็บไว้ในส่วนที่เรียกว่า หน่วยความจำ และส่งข้อมูลจากหน่วยความจำไปประมวลผลในหน่วยประมวลผลกลาง โดยผ่านคำสั่งต่าง ๆ เมื่อประมวลผลเสร็จแล้วก็นำผลลัพธ์ที่ได้กลับมาเก็บไว้ที่หน่วยความจำอีก เมื่อต้องการให้แสดงผลก็จะใช้คำสั่งให้ไปอ่านข้อมูลจากหน่วยความจำ ส่งข้อมูลนั้นไปยังอุปกรณ์แสดงผล

ชนิดของข้อมูล คือ สิ่งที่ใช้กำหนดลักษณะและขอบเขตของข้อมูลนั้นๆโดยข้อมูลที่มีชนิดของข้อมูลแตกต่างกัน ก็จะเก็บข้อมูลได้ในลักษณะแตกต่างกัน และขอบเขตของข้อมูลที่เก็บได้ก็จะไม่เท่ากัน การประกาศตัวแปรมี 2 ลักษณะได้แก่ตัวแปรโกบอล และ ตัวแปรโลคอล ซึ่งมีรูปแบบการประกาศตัวแปรที่เหมือนกัน แต่จะมีคุณสมบัติต่างกัน โดยจะเห็นได้ชัดเจนเมื่อมีการสร้างฟังก์ชันมาใช้งาน เป็นพารามิเตอร์ที่ใช้งานในภาษาซี เพื่อกำหนดลักษณะฟังก์ชันต้นแบบ และ ตัวแปรที่ใช้งานในโปรแกรม มี 4 ชนิด

3.1. ชนิดข้อมูลแบบไม่มีค่า (void Type)

เป็นพารามิเตอร์ที่ใช้งานในส่วนของฟังก์ชันโพรโทรไทพ์ การสร้างและรับค่าจากการเรียกใช้งานฟังก์ชัน จะไม่ใช้ชนิดข้อมูลแบบ void นี้กำหนดค่าให้กับตัวแปร แต่จะแนะนำชนิดข้อมูลประเภทนี้กำหนดไว้ที่ฟังก์ชันในกรณีที่ไม่ต้องการให้ฟังก์ชันมีการรับค่าใดๆ เข้ามาหรือส่งค่าใดๆ กลับไป

3.2. ชนิดข้อมูลแบบตัวอักษร (character Type)

ชนิดข้อมูลประเภท char ซึ่งชนิดของข้อมูลในรูปแบบนี้จะเก็บข้อมูลได้ 1 ตัวอักษรเท่านั้น ตัวอักษรตั้งแต่ A-Z เลข 0-9 และสัญลักษณ์ ต่างๆ ตามมาตรฐาน ACSII (American Standard Code Information Interchange) และสามารถรับข้อมูลจานวนเต็มตั้งแต่ -128 ถึง 127 จะใช้ขนาดหน่วยความจำ 1 ไบต์ หรือ 8 บิต

3.3. ชนิดข้อมูลแบบเลขจำนวนเต็ม (Integer Type)

เป็นชนิดข้อมูลแบบเลขจำนวนเต็ม ไม่มีทศนิยม ซึ่งภาษาซีจะแบ่งชนิดข้อมูลนี้ออกได้เป็น 4 ระดับ คือ shot int เป็นเลขจำนวนเต็มแบบสั้น int เป็นเลขจำนวนเต็มแบบปกติ long int เป็นเลขจำนวนเต็มแบบยาว

- short int ข้อมูลชนิดจำนวนเต็มขนาด 2 bytes
- int ข้อมูลชนิดจำนวนเต็มขนาด 2 bytes หรือ 4 byte(ในคอมไพเลอร์ 32 bit ตัวแปร int มีขนาด 4 byte แต่ในคอมไพเลอร์ 16 bit ตัวแปร int มีขนาด 2 byte)
- long int ข้อมูลชนิดจำนวนเต็มขนาด 4 bytes
- long long int เป็นข้อมูลชนิดจำนวนเต็มขนาด 8 bytes

3.4. ชนิดข้อมูลแบบเลขทศนิยม (Floating point Type)

Float เป็นพารามิเตอร์หลักที่ใช้กับข้อมูลชนิดจำนวนทศนิยม โดยมีการใช้งาน 3 รูปแบบ

-float ข้อมูลชนิดจำนวนทศนิยมขนาด 4 byte

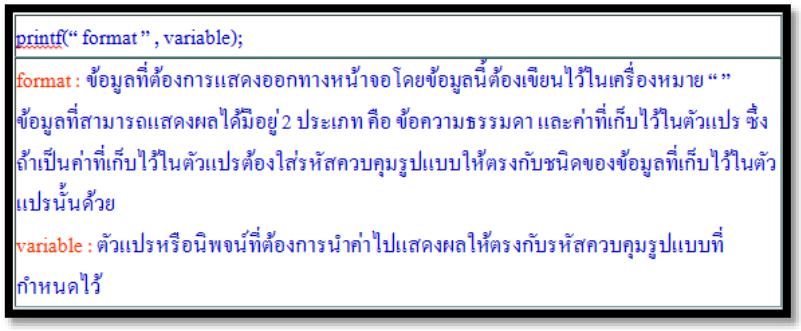
-double ข้อมูลชนิดจำนวนทศนิยมขนาด 8 byte

-long double ข้อมูลชนิดจำนวนทศนิยมขนาด 10 byte

4. การแสดงผลและการรับข้อมูล (Printf and Scanf)

4.1. การแสดงผลออกทางหน้าจอ

การแสดงผลข้อมูลออกทางหน้าจอสามารถทำได้ง่าย โดยเรียกใช้คำสั่งหรือฟังก์ชันมาตรฐานที่ภาษาซีเตรียมไว้ให้ใช้คำสั่ง printf ถือว่าเป็นคำสั่งพื้นฐานที่สุดในการแสดงผลข้อมูลทุกชนิดออกทางหน้าจอ ไม่ว่าจะเป็นจำนวนเต็ม (int) , ทศนิยม (float) , ข้อความ (string) หรืออักขระ นอกจากนี้คำสั่งยังมีความยืดหยุ่นสูง โดยเราสามารถกำหนดหรือจัดรูปแบบการแสดงผลให้มีระเบียบ หรือเหมาะสมตามความต้องการได้อีกด้วย รูปแบบการเรียกใช้คำสั่ง printf แสดงได้ดังนี้



ภาพที่ 2-1 การเรียกใช้ printf

4.2. อักขระควบคุมการแสดงผล

นอกจากนี้เรายังสามารถจัดรูปแบบการแสดงผลให้ดูเป็นระเบียบมากขึ้น เช่น การขึ้นบรรทัดใหม่หลังแสดงข้อความ หรือเว้นระยะแท็บระหว่างข้อความ โดยใช้อักขระควบคุม การแสดงผลร่วมกับคาสั่ง printf ในภาษาซีมี อักขระควบคุมการแสดงผลหลายรูปแบบด้วยกัน ดังแสดงต่อไปนี้

รหัสควบคุมรูปแบบ	การนำไปใช้งาน
%d	สำหรับแสดงผลค่าของตัวแปรชนิดจำนวนเต็ม (int, short, unsigned short, long, unsigned long)
%u	สำหรับแสดงผลตัวเลขจำนวนเต็มบวก (unsigned short, unsigned long)
%o	สำหรับแสดงผลออกมาในรูปแบบของเลขฐานแปด
%x	สำหรับแสดงผลออกมาในรูปแบบของเลขฐานสิบหก
%f	สำหรับแสดงผลค่าของตัวแปรชนิดจำนวนทศนิยม (float, double, long double)
%e	สำหรับแสดงผลตัวเลขทศนิยมออกมาในรูปแบบของ (E หรือ e) ยกกำลัง (float, double, long double)
%c	สำหรับแสดงผลอักขระ 1 ตัว (char)
%s	สำหรับแสดงผลข้อความ (string หรืออักขระมากกว่า 1 ตัว)
%p	สำหรับแสดงผลตัวชี้ตำแหน่ง (pointer)

ภาพที่ 2-2 รหัสควบคุมรูปแบบ

4.3. รับข้อมูลจากคีย์บอร์ด

การทำงานของโปรแกรมส่วนใหญ่มักจะเป็นการเชื่อมโยงกับผู้ใช้แบบ 2 ทิศทาง นั่นก็คือทั้งภาคของแสดงผลการทางานออกทางหน้าจอ และภาคของการรับข้อมูลจากผู้ใช้เข้ามาทางคีย์บอร์ดเพื่อร่วมในการประมวลผลของโปรแกรม ซึ่งในภาคของการรับข้อมูลจากผู้ใช้ ภาษาซีกำหนดคำสั่งและฟังก์ชันมาตรฐานเอาไว้ให้เรียกใช้แล้ว เช่นเดียวกับภาคของการแสดงผล รายละเอียดของคำสั่งเหล่านี้ได้แก่ คำสั่ง scanf() ในภาษาซี การรับข้อมูลจากคีย์บอร์ดสามารถทำได้โดยการเรียกใช้ฟังก์ชัน scanf() ซึ่งเป็นฟังก์ชันมาตรฐานสำหรับรับข้อมูลจากคีย์บอร์ด โดยสามารถรับข้อมูลได้ทุกประเภท ไม่ว่าจะเป็นจำนวนเต็ม (int) , ทศนิยม (float) , อักขระ (char) หรือข้อความก็ตาม รูปแบบการเรียกใช้คำสั่ง scanf() คล้ายกับการเรียกใช้คาสั่ง printf() ดังแสดงต่อไปนี้

```
scanf("format" , &variable) ;
```

format : การใช้รหัสควบคุมรูปแบบ เพื่อกำหนดชนิดของข้อมูลที่จะรับเข้ามาจากคีย์บอร์ด โดยรหัสควบคุมรูปแบบจะใช้ชุดเดียวกับรหัสควบคุมรูปแบบของคำสั่ง printf()

variable : ตัวแปรที่จะใช้เก็บค่าข้อมูลที่รับเข้ามาจากคีย์บอร์ด โดยชนิดของตัวแปรจะต้องตรงกับรหัสควบคุมรูปแบบที่กำหนดไว้ นอกจากนี้ หน้าชื่อของตัวแปรจะต้องนำหน้าด้วยเครื่องหมาย & ยกเว้นตัวแปรสตริงสำหรับเก็บข้อความเท่านั้นที่ไม่ต้องนำหน้าด้วยเครื่องหมาย &

ภาพที่ 2-3 การเรียกใช้ scanf

5. ตัวนำเดินการทางคณิตศาตร์

ตัวนำเดินการทางคณิตศาตร์ในโปรแกรมภาษาซี คือการนำค่าคงที่หรือตัวแปรมาเชื่อมต่อกันด้วยเครื่องหมายทางคณิตศาสตร์ นิพจน์คณิตศาสตร์จะมีลักษณะคล้ายกับสมการทางคณิตศาสตร์ เช่น c = a * b , (10 + 5) * 10 % 9 = 15

6. คำสั่งเงื่อนไข

เป็นคำสั่งที่ใช้เลือกทำโดยพิจารณาจากเงื่อนไขที่กำหนด

6.1. คำสั่งเงื่อนไข (if-else)

เป็นคำสั่งที่ช่วยให้การตรวจสอบเงื่อนไขสมบูรณ์ขึ้น โดยหากตรวจสอบเงื่อนไขของคำสั่ง if เป็นเท็จ ก็จะเข้ามาทำงานภายในบล็อกของคำสั่ง else แทน กล่าวคือ หากตรวจสอบเงื่อนไขแล้วเป็นจริง ก็จะประมวลผลคำสั่งในบล็อกของ if แต่หากเงื่อนไขเป็นเท็จ ก็จะประมวลผลคำสั่งในบล็อกของ else แทน และเมื่อตรวจสอบเงื่อนไขและประมวลผลตามคำสั่งเงื่อนไข if-else เรียบร้อยแล้ว ก็จะทำงานตามคำสั่งที่อยู่ถัดจาก if-else นั้นถัดไป

6.2. คำสั่งเงื่อนไข (switch-case)

เป็นข้อความสั่งที่ให้เลือกทำข้อความสั่ง หรือกลุ่มข้อความสั่งใดๆโดยพิจารณาจากค่าของนิพจน์ ถ้าค่าของนิพจน์มีค่าเท่ากับค่าใด ก็จะทำข้อความสั่งที่อยู่ใน case นั้น ถ้าค่าของนิพจน์ตรงกับค่าของ case ชุดไหน ก็จะทำชุดคำสั่งชุดนั้น แต่ถ้าค่าของนิพจน์ไม่ตรงกับค่าของ case ชุดไหนเลยก็จะทำชุดข้อความสั่งใน default

7. คำสั่งทำซ้ำ

ลูป(loop)ในที่นี้มีความหมายว่า การวนซ้ำซึ่งการวนซ้ำในทางภาษาคอมพิวเตอร์ คือ การทำคำสั่ง หรือชุดคำสั่งนั้นซ้ำกันหลายๆครั้ง รูปแบบของลูปในการตรวจสอบว่าจะให้ลูปนั้นจบการทำงานเมื่อไรนั้น จะมีรูปแบบของการตรวจสอบเงื่อนไขอยู่ 2 แบบ

- Pretest Loop ลูปประเภทนี้จะทำการตรวจสอบเงื่อนไขก่อนว่าเป็นจริง หรือเป็นเท็จถ้าเป็นจริง ก็ให้เข้าไปทำคำสั่งหรือชุดคำสั่งต่อไป และเมื่อทำคำสั่งหรือชุดคำสั่งเสร็จแล้ว ก็จะกลับมาทำการตรวจสอบเงื่อนไขอีกครั้ง และจะทำเช่นนี้ไปเรื่อยๆจนกว่าเงื่อนไขจะเป็นเท็จ ก็จบการทำงานของลูป

- Posttest loop ลูปประเภทนี้จะทำคำสั่ง หรือชุดคำสั่งก่อน เมื่อเสร็จแล้วถึงจะมาตรวจสอบเงื่อนไขว่าเป็นจริงหรือเป็นเท็จ ถ้าเป็นจริงก็จะกลับไปทำคำสั่ง หรือชุดคำสั่งเดิมอีกครั้ง และจะทำจนกว่าเงื่อนไขจะเป็นเท็จ เช่นเดียวกันการกำหนดและปรับปรุง

ในการใช้ลูป จะมีการกระทำที่สำคัญอยู่ 2 อย่าง ที่จะขาดไม่ได้เลยซึ่งถ้าขาดไปจะทำให้ลูปนั้นไม่ทำงาน หรือลูปทำงานแบบไม่มีวันจบ

7.1. คำสั่งทำซ้ำ (for)

เป็นคำสั่งวนซ้ำแบบ pre-test loop คือ ประมวลผลก่อนการทำซ้ำ เหมาะกับการทำซ้ำแบบรู้จำนวนรอบของการทำซ้ำ คำสั่งการทำซ้ำ for ประกอบด้วย 3 ส่วน คือ

7.1.2 ส่วนของการกำหนดค่าเริ่มต้นกับตัวแปร

7.1.3 ส่วนของเงื่อนไขที่ต้องการตรวจสอบ

7.1.4 ส่วนการปรับค่าตัวแปร

7.2. คำสั่งทำซ้ำ (while)

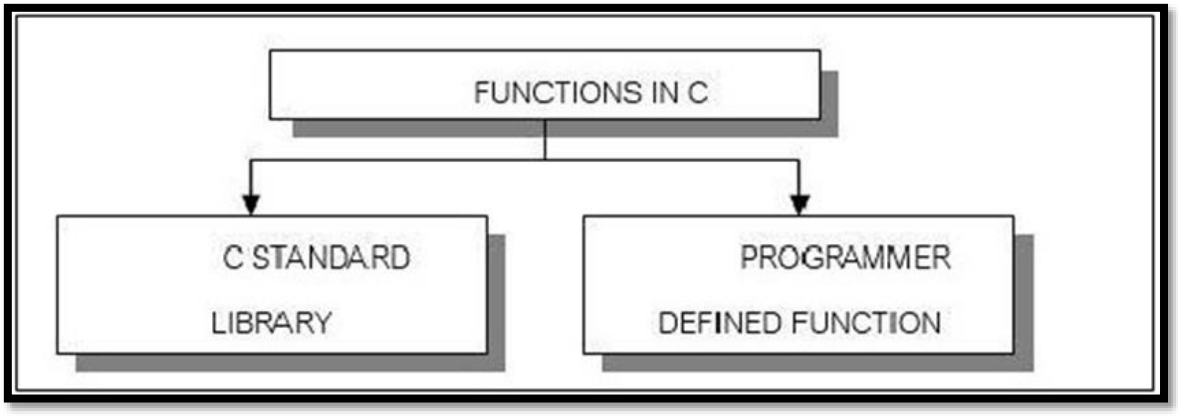
หลักการทำงาน คือ จะทำการตรวจสอบเงื่อนไขก่อนการทางานทุกครั้ง หากเงื่อนไขเป็นจริง จึงเข้าทำงานในบล็อกการทำงานของลูป while แต่หากเงื่อนไขเป็นเท็จจะไม่เข้าสู่การทำงานของลูป while แต่จะไปทำงานคำสั่งถัดไปที่อยู่นอกลูป while ทันที

7.3. คำสั่งทำซ้ำ (do-while)

หลักการทำงาน คือ คำสั่งนี้จะทำงานอย่างน้อยที่สุด 1 ครั้งก่อนเสมอไม่ว่าเงื่อนไขจะเป็นจริงหรือเท็จก็ตาม จากนั้นจึงค่อยตรวจสอบเงื่อนไขภายหลัง ซึ่งหากเงื่อนไขเป็นจริง ก็จะวนลูปกลับไปทำงานที่บล็อกเงื่อนไข do-while อีกครั้ง แต่หากเงื่อนไขเป็นเท็จก็จะหลุดจากการทำงานของคำสั่ง do-while ไปทำคำสั่งที่อยู่นอกลูปต่อไป

8. ฟังก์ชัน (Function)

โปรแกรมย่อยในภาษาซี จะถูกเรียกว่า ฟังก์ชันในภาษาซี ซึ่งสามารถแบ่งตามแหล่งที่มาได้ 2 ประเภท คือ



ภาพที่ 2-4 ฟังก์ชัน (Function)

8.1. ฟังก์ชันมาตรฐานในภาษาซี (C Standard Function)

ฟังก์ชันมาตรฐานในภาษาซี ซึ่งจะอยู่ในไลบรารีภาษาซีมาตรฐาน (C Standard Library) ไลบรารีภาษาซีมาตรฐานประกอบด้วยฟังก์ชันต่างๆมากมาย ไม่ว่าจะใช้สำหรับการคำนวณทางคณิตศาสตร์ การจัดการกับข้อความ การจัดการกับ input/output และอื่นๆ ซึ่งจะทำให้งานของโปรแกรมเมอร์ง่ายขึ้น โดยการใช้งานฟังก์ชันประเภทนี้จะต้องรวม (include) ไลบรารีที่ต้องการใช้งาน เพื่อให้ตัวแปลภาษารู้ว่าฟังก์ชันที่โปรแกรมเมอร์ต้องการใช้อยู่ในไลบรารีมาตรฐานตัวใด ตัวอย่างเช่น หากต้องการใช้ฟังก์ชัน printf () ซึ่งอยู่ในไลบรารีมาตรฐานสำหรับเกี่ยวกับอินพุตและเอาต์พุต (standard input/output) ที่ชื่อ stdio เราใช้คำสั่งดังนี้ #include<stdio.h>

8.2. ฟังก์ชันที่สร้างขึ้นใหม่ (Defined Function)

โปรแกรมเมอร์สามารถเขียนฟังก์ชันเพื่อนิยามการทำงานที่จะเรียกใช้ในส่วนต่างๆของโปรแกรม โดยฟังก์ชันการทำงานดังกล่าวจะถูกเขียนไว้ในฟังก์ชันเพียงครั้งเดียวเท่านั้น แต่สามารถเรียกใช้งานได้หลายครั้ง ตัวแปรที่ประกาศหรือคำสั่งที่เรียกในฟังก์ชันใดๆมีขอบเขตการใช้งานอยู่ในฟังก์ชันนั้นๆเท่านั้น นั่นหมายความว่าฟังก์ชันอื่นๆรวมทั้งฟังก์ชัน main() จะไม่ทราบการทำงานภายในหรือคำสั่งต่างๆ ในฟังก์ชันนั้น หรือพูดอีกอย่างได้ว่าฟังก์ชันแต่ละฟังก์ชันจะไม่ทราบการทำงานภายในของฟังก์ชันอื่น แต่จะสามารถเรียกใช้งานฟังก์ชันนั้นได้เท่านั้น

ฟังก์ชันจะถูกเรียกให้ทำงาน ก็ต่อเมื่อมีการเรียกใช้ฟังก์ชัน (call function) การเรียกใช้ฟังก์ชันจะต้องระบุชื่อของฟังก์ชัน ข้อมูลที่จะให้กับฟังก์ชันนั้น (argument) ซึ่งฟังก์ชันนั้นต้องการเพื่อใช้ในการทำงานตามที่ได้ออกแบบไว้ หากเราจะเปรียบเทียบการทำงานของฟังก์ชันได้กับการสั่งงานของหัวหน้า (Boss) คนหนึ่ง ซึ่งเป็นผู้สั่งงาน (caller) ที่สั่งงาน (calling function) ให้คนงาน (Worker) ทำงานอย่างใดอย่างหนึ่ง (called function) โดยให้รายละเอียดการทำงานแก่คนงาน หลังจากทำงานเสร็จสิ้น คนงานจะรายงานผลกลับมายังหัวหน้า (return)

9. ตัวแปรแถวลำดับ (Array)

ตัวแปรแถวลำดับ (Array) คือ ตัวแปรที่สามารถเก็บข้อมูลได้เป็นชุด โดยสร้างตัวแปรขึ้นมาเพียงตัวแปรเดียว (ตัวแปร 1 ตัว ตัดเก็บข้อมูลได้หลายค่า) แต่ข้อมูลนั้นต้องเป็นชนิดเดียวกัน ประเภทของตัวแปรชุด มีดังนี้

9.1. ตัวแปรชุด 1 มิติ (one dimension arrays หรือ single dimension arrays) ตัวแปรชุด 1 มิติ คือ ตัวแปรชุดที่มีตัวเลขกำกับ (subscript) เพียง 1 ตัวและ salary [20] เป็นต้น

9.2. ตัวแปรชุดหลายมิติ (multi-dimension arrays) ตัวแปรชุดหลายมิติ คือตัวแปรชุดที่มีตัวเลขกำกับ (subscript) ตั้งแต่ 2 ตัวขึ้นไป

10. การโปรแกรมกับแฟ้มข้อมูล (File)

แฟ้มเป็นหน่วยงานภายนอก (External unit) เป็นที่เก็บรวบรวมข้อมูลที่เกี่ยวข้องกันเป็นรายการ โดยข้อมูลที่เก็บไว้ในแฟ้ม เมื่อปิดเครื่องคอมพิวเตอร์ข้อมูลจะไม่สูญหาย มักเป็นข้อมูลที่มีขนาดใหญ่ ไม่สามารถเก็บไว้ในหน่วยความจำได้ แฟ้มถูกจัดเก็บในหน่วยความจารอง

(Auxiliary/Secondary storage device) เมื่อโปรแกรมทำการอ่านข้อมูลจากแฟ้ม หมายถึงการสำเนาข้อมูลจากอุปกรณ์ภายนอกไปหน่วยความจำ แต่เมื่อโปรแกรมทำการเขียนข้อมูลลงแฟ้มจะหมายถึงการสำเนาข้อมูลจากความจำภายในไปยังอุปกรณ์ภายนอก ในการสำเนาข้อมูลจำเป็นต้องใช้พื้นที่ชั่วคราวในการพักข้อมูล ซึ่งเป็นส่วนหนึ่งของหน่วยความจำ เรียกว่า บัฟเฟอร์ (Buffer)

11. ตัวชี้ (Pointer)

ค่าของข้อมูลที่ถูกระบุด้วยตัวชี้จะเป็นตัวดำเนินการที่อยู่ ซึ่งแทนด้วยเครื่องหมาย & เป็นการอ้างถึงที่อยู่ตัวแปรนั้นๆ ทำให้เราสามารถเข้าไปแก้ไขค่า หรือข้อมูลต่างๆที่อยู่ในตัวแปรนั้นๆได้โดยตรง

11.1. การเข้าถึงตัวแปรผ่านตัวชี้

การที่เราอ้างถึงค่าที่ตัวแปรประเภทตัวชี้อยู่นั้นว่ามีค่าอะไร นั่นก็คือการใช้เครื่องหมาย * เป็นการอ้างถึงข้อมูลในตัวแปรที่เก็บนั้นๆว่าเก็บค่าอะไรอยู่

- การประกาศตัวแปรชนิดตัวชี้ ชนิดของตัวแปร *ชื่อตัวแปรประเภทตัวชี้

- การใส่ค่าที่อยู่ให้กับตัวแปรแบบตัวชี้ ชื่อตัวแปรประเภทตัวชี้ = &ชื่อตัวแปรปกติ

- การแสดงค่าทางจอภาพตัวแปรแบบตัวชี้ printf (“%ชนิดของข้อมูล”,*ชื่อตัวแปรประเภทตัวชี้) นอกจากนี้ยังมีการเข้าถึงตัวชี้ของตัวชี้อีกที่หนึ่ง ซึ่งในโครงงานนี้ไม่มีการใช้ตัวชี้ระดับนั้นจึงขอข้ามเรื่องนี้

12. การกำหนดชนิดโครงสร้าง (structure)

โครงสร้างหมายถึงองค์ประกอบของข้อมูลที่เกี่ยวข้องกัน ซึ่งแต่ละองค์ประกอบอาจมีชนิดของข้อมูลที่เหมือนหรือแตกต่างกันก็ได้ ในเรื่องแถวลำดับจะเห็นได้ว่าสมาชิกทุกตัวต้องเป็นชนิดเดียวกัน แต่ถ้าโครงสร้างสมาชิกหรือองค์ประกอบต่างๆอาจเป็นชนิดที่แตกต่างกันได้

- ฟิลด์ คือ ชนิดของตัวแปรต่างๆจะเป็น int, char, float ฯลฯ

- ตัวแปรลิสต์ คือ ชื่อตัวแปรที่อยู่ภายใต้วงเล็บปีกกาของโครงสร้าง {}

13. สตริง (String or Array of charactor)

สตริง หมายถึง เซตของอักขระที่เรียงต่อกัน ซึ่งอาจมีความยาวที่แน่นอนหรือความยาวที่เปลี่ยนแปลงได้ กล่าวคือ char หลายๆตัวเรียงต่อกัน ซึ่งในสตริงในภาษาซีจะถูกเก็บในตัวแปรประเภทแถวลำดับ ซึ่งมีความยาวแปรเปลี่ยนได้และใช้อักขระ Null แทนจุดสิ้นสุดสตริง

13.1. การประกาศตัวแปรสตริง

การประกาศตัวแปรสตริงประกาศได้ 2 แบบ คือ

- ประกาศเป็นแถวลำดับอักขระ

- ประกาศเป็นชนิดตัวชี้

2.2 ปริศนา (Puzzle)

คือปัญหาสำหรับท้าทายความเฉลียวฉลาด (ingenuity) ของมนุษย์ ปริศนามักจะถูกออกแบบมาเพื่อความบันเทิง แต่บางครั้งก็กลายเป็นปัญหาทางตรรกศาสตร์หรือคณิตศาสตร์อย่างจริงจัง สำหรับกรณีหลัง ผลสำเร็จของปริศนาอาจมีความสำคัญในการพิสูจน์และการวิจัยทางด้านคณิตศาสตร์

การหาผลสำเร็จของปริศนาบางอย่างอาจต้องใช้แบบแผน (pattern) และขั้นตอนที่เฉพาะเจาะจง บุคคลที่มีความสามารถในการเรียนรู้ได้เร็ว อาจสามารถไขปัญหาได้ดีกว่าบุคคลอื่น ปริศนาซึ่งมีพื้นฐานอยู่บนการเสาะหาและการค้นพบแนวทางในการแก้ปัญหา อาจแก้ได้รวดเร็วกว่าด้วยทักษะการอนุมานที่ดี

1. Puzzle มีประโยชน์อย่างไร

ของเล่นพัฒนาทักษะนั้นอาจจะมีหลากหลายแบบ ซึ่งมักจะถูกเรียกรวมๆว่า Puzzle โดยมักจะมีลักษณะเด่นๆสำคัญ ก็คือ มักจะเป็นปัญหารูปแบบต่างๆ เช่น ภาพที่ไม่สมบูรณ์ ตัวต่อที่มีชิ้นส่วนและแบบให้ดู หรือ และวิธีการเล่น ก็คือการแก้ปัญหา และนั่นทำให้มันกลายเป็นของเล่นยอดฮิตที่คุณพ่อคุณแม่มักจะให้ความสนใจ เพราะเป็นจุดกึ่งกลางระหว่าง การเรียนรู้ และความสนุกนั่นเอง

บทที่ 3

วิธีการดำเนินโครงงาน

3.1 วิเคราะห์และออกแบบความสำมารถของระบบ

ผู้จัดทำต้องทำการตรวจสอบก่อนว่า ระบบลีนุกซ์ของทางเซิร์ฟเวอร์ dekdee.buu.ac.th ที่จะทำการส่งโปรแกรมนั้นสามารถใช้ library อะไรได้บ้าง เพราะบางโปรแกรมต้องใช้ library แปลกซึ่งทำแล้วเกิดปัญหาคือตัวโปรแกรมที่ทำมาแล้วไม่สามารถคอมไพล์ได้ เนื่องจากการนำเข้า library ดังกล่าวไม่ได้ จึงต้องตรวจสอบก่อน

1. ความสามารถของระบบสำหรับผู้ดูแลระบบ
   1.1. สามาระแก้ไขข้อมูลคะแนนได้
2. ความสามารถของระบบสำหรับผู้เล่น
   2.1. สมารถเลือกระดับความยากของเกม ง่าย ปานกลาง ยาก
   2.2. สามารถใส่ชื่อผู้เล่น
   2.3. สามารถระบุเวลา วันที่ และจำนวนสไลด์แต่ละรอบที่เล่น

3.2 รหัสเทียมของโปรแกรม

อัลกอริทึม : เกมเรียงตัวเลข (Puzzle number)

Input: ชื่อ และ การควบคุมเกม

Output: เกมเรียงตัวเลข (Puzzlenumber)

1. เริ่มต้น
2. ประกาศ library string.h
3. ประกาศ library time.h
4. ประกาศ library stdlib.h
5. ประกาศ library unistd.h
6. ประกาศ library termios.h
7. ประกาศ struct ชื่อ PLAYER ใช้เก็บ

   7.1.ประกาศตัวแปร name เป็นชนิดตัวอักขระ และเป็นตัวแปรแถวลำดับ มีค่าดัชนี 10

   7.2.ประกาศตัวแปร level เป็นชนิดตัวอักขระ และเป็นตัวแปรแถวลำดับ มีค่าดัชนี 5

7.3. ประกาศตัวแปร time เป็นชนิดเลขทศนิยม

7.4. ประกาศตัวแปร slide เป็นชนิดจำนวนเต็ม

8. ประกาศฟังก์ชัน clear_data ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์

8.1. ประกาศตัวแปร in เป็นชนิด FILE แบบตัวชี้

8.2. ประกาศตัวแปร i, check=0 เป็นชนิดจำนวนเต็ม

8.3. ประกาศตัวแปร ch, ans, เป็นชนิดตัวอักขระ และ use, pass เป็นชนิดตัวอักขระและเป็นตัวแปรแถวลำดับ ที่มีดัชนีเท่ากับ 10

8.4. ทำซ้ำโดยใช้ while โดยที่ strcmp (ans,"edit") != 0 และ strcmp (ans,"exit") != 0

8.4.1. แสดง Do you want to exit or edit? (exit/edit): ออกทางหน้าจอ

8.5. ถ้า strcmp (ans,"edit") == 0

8.5.1. ทำซ้ำโดยใช้ while โดยที่strcmp (ans,"yes") !=0 && strcmp(ans,"no")!=0

8.5.1.1. แสดงDo you want to clear data? (Yes/no):ออกทางหน้าจอ

8.5.1.2. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ans

8.5.2. ถ้า strcmp(ans,"yes") == 0

8.5.2.1. ทำซ้ำโดยใช้ while โดยที่ check < 3

8.5.2.1.1. ทำซ้ำโดยใช้ while โดยที่ strcmp (use,"admin") != 0

1. กำหนดค่า check++;
2. แสดง Username: ออกทางหน้าจอ
3. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ use
4. ถ้า check > 3 ให้หยุดลูป

8.5.2.1.2. กำหนด check = 0;

8.5.2.1.3. ทำซ้ำโดยใช้ whileโดยที่strcmp (pass,"manzaza") != 0

1. กำหนดค่า check++;
2. แสดง Password: ออกทางหน้าจอ
3. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ pass

8.5.2.1.4. ถ้า check < 3

1. เปิดไฟล์ Score_data.txt โหมด w
2. แสดง Clear data completely... ออกทางหน้าจอ
3. แสดง Press X to menu: ออกทางหน้าจอ
4. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'
5. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch

8.5.2.1.5. มิฉะนั้น แสดง Press X to menu: ออกทางหน้าจอ

8.5.2.1.6. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'

1. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch

8.5.3. มิฉะนั้น แสดง Data not clear... ออกทางหน้าจอ

9. ประกาศฟังก์ชัน credit ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์

9.1. ประกาศตัวแปร ch, text เป็นชนิดตัวอักขระ

9.2. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม

9.3. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้

9.4. เปิดไฟล์ credit.txt โหมด r

9.5. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์

9.6. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text

9.7. ทำซ้ำโดยใช้ while โดยที่ result != EOF

9.7.1. แสดงค่า text ออกทางหน้าจอ

9.7.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์

9.7.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text

9.8. ปิดไฟล์ in

9.9. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'

9.9.1. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch

10. ประกาศฟังก์ชัน enter_name ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์

10.1. ประกาศตัวแปร ch, text เป็นชนิดตัวอักขระ

10.2. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม

10.3. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้

10.4. เปิดไฟล์ entername.txt โหมด r

10.5. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์

10.6. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text

10.7. ทำซ้ำโดยใช้ while โดยที่ result != EOF

10.7.1. แสดงค่า text ออกทางหน้าจอ

10.7.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์

10.7.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text

10.8. ปิดไฟล์ in

10.9. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'

10.9.1. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch

11. ประกาศฟังก์ชัน high_score ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์
    11.1. ประกาศตัวแปร ch, text, ch2, text2 เป็นชนิดตัวอักขระ
    11.2. ประกาศตัวแปร result, result2 เป็นชนิดจำนวนเต็ม
    11.3. ประกาศตัวแปร in, in2 เป็นชนิด FILE ประเภทตัวชี้
    11.4. เปิดไฟล์ scoretable.txt โหมด r
    11.5. กำหนด result2 เท่ากับ การรับค่าจากการอ่านไฟล์
    11.6. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text2
    11.7. ทำซ้ำโดยใช้ while โดยที่ result2 != EOF
        11.7.1. แสดงค่า text2 ออกทางหน้าจอ
        11.7.2. กำหนด result2 เท่ากับ การรับค่าจากการอ่านไฟล์
        11.7.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text2
    11.8. ปิดไฟล์ in2
    11.9. เปิดไฟล์ Score_data.txtโหมด r
    11.10. ถ้า result != EOF
        11.10.1. ทำซ้ำโดยใช้ while โดยที่ result != EOF
            11.10.1.1. แสดงค่า text ออกทางหน้าจอ
            11.10.1.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
            11.10.1.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
        11.10.2. เรียกใช้ฟังก์ชัน clear_data
    11.11. มิฉะนั้น แสดง No data... ออกทางหน้าจอ
    11.12. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'
        11.12.1. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch

12. ปิดไฟล์ in

13. ประกาศฟังก์ชัน how2play ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์
    13.1. ประกาศตัวแปร ch, text เป็นชนิดตัวอักขระ
    13.2. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม
    13.3. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้
    13.4. เปิดไฟล์ how2play.txt โหมด r
    13.5. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
    13.6. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text

13.7. ทำซ้ำโดยใช้ while โดยที่ result != EOF
13.7.1. แสดงค่า text ออกทางหน้าจอ
13.7.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
13.7.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
13.8. ปิดไฟล์ in
13.9. ทำซ้ำโดยใช้ while โดยที่ ch != 'x' && ch != 'X'
13.9.1. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ch
14. ประกาศฟังก์ชัน main_menu ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์
14.1. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม
14.2. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้
14.3. เปิดไฟล์ main_menu.txt โหมด r
14.4. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
14.5. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
14.6. ทำซ้ำโดยใช้ while โดยที่ result != EOF
14.6.1. แสดงค่า text ออกทางหน้าจอ
14.6.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
14.6.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
14.7. ปิดไฟล์ in
15. ประกาศฟังก์ชัน mickey ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์
15.1. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม
15.2. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้
15.3. เปิดไฟล์ mickey.txt โหมด r
15.4. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
15.5. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
15.6. ทำซ้ำโดยใช้ while โดยที่ result != EOF
15.6.1. แสดงค่า text ออกทางหน้าจอ
15.6.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
15.6.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
15.7. ปิดไฟล์ in
16. ประกาศฟังก์ชัน puzzle ชนิด ไม่มีการคืนค่าและไม่มีพารามิเตอร์
16.1. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม

16.2. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้
16.3. เปิดไฟล์ puzzle.txt โหมด r
16.4. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
16.5. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
16.6. ทำซ้ำโดยใช้ while โดยที่ result != EOF
16.6.1. แสดงค่า text ออกทางหน้าจอ
16.6.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
16.6.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
16.7. ปิดไฟล์ in
17. ประกาศฟังก์ชัน rd_check ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์
17.1. รับพารามิเตอร์ int size, a[][size], b[][size]
17.2. ประกาสตัวแปร i, j เป็นชนิดจำนวนเต็ม
17.3. ทำซ้ำโดยใช้ for โดยที่ i<size
17.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size
17.3.1.1. ถ้า a[i][j] == b[i][j]
17.3.1.1.1. เรียกใช้ฟังก์ชัน rd_puzzle(size,a)
17.3.1.1.2. j++
17.3.2. i++
18. ประกาศฟังก์ชัน rd_puzzle ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์
18.1. รับพารามิเตอร์ int size,int a[][size]
18.2. ประกาศตัวแปร random,i เป็นชนิดจำนวนเต็ม
18.3. ข้อความคำสั่ง srand(time(NULL))
18.4. ทำซ้ำโดยใช้ for โดยที่ i<50
18.4.1. ข้อความคำสั่ง random = rand() % 9
18.4.2. เข้า Switch case
18.4.2.1. เมื่อ choice เท่ากับ case : 0
18.4.2.1.1. เรียกใช้ฟังก์ชัน slideright(size,a)
18.4.2.1.2. เรียกใช้ฟังก์ชัน slideleft(size,a)
18.4.2.1.3. เรียกใช้ฟังก์ชัน slideup(size,a)
18.4.2.1.4. เรียกใช้ฟังก์ชัน slideright(size,a)
18.4.2.1.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.2. เมื่อ choice เท่ากับ case : 1

18.4.2.2.1. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.2.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.2.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.2.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.2.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.3. เมื่อ choice เท่ากับ case : 2

18.4.2.3.1. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.3.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.3.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.3.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.3.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.4. เมื่อ choice เท่ากับ case : 3

18.4.2.4.1. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.4.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.4.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.4.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.4.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.5. เมื่อ choice เท่ากับ case : 4

18.4.2.5.1. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.5.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.5.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.5.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.5.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.6. เมื่อ choice เท่ากับ case : 5

18.4.2.6.1. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.6.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.6.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.6.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.6.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.7. เมื่อ choice เท่ากับ case : 6

18.4.2.7.1. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.7.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.7.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.7.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.7.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.8. เมื่อ choice เท่ากับ case : 7

18.4.2.8.1. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.8.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.8.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.8.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.8.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.9. เมื่อ choice เท่ากับ case : 8

18.4.2.9.1. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.9.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.9.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.9.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.9.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

18.4.2.10. เมื่อ choice เท่ากับ case : 9

18.4.2.10.1. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.10.2. เรียกใช้ฟังก์ชัน slideleft(size,a)

18.4.2.10.3. เรียกใช้ฟังก์ชัน slideup(size,a)

18.4.2.10.4. เรียกใช้ฟังก์ชัน sliderright(size,a)

18.4.2.10.5. เรียกใช้ฟังก์ชัน slideunder(size,a)

19. ประกาศฟังก์ชัน show_easy ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

19.1. รับพารามิเตอร์ int s[][3]

19.2. ประกาสตัวแปร i, j เป็นชนิดจำนวนเต็ม

19.3. ทำซ้ำโดยใช้ for โดยที่ i<3

19.3.1. แสดง _____ _____ _____ ออกทางหน้าจอ

19.3.2. แสดง | | | | ออกทางหน้าจอ

19.3.3. ทำซ้ำโดยใช้ for โดยที่ j<3

19.3.3.1. ถ้า s[i][j] != 9

19.3.3.1.1. แสดง | ออกทางหน้าจอ

19.3.3.1.2. แสดงค่า s[i][j]

19.3.3.2. มิฉะนั้น แสดง | ออกทางหน้าจอ

19.3.3.3. แสดง ออกทางหน้าจอ

19.3.3.4. j++

19.3.4. แสดง | ออกทางหน้าจอ

19.3.5. i++

19.4. แสดง |_____|_____|_____| ออกทางหน้าจอ

20. ประกาศฟังก์ชัน show_normal ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

20.1. รับพารามิเตอร์ int s[][4]

20.2. ประกาสตัวแปร i, j เป็นชนิดจำนวนเต็ม

20.3. ทำซ้ำโดยใช้ for โดยที่ i<4

20.3.1. แสดง _____ _____ _____ _____ออกทางหน้าจอ

20.3.2. แสดง | | | | |ออกทางหน้าจอ

20.3.3. ทำซ้ำโดยใช้ for โดยที่ j<4

20.3.3.1. ถ้า s[i][j] != 16

20.3.3.1.1. แสดง | ออกทางหน้าจอ

20.3.3.1.2. แสดงค่า s[i][j]

20.3.3.2. มิฉะนั้น แสดง | ออกทางหน้าจอ

20.3.3.3. แสดง ออกทางหน้าจอ

20.3.3.4. j++

20.3.4. แสดง | ออกทางหน้าจอ

20.3.5. i++

20.4. แสดง |_____|_____|_____|_____| ออกทางหน้าจอ

21. ประกาศฟังก์ชัน show_hard ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

21.1. รับพารามิเตอร์ int s[][5]

21.2. ประกาสตัวแปร i, j เป็นชนิดจำนวนเต็ม

21.3. ทำซ้ำโดยใช้ for โดยที่ i<5

21.3.1. แสดง _____ _____ _____ _____ _____ออกทางหน้าจอ

21.3.2. แสดง | | | | | |ออกทางหน้าจอ

21.3.3. ทำซ้ำโดยใช้ for โดยที่ j<5

21.3.3.1. ถ้า s[i][j] != 25

21.3.3.1.1. แสดง | ออกทางหน้าจอ

21.3.3.1.2. แสดงค่า s[i][j]

21.3.3.2. มิฉะนั้น แสดง | ออกทางหน้าจอ

21.3.3.3. แสดง ออกทางหน้าจอ

21.3.3.4. j++

21.3.4. แสดง | ออกทางหน้าจอ

21.3.5. i++

21.4. แสดง |_____|_____|_____|_____|_____| ออกทางหน้าจอ

22. ประกาศฟังก์ชัน slideright ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

22.1. รับพารามิเตอร์ int size,int data[][size]

22.2. ประกาสตัวแปร i, j, temp เป็นชนิดจำนวนเต็ม

22.3. ทำซ้ำโดยใช้ for โดยที่ i<size

22.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size

22.3.1.1. ถ้า s[i][j] == (size * size)

22.3.1.1.1. ถ้า j>0

1. กำหนด temp เท่ากับ data[i][j]
2. กำหนด data[i][j] เท่ากับ data[i][j-1]
3. กำหนด data[i][j-1] เท่ากับ temp

22.3.1.1.2. มิฉะนั้น แสดง *Can't do it ออกทางหน้าจอ

22.3.1.2. ข้อความสั่ง Break;

22.3.1.3. j++

22.3.2. i++

23. ประกาศฟังก์ชัน slideleft ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

23.1. รับพารามิเตอร์ int size,int data[][size]

23.2. ประกาสตัวแปร i, j, temp เป็นชนิดจำนวนเต็ม

23.3. ทำซ้ำโดยใช้ for โดยที่ i<size

23.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size

23.3.1.1. ถ้า s[i][j] == (size * size)

23.3.1.1.1. ถ้า j>(size-1)

1. กำหนด temp เท่ากับ data[i][j]
2. กำหนด data[i][j] เท่ากับ data[i][j+1]
3. กำหนด data[i][j+1] เท่ากับ temp

23.3.1.1.2. มิฉะนั้น แสดง *Can't do it ออกทางหน้าจอ

23.3.1.2. ข้อความสั่ง Break;

23.3.1.3. j++

23.3.2. i++

24. ประกาศฟังก์ชัน slideunder ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

24.1. รับพารามิเตอร์ int size,int data[][size]

24.2. ประกาสตัวแปร i, j, temp เป็นชนิดจำนวนเต็ม

24.3. ทำซ้ำโดยใช้ for โดยที่ i<size

24.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size

24.3.1.1. ถ้า s[i][j] == (size * size)

24.3.1.1.1. ถ้า i>0

1. กำหนด temp เท่ากับ data[i][j]
2. กำหนด data[i][j] เท่ากับ data[i-1][j]
3. กำหนด data[i-1][j] เท่ากับ temp

24.3.1.1.2. มิฉะนั้น แสดง *Can't do it ออกทางหน้าจอ

24.3.1.2. ข้อความสั่ง Break;

24.3.1.3. j++

24.3.2. i++

25. ประกาศฟังก์ชัน slideup ชนิด ไม่มีการคืนค่าและมีพารามิเตอร์

25.1. รับพารามิเตอร์ int size,int data[][size]

25.2. ประกาสตัวแปร i, j, temp เป็นชนิดจำนวนเต็ม

25.3. ทำซ้ำโดยใช้ for โดยที่ i<size

25.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size

25.3.1.1. ถ้า s[i][j] == (size * size)

25.3.1.1.1. ถ้า j>0

1. กำหนด temp เท่ากับ data[i][j]
2. กำหนด data[i][j] เท่ากับ data[i+1][j]

3. กำหนด data[i+1][j] เท่ากับ temp

25.3.1.1.2. มิฉะนั้น แสดง *Can't do it ออกทางหน้าจอ

25.3.1.2. ข้อความสั่ง return;

25.3.1.3. j++

25.3.2. i++

26. ประกาศฟังก์ชัน check ชนิด มีการคืนค่าเป็นชนิดจำนวนเต็มและมีพารามิเตอร์

26.1. รับพารามิเตอร์ int size,int a[][size], int b[][size]

26.2. ประกาสตัวแปร i, j, count=0 เป็นชนิดจำนวนเต็ม

26.3. ทำซ้ำโดยใช้ for โดยที่ i<size

26.3.1. ทำซ้ำโดยใช้ for โดยที่ j<size

26.3.1.1. ถ้า s[i][j] == (size * size)

26.3.1.1.1. กำหนด count++

26.3.1.2. j++

26.3.2. i++

26.4. คืนค่า count

27. ประกาศฟังก์ชัน process_time ชนิด มีการคืนค่าเป็นชนิดเลขทศนิยมและมีพารามิเตอร์

27.1. รับพารามิเตอร์ double a

27.2. ประกาศตัวแปร min เป็นชนิดเลขทศนิยม

27.3. ถ้า a > 60 || a == 60

27.3.1. กำหนด min = a / 60

27.3.2. คืนค่า min

27.4. มิฉะนั้น คืนค่า a

28. ประกาศฟังก์ชัน playgame ชนิด มีการคืนค่าเป็นชนิดตัวอักขระและมีพารามิเตอร์

28.1. ข้อความคำสั่ง time_t rawtime;

28.2. ข้อความคำสั่ง time (&rawtime);

28.3. ข้อความคำสั่ง time_t start,end;

28.4. ประกาศตัวแปร time_end เป็นชนิดเลขทศนิยม(double)

28.5. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้

28.6. กำหนด slide = 0

28.7. ประกาศตัวแปร i,j,count, ch_correct เป็นชนิดจำนวนเต็ม

28.8. ประกาศตัวแปร choice, ck_ready เป็นชนิดตัวอักขระ

28.9. ประกาศตัวแปรแถวลำดับ ck_arr สองมิติ เป็นชนิดจำนวนเต็ม ที่มีดัชนีเท่ากับ size และ size

28.10. ทำซ้ำโดยใช้ for โดยที่ i<size

28.10.1. ทำซ้ำโดยใช้ for โดยที่ j<size

28.10.1.1. กำหนด ck_arr[i][j] เท่ากับ arr2[i][j];

28.10.1.2. j++

28.10.2. i++

28.11. ข้อความคำสั่ง system("clear");

28.12. เรียกใช้ฟังก์ชัน enter_name(name);

28.13. ข้อความคำสั่ง system("clear");

28.14. เรียกใช้ฟังก์ชัน rd_check(size,arr1,ck_arr);

28.15. ข้อความคำสั่ง system("clear");

28.16. กำหนด ck_ready เท่ากับค่าที่คืนมาจากฟังก์ชัน ready()

28.17. กำหนด count = 0

28.18. กำหนด ck_correct เท่ากับ size * size

28.19. ถ้า ck_ready != 'a'

28.19.1. ข้อความคำสั่ง system("clear");

28.19.2. ข้อความคำสั่ง time (&start);

28.19.3. ทำซ้ำโดยใช้ do...while โดยมีนิพจ์ (choice != 'x' && choice != 'X')

28.19.3.1. ถ้า size == 3

28.19.3.1.1. เรียกใช้ฟังก์ชัน show_easy(arr1)

28.19.3.2. ถ้า size == 4

28.19.3.2.1. เรียกใช้ฟังก์ชัน show_normal(arr1)

28.19.3.3. ถ้า size == 5

28.19.3.3.1. เรียกใช้ฟังก์ชัน show_hard(arr1)

28.19.3.4. แสดงค่า count, ck_correct, slide ออกมาทางหน้าจอ

28.19.3.5. แสดง _____ ออกมาทางหน้าจอ

28.19.3.6. แสดง %s ผ่านตัวแปร level ออกมาทางหน้าจอ

28.19.3.7. แสดง _____ ออกมาทางหน้าจอ

28.19.3.8. แสดง *** Please full screen *** ออกมาทางหน้าจอ

28.19.3.9. แสดง Give up press X ออกมาทางหน้าจอ

28.19.3.10. แสดง Finished Press any key to continues... ออกมาทางหน้าจอ

28.19.3.11. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ choice

28.19.3.12. กำหนด count เท่ากลับค่าที่คืนมาจากฟังก์ชัน check(size,arr1,ck_arr)

28.19.3.13. ข้อความคำสั่ง system("clear");

28.19.3.14. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ choice

28.19.3.15. เข้า Switch case

28.19.3.15.1. เมื่อ choice เท่ากับ case : ‘w’

1. เรื่องใช้ฟังก์ชัน slideup(size,arr1)
2. กำหนด slide++

28.19.3.15.2. เมื่อ choice เท่ากับ case : ‘d’

1. เรื่องใช้ฟังก์ชัน sliderright(size,arr1)
2. กำหนด slide++

28.19.3.15.3. เมื่อ choice เท่ากับ case : ‘s’

1. เรื่องใช้ฟังก์ชัน slideunder(size,arr1)
2. กำหนด slide++

28.19.3.15.4. เมื่อ choice เท่ากับ case : ‘a’

1. เรื่องใช้ฟังก์ชัน slideleft(size,arr1)
2. กำหนด slide++

28.19.3.15.5. เมื่อ choice เท่ากับ case : ‘W’

1. เรื่องใช้ฟังก์ชัน slideup(size,arr1)
2. กำหนด slide++

28.19.3.15.6. เมื่อ choice เท่ากับ case : ‘D’

1. เรื่องใช้ฟังก์ชัน sliderright(size,arr1)
2. กำหนด slide++

28.19.3.15.7. เมื่อ choice เท่ากับ case : ‘S’

1. เรื่องใช้ฟังก์ชัน slideunder(size,arr1)
2. กำหนด slide++

28.19.3.15.8. เมื่อ choice เท่ากับ case : ‘A’

1. เรื่องใช้ฟังก์ชัน slideleft(size,arr1)

2. กำหนด slide++

28.19.3.15.9. ถ้า count == ck_correct

1. ข้อความคำสั่ง system("clear");
2. ข้อความคำสั่ง time (&end);
3. กำหนด time_end เท่ากับ difftime (end,start)
4. เปิดไฟล์ “Socre_data.txt” โหมด a
5. กำหนด time_show เท่ากับค่าที่คืนมาจากฟังก์ชัน process_time(time_end)
6. ถ้า time_end < 60 && time_end > 0
   6.1. แสดง Your time is %5.2f second ผ่านตัวแปร time_end ออกมาทางหน้าจอ
   6.2. แสดง Times of slide is %d times ผ่านตัวแปร slide ออกมาทางหน้าจอ
   6.3. เขียน > Level %s จากตัวแปร level ลงไฟล์
   6.4. เขียน %s | %d times | %5.2f seconds | %s ผ่านตัวแปร name,slide,time_show,ctime(&rawtime) ลงไฟล์
7. มิฉะนั้นแสดง แสดง Your time is %5.2f second ผ่านตัวแปร time_end ออกมาทางหน้าจอ
8. แสดง Times of slide is %d times จากตัวแปร slide ออกมาทางหน้าจอ
9. เขียน > Level %s จากตัวแปร level ลงไฟล์
10. เขียน %s | %d times | %5.2f seconds | %s ผ่านตัวแปร name,slide,time_show,ctime(&rawtime) ลงไฟล์
11. ปิดไฟล์ in

28.19.3.15.10. แสดง Press X to exit ออกมาทางหน้าจอ

28.19.3.15.11. ทำซ้ำโดยใช้ while โดยที่

choice != 'x' && choice != 'X'

28.19.3.15.12. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ choice

28.19.4. ข้อความสั่ง time (&end);

29. ประกาศฟังก์ชัน ready ชนิดมีการคืนค่าเป็นชนิดตัวอักขระและมีพารามิเตอร์

29.1. ประกาศตัวแปร ans, text เป็นชนิดตัวอักขระ

29.2. ประกาศตัวแปร result เป็นชนิดจำนวนเต็ม
29.3. ประกาศตัวแปร in เป็นชนิด FILE ประเภทตัวชี้
29.4. เปิดไฟล์ ready.txt โหมด r
29.5. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
29.6. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
29.7. ทำซ้ำโดยใช้ while โดยที่ result != EOF
29.7.1. แสดงค่า text ออกทางหน้าจอ
29.7.2. กำหนด result เท่ากับ การรับค่าจากการอ่านไฟล์
29.7.3. อ่านค่าจากไฟล์ มาเก็บไว้ที่ตัวแปร text
29.8. ปิดไฟล์ in
29.8.1. ทำซ้ำโดยใช้ strcmp (ans,"yes") != 0 && strcmp (ans,"no") != 0
29.8.1.1. แสดง Are you ready? (yes/no) : ออกทางหน้าจอ
29.8.1.2. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ ans
29.9. ถ้า strcmp(ans,"yes") != 0
29.9.1. คืนค่า ‘a’
29.10. มิฉะนั้น คืนค่า ‘b’
30. ประกาศตัวแปร choice เป็นชนิดตัวอักขระ
31. ทำซ้ำโดยใช้ do...while โดยมีนิพจ์ (choice != ‘5’)
31.1. ข้อความคำสั่ง system(“clear”);
31.2. เรียกใช้ฟังก์ชัน puzzle
31.3. เรียกใช้ฟังก์ชัน main_menu
31.4. ข้อความคำสั่ง time_t rawtime;
31.5. ข้อความคำสั่ง time (&rawtime)
31.6. แสดงเวลา (rawtime) ออกทางหน้าจอ
31.7. รับค่าทางแป้นพิมพ์มาเก็บไว้ที่ choice
31.8. เข้า Switch case
31.8.1. เมื่อ choice เท่ากับ case : ‘1’
31.8.1.1. ข้อความคำสั่ง system(“clear”);
31.8.1.2. เรียกใช้ฟังก์ชัน play
31.8.2. เมื่อ choice เท่ากับ case : ‘2’
31.8.2.1. ข้อความคำสั่ง system(“clear”);

31.8.2.2. เรียกใช้ฟังก์ชัน how2play

31.8.3. เมื่อ choice เท่ากับ case : ‘3’

31.8.3.1. ข้อความคำสั่ง system(“clear”);

31.8.3.2. เรียกใช้ฟังก์ชัน high_score

31.8.4. เมื่อ choice เท่ากับ case : ‘4’

31.8.4.1. ข้อความคำสั่ง system(“clear”);

31.8.4.2. เรียกใช้ฟังก์ชัน credit

31.8.5. เมื่อ case default

31.8.5.1. แสดง Please again ออกทางหน้าจอ

32. ข้อความคำสั่ง system(“clear”);

33. เรียกใช้ฟังก์ชัน mickey

34. จบการทำงาน

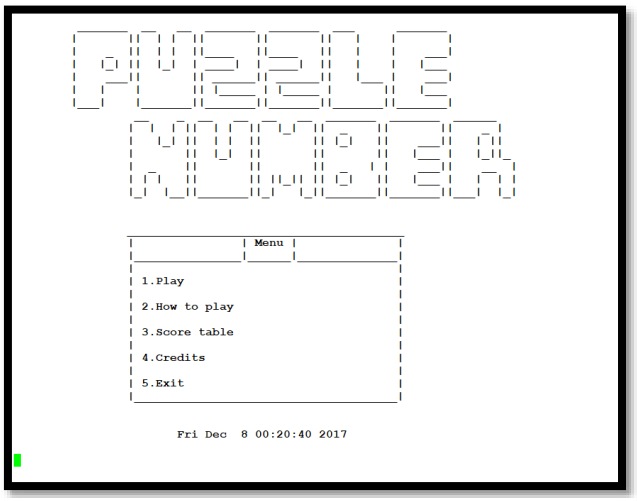
บทที่ 4

ผลการดำเนินงานโครงงาน

4.1 การทำงานของโปรแกรม

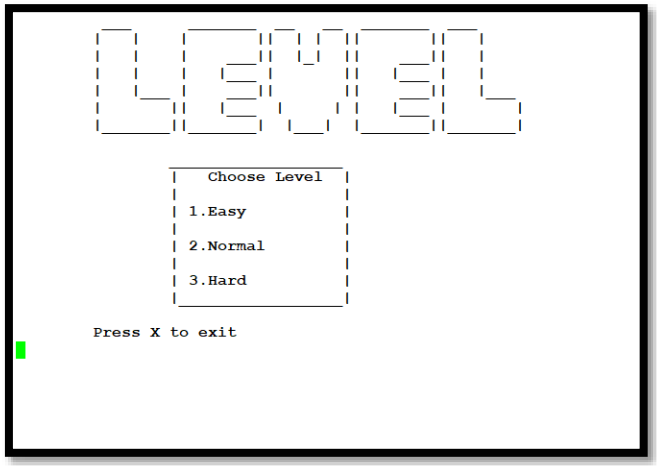
การทำงานของโปรแกรมเกมเรียงตัวเลข (Puzzle number) จะแบ่งการทำงานเป็น 5 เมนู และจะมีการทำงานย่อยดังต่อไปนี้

1. หน้าแรก



ภาพที่ 4-1 หน้าแรก

2. เมนู Play

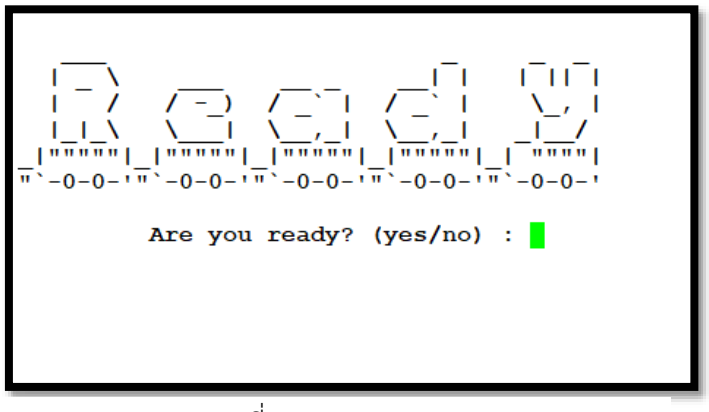


ภาพที่ 4-2 เมนู Play

3. เมนู ใส่ชื่อผู้เล่น

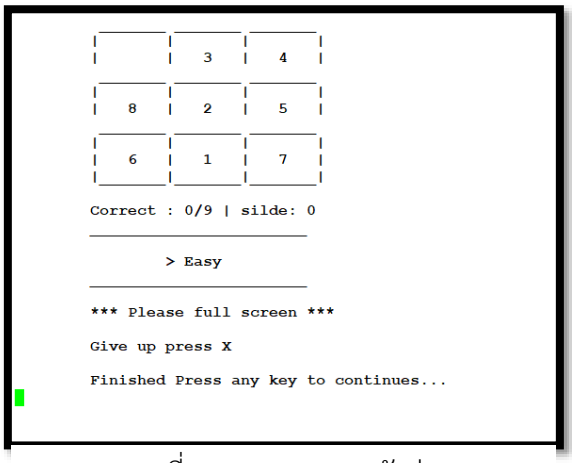
ภาพที่ 4-3 เมนู ใส่ชื่อผู้เล่น

4. เมนู Ready?



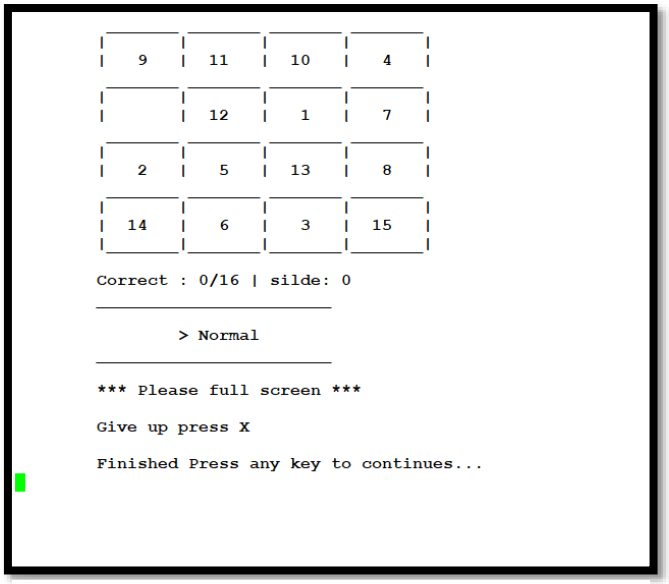
ภาพที่ 4-4 เมนู Ready?

5. แสดงเกมระดับง่าย



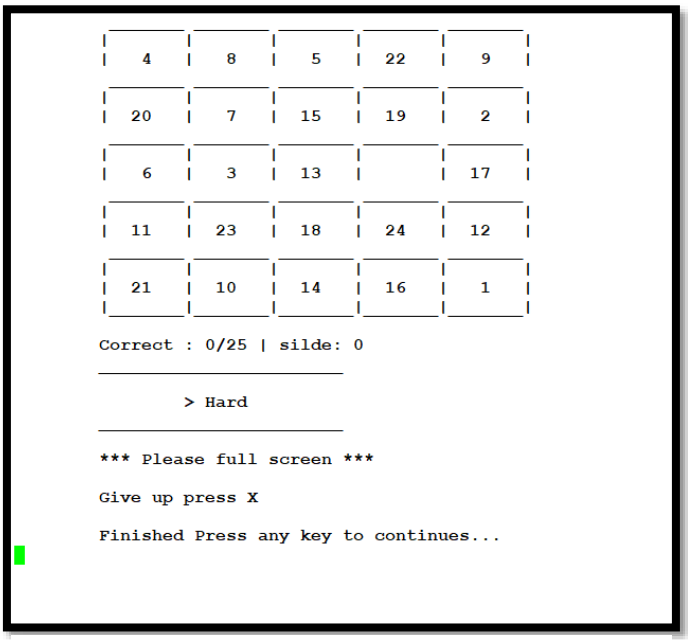
ภาพที่ 4-5 แสดงเกมระดับง่าย

6. แสดงเกมระดับปานกลาง



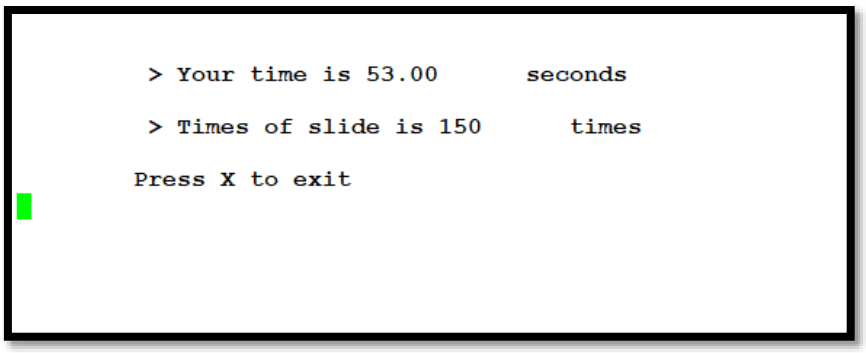
ภาพที่ 4-6 แสดงเกมระดับปานกลาง

7. แสดงเกมระดับยาก



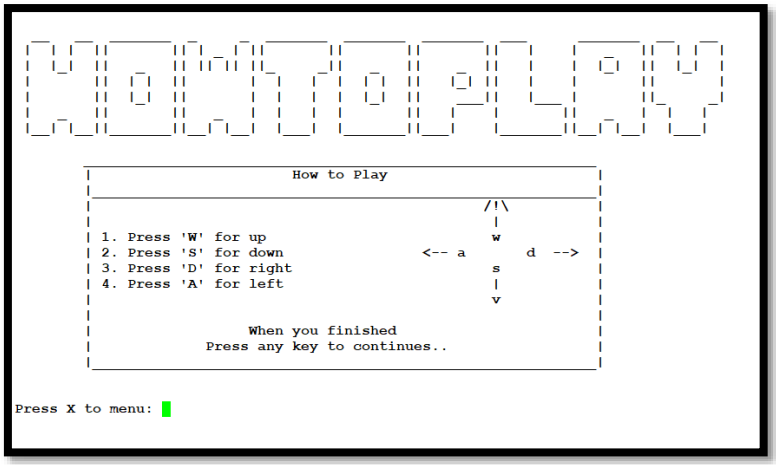
ภาพที่ 4-7 แสดงเกมระดับยาก

8. เมื่อชนะเกม



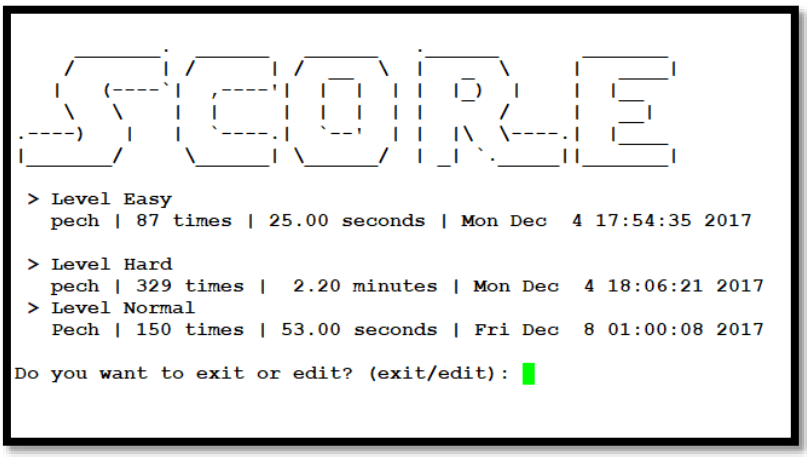
ภาพที่ 4-8 เมื่อชนะเกม

9. เมนู วิธีการเล่น



ภาพที่ 4-9 เมนู วิธีการเล่น

10. เมนู ตารางคะแนน

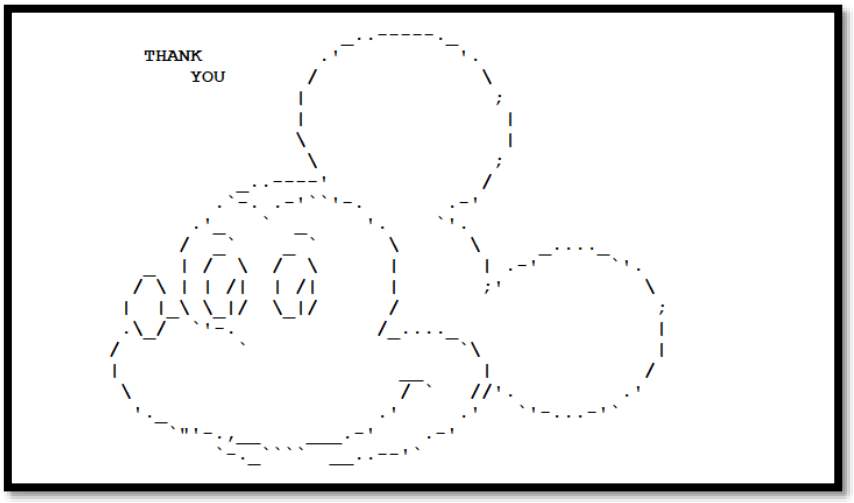


ภาพที่ 4-10 เมนู ตารางคะแนน

11. เมนู เครดิตผู้จัดทำ

ภาพที่ 4-11 เมนู เครดิตผู้จัดทำ

12. สิ้นสุดการทำงาน



ภาพที่ 4-12 สิ้นสุดการทำงาน

บทที่ 5

สรุปและวิเคราะห์ผลการดำเนินการ

โปรแกรมเกมเรียงเลข (Puzzle number) นี้มีวัตถุประสงค์เพื่อสร้างความสนุกสนานและสามารถฝึกทักษะให้กับผู้เล่นไปในตัว

5.1 ประโยชน์ที่ได้รับ

1. เพื่อผ่อนคลายจากการเรียน และการทำงาน
2. เพื่อฝึกทักษะการจัดเรียงตัวเลขให้กับนิสิต สาขาวิชาวิศวกรรมซอฟต์แวร์

5.2 ปัญหาและอุปสรรคในการจัดทำโครงงาน

1. เวลาในการจัดทำน้อยเกินไป
2. ความรู้ความสามารถในด้านภาษาซียังไม่แน่นพอ
3. ปัญหาการเชื่อมต่อเครือข่าย Dekdee.buu.ac.th

5.3 ข้อจำกัดของโปรแกรม

1. ต้องใช้แป้นพิมพ์ภาษาอังกฤษเท่านั้น เพื่อป้องกันการผิดพลาด
2. ปรับตัวอักษรได้ไม่ใหญ่มาก จะทำให้ภาพล้นจอ

บรรณานุกรม

[1] ประวัติความเป็นมาของภาษาซี.(ออนไลน์). อ้างอิงจาก https://sites.google.com/site/bbmm2553/ สืบค้นเมื่อวันที่ 2 ธันวาคม 2560.

[2] ตัวแปรชุดของอักขระ String.(ออนไลน์). อ้างอิงจาก http://slideplayer.in.th/slide/3185910/ สืบค้นเมื่อวันที่ 2 ธันวาคม 2560.

[3] ตัวแปรแบบโครงสร้าง.(ออนไลน์). อ้างอิงจาก http://slideplayer.in.th/slide/2114648 สืบค้นเมื่อวันที่ 3 ธันวาคม 2560.

[4] Puzzle คืออะไร.(ออนไลน์). อ้างอิงจาก https://www.raisegeniusschool.com/puzzle/ สืบค้นเมื่อวันที่ 3 ธันวาคม 2560.

[5] นวลศรี เด่นวัฒนา (2555) หลักการและวิธีการโปรแกรมสำหรับวิศวกรรมซอฟต์แวร์ด้วยภาษาซี สาขาวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยบูรพา

ภาคผนวก

ภาคผนวก ก

Source Code

ส่วนของไฟล์ Puzzle.h

```
#include<stdio.h> //declare header
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include <unistd.h> //search form Internet
#include <termios.h> //search form Internet
typedef struct{
    char name[10];
    char level[5];
    float time;
    int slide;
    int size;
}PLAYER;
void clear_data(); //declare function
void credit();
void enter_name(char name[]);
void high_score();
void how2play();
void main_menu();
void mickey();
void play();
void puzzle();
void rd_puzzle(int size,int a[][size]);
void rd_check(int size,int a[][size],int b[][size]);
void show_easy(int s[][3]);
void show_normal(int s[][4]);
```

```
void show_normal(int s[][4]);
void show_hard(int s[][5]);
void slideright(int size,int data[][size]);
void slideleft(int size,int data[][size]);
void slideunder(int size,int data[][size]);
void slideup(int size,int data[][size]);
int check( int size,int a[][size], int b[][size]);
float process_time(double a);
char playgame(int size,int arr1[][size],int arr2[][size],int slide,char name[],char level[],float time_show);
char getch();
char ready();
```

ส่วนของไฟล์ Puzzlelib.c

```c
#include"puzzle.h"
void main_menu()
{
    FILE *in;                                       //declare for open file
    int result;                                     //declare for check EOF!
    char text;                                      //declare for keep charactor form file
    in = fopen("main_menu.txt","r");                //open file "read"
    result = fscanf(in,"%c",&text);                 //check EOF! and keep charactor
    while (result != EOF){                          //condition (while) for check EOF!
        printf("%c",text);                          //output data in file
        result = fscanf(in,"%c",&text);             //check EOF! and keep charactor
    }
    printf("\n");
    fclose(in);                                     //closr file
} //main_menu
void play()
{
    PLAYER easy,normal,hard;
    easy.size = 3;
    strcpy(easy.level,"Easy");
    normal.size = 4;
    strcpy(normal.level,"Normal");
    hard.size = 5;
    strcpy(hard.level,"Hard");
    int arr_easy[3][3]  =  {{ 1 , 2 , 3 },              //declare arr for everthing
                           { 4 , 5 , 6 },
                           { 7 , 8 , 9 }};
    int arr_normal[4][4] = {{ 1 , 2 , 3 , 4 },
                           { 5 , 6 , 7 , 8 },
                           { 9 ,10 ,11 ,12 },
                           {13 ,14 ,15 ,16 }};
```

```c
    int arr_hard[5][5] = {{ 1 , 2 , 3 , 4 , 5},
                          { 6 , 7 , 8 , 9 ,10},
                          {11 ,12 ,13 ,14 ,15},
                          {16 ,17 ,18 ,19 ,20},
                          {21 ,22 ,23 ,24 ,25}};
    char choice,ck_exit;                                //declare for choose choice
    do{                                                 //condition (do_while) for menu
        system("clear");                                        //clear screen
        FILE *in;                                       //declare for open file
        int result;
        char text;                                      //declare for keep charector
        in = fopen("level.txt","r");
        result = fscanf(in,"%c",&text);
        while (result != EOF){
        printf("%c",text);
        result = fscanf(in,"%c",&text);
        }
        fclose(in);                                     //close file
        printf("\n\t\t\t_________________\n");          //show menu level
        printf("\t\t|   Choose Level  |\n");
        printf("\t\t|                 |\n");
        printf("\t\t| 1.Easy          |\n");
        printf("\t\t|                 |\n");
        printf("\t\t| 2.Normal        |\n");
        printf("\t\t|                 |\n");
        printf("\t\t| 3.Hard          |\n");
        printf("\t\t|_________________|\n");
        printf("\n\tPress X to exit\n");
        choice = getch();                               //choose choice
        switch(choice){                                 //condition (switch case) for choose choice
            case '1':                                   //calling function
                do{
                    ck_exit = playgame(easy.size,arr_easy,arr_easy,easy.slide,easy.name,easy.level,easy.time);
                }while(ck_exit != 'e'); break;
            case '2':
                do{
                    ck_exit = playgame(normal.size,arr_normal,arr_normal,normal.slide,normal.name,normal.level,normal.time);
                }while(ck_exit != 'e'); break;
            case '3':
                do{
                    ck_exit = playgame(hard.size,arr_hard,arr_hard,hard.slide,hard.name,hard.level,hard.time);
```

```
76                      ck_exit = playgame(hard.size,arr_hard,arr_hard,hard.slide,hard.name,hard.level,hard.time);
77                  }while(ck_exit != 'e'); break;
78              default : printf("Please again...\n");      //another case
79          }
80      }while(choice != 'x' && choice != 'X');
81  } //play
82  void how2play()
83  {
84      char ch,text;               //declare "ch" for choice exit //declare "text" for keep charactor form file
85      FILE *in;                                           //declare for open file
86      int result;                                         //declare for check EOF!
87      in = fopen("how2play.txt","r");                     //openfile "read"
88      result = fscanf(in,"%c",&text);                     //check EOF! and keep charector
89      while (result != EOF){                              //condition (while) for check EOF!
90          printf("%c",text);                              //output data in file
91          result = fscanf(in,"%c",&text);                 //condition (while) for check EOF!
92      }
93      printf("\n");
94      fclose(in);                                         //close file
95      printf("\n\nPress X to menu: ");
96      while(ch != 'x' && ch != 'X'){                      //condition (while) for exit
97       ch = getch();
98      }
99  } //how2play
100 void high_score()
101 {
102     char ch,text;             //declare "ch" for choice exit //declare "text" for keep charactor form file
103     int result;                                         //declare for check EOF!
104     FILE * in;                                          //declare for open file
105     system("clear");                                            //clear screen
106         FILE *in2;                                      //declare for open file
107         int result2;
108         char text2;                                     //declare for keep charector
109         in2 = fopen("scoretable.txt","r");
110         result = fscanf(in2,"%c",&text2);
111         while (result2 != EOF){
112         printf("%c",text2);
113         result2 = fscanf(in2,"%c",&text2);
114         }
115         fclose(in2);                                        //close file
116     in = fopen("Score_data.txt","r");                   //open file "read"
```

```
116     in = fopen("Score_data.txt","r");                   //open file "read"
117     result = fscanf(in,"%c",&text);                     //check EOF! and keep charector
118     if(result != EOF){                                  //condition (if) for check data
119         while (result != EOF){                          //if have data will show
120         printf("%c",text);
121         result = fscanf(in,"%c",&text);
122         }
123         clear_data();
124     }else {                                             //else show "No data..."
125         printf("No data...\n");
126         printf("\n\nPress X to menu: ");
127         while(ch != 'x' && ch != 'X'){                  //condition (while) for exit
128             ch = getch();
129         }
130     }
131     fclose(in); //close file
132 } //high_score
133 void credit()
134 {
135     char ch,text;             //declare "ch" for choice exit //declare "text" for keep charactor form file
136     FILE *in;                                           //declare for open file
137     int result;                                         //declare for check EOF!
138     in = fopen("credit.txt","r");                       //open file "read"
139     result = fscanf(in,"%c",&text);                     //check file and keep charector
140     while (result != EOF){                              //condition (while) for check EOF!
141         printf("%c",text);                              //output data in file
142         result = fscanf(in,"%c",&text);                 //condition (while) for check EOF!
143     }
144     printf("\n");
145     fclose(in);                                         //close file
146     printf("\n\nPress X to menu: ");
147     while(ch != 'x' && ch != 'X'){                      //condition (while) for exit
148      ch = getch();
149     }
150 } //credit
151 void show_easy(int s[][3])
152 {
153     int i,j;                                            //declare for condition (for loop)
154     for(i=0;i<3;i++)                                    //condition (for) for show table of game easy EZ!
155     {
156         printf("\t _______ _______ _______ \n");
```

```
156         printf("\t _______ _______ _______ \n");
157         printf("\t|       |       |       |\n");
158         for(j=0;j<3;j++){
159             if(s[i][j]!=9){                             //condition (if) for check blank
160                 printf("\t|");
161                 printf("%4d",s[i][j]);                  //if != 9 show number
162             }else{
163                     printf("\t|");
164                     printf("    ");                     //else show blank
165             }
166         }
167         printf("\t|\n");
168     }
169         printf("\t|_______|_______|_______|\n");
170 } //show_easy
171 void show_normal(int s[][4])
172 {
173     int i,j;                                            //declare for condition (for loop)
174     for(i=0;i<4;i++)                                    //condition (for) for show table of game normal
175     {
176         printf("\t _______ _______ _______ _______ \n");
177         printf("\t|       |       |       |       |\n");
178         for(j=0;j<4;j++){
179             if(s[i][j]!=16){                            //condition (if) for check blank
180                 printf("\t|");
181                 printf("%4d",s[i][j]);                  //if != 16 show number
182             }else{
183                 printf("\t|");
184                 printf("    ");                         //else show blank
185             }
186         }
187         printf("\t|\n");
188     }
189         printf("\t|_______|_______|_______|_______|\n");
190 } //show_normal
191 void show_hard(int s[][5])
192 {
193     int i,j;                                            //declare for condition (for loop)
194     for(i=0;i<5;i++)                                    //condition (for) for show table of game hard
195     {
196         printf("\t _______ _______ _______ _______ _______\n");
```

```
197         printf("\t|       |       |       |       |       |\n");
198         for(j=0;j<5;j++){
199             if(s[i][j]!=25){                            //condition (if) for check blank
200                 printf("\t|");
201                 printf("%4d",s[i][j]);                  //if != 25 show number
202             }else{
203                 printf("\t|");
204                 printf("    ");                         //else show blank
205             }
206         }
207         printf("\t|\n");
208         }
209         printf("\t|_______|_______|_______|_______|_______|\n");
210 } //show_hard
211 void slideright(int size,int data[][size])
212 {
213     int i,j,temp;                                       //declare for function slide
214     for(i=0;i<size;i++){
215         for(j=0;j<size;j++){
216             if(data[i][j] == (size * size) ){           //condition (if) for this number is blank (9-16-25)
217                 if(j>0){                                //if col > 0
218                     temp=data[i][j];                    //blank --> temp
219                     data[i][j]=data[i][j-1];            //replace blank with col -1
220                     data[i][j-1]=temp;                  //blank --> col -1
221                 }else{                                  //when can't slide
222                     printf("\n\t  \a*Can't do it   \n\n");
223                 }
224             break;
225             }
226         }
227     }
228 } //slideright
229 void slideleft(int size,int data[][size])
230 {
231     int i,j,temp;                                       //delare for function slide
232     for(i=0;i<size;i++){
233         for(j=0;j<size;j++){
234             if(data[i][j]==(size * size)  ){            //condition (if) for this number is blank (9-16-25)
235                 if(j<(size-1)){                         //if col < (column of table)
236                     temp=data[i][j];                    //blamk --> temp
237                     data[i][j]=data[i][j+1];            //replace blank with col +1
```

```
                    data[i][j]=data[i][j+1];            //replace blank with col +1
                    data[i][j+1]=temp;                  //blank --> col+1
                }else{                                  //when can't slide
                    printf("\n\t  \a*Can't do it   \n\n");
                }
            break;
            }
        }
    }
} //slideleft
void slideunder(int size,int data[][size])
{
    int i,j,temp;                                       //declare for function slide
    for(i=0;i<size;i++){
        for(j=0;j<size;j++){
            if(data[i][j]==(size * size)  ){            //condition (if) for this number is blank (9-16-25)
                if(i>0){                                //if row > o
                    temp=data[i][j];                    //blank --> temp
                    data[i][j]=data[i-1][j];            //replace blank with row -1
                    data[i-1][j]=temp;                  //blank --> row -1
                }else{                                  //when can't slide
                    printf("\n\t  \a*Can't do it   \n\n");
                }
            break;
            }
        }
    }
} //slideunder
void slideup(int size,int data[][size])
{
    int i,j,temp;                                       //declare for function slide
    for(i=0;i<size;i++){
        for(j=0;j<size;j++){
            if(data[i][j]==(size * size) ){             //condition (if) for this number is blank (9-16-25)
                if(i<(size-1)){                         //if row < (row of table)
                    temp=data[i][j];                    //blank --> temp
                    data[i][j]=data[i+1][j];            //replace blank with row +1
                    data[i+1][j]=temp;                  //blank --> row+1
                }else{                                  //when can't slide
                    printf("\n\t  \a*Can't do it   \n\n");
                }
```

```
                }
            return;                                     //for else last row will go to first (error)
            }
        }
    }
} //slideup
char playgame(int size,int arr1[][size],int arr2[][size],int slide,char name[],char level[],float time_show)
{
    time_t rawtime;                                     //(time) search form Internet
    time (&rawtime);
    double time_end;
    time_t start,end;
    FILE * In;                                          //declare for open file
    slide = 0;                                          //set slide = o
    int count,ck_correct;                                  //declare for check amount of correct
    char choice,ck_ready;                               //declare "choice" for play or exit  "ck_ready" for exit
    int i,j,ck_arr[size][size];
    for (i=0;i<size;i++){
        for(j=0;j<size;j++){
            ck_arr[i][j] = arr2[i][j];
        }
    }
    system("clear");                                                //clear screen
    enter_name(name);                                   //function enter name
    system("clear");
    rd_check(size,arr1,ck_arr);                         //calling function for random table
    system("clear");
    ck_ready = ready();                                 //calling function for ready!!
    count = 0;
    ck_correct = (size * size);
    if(ck_ready != 'a'){
    system("clear");
    time (&start);                                      //start timer
    do{                                                 //condition (do_while) for play game
        if (size == 3){
            show_easy(arr1);                                //calling function for show table
        }else if (size == 4){
            show_normal(arr1);
        }else if (size == 5){
            show_hard(arr1);
        }
```

```
317         }
318         printf("\n\tCorrect : %d/%d | silde: %d\n",count,ck_correct,slide); //show amount of channal correct and slide
319         printf("\t_________________________\n");
320         printf("\t                         \n");
321         printf("\t        > %s             \n",level);
322         printf("\t_________________________\n\n");
323         printf("\t*** Please full screen ***\n\n");
324         printf("\tGive up press X \n\n");
325         printf("\tFinished Press any key to continues...\n");
326         choice = getch();
327         count = check(size,arr1,ck_arr);                //check amount of channal correct
328         system("clear");
329         switch(choice){                                 //codition (switch case) for slide number and exit
330             case 'w': slideup(size,arr1); slide++; break;
331             case 'd': slideright(size,arr1); slide++; break;
332             case 's': slideunder(size,arr1); slide++; break;
333             case 'a': slideleft(size,arr1); slide++; break;
334             case 'W': slideup(size,arr1); slide++; break;
335             case 'D': slideright(size,arr1); slide++; break;
336             case 'S': slideunder(size,arr1); slide++; break;
337             case 'A': slideleft(size,arr1); slide++; break;
338         }
339         if(count == ck_correct){                                //conditiom (if) for check if when you finish
340             system("clear");
341             time (&end);
342                     time_end = difftime (end,start);
343                     In = fopen("Score_data.txt","a"); //open file "write again"
344                     time_show = process_time(time_end); //set easy.time form function process_time
345             if(time_end < 60 && time_end > 0){          //condition (if) for check "second" and "minutes"
346                 printf("\n\n\t > Your time is %5.2f      seconds \n\n",time_end);
347                 printf("\t > Times of slide is %d      times \n",slide);
348                 fprintf(In," > Level %s \n",level);
349                 fprintf(In,"   %s | %d times | %5.2f seconds | %s\n",name,slide,time_show,ctime(&rawtime));
350             }else {
351                 printf("\n\n\t > Your time is %5.2f      minutes \n\n",time_end/60);
352                 printf("\t > Times of slide is %d      times \n",slide);
353                 fprintf(In," > Level %s \n",level);
354                 fprintf(In,"   %s | %d times | %5.2f minutes | %s\n",name,slide,time_show,ctime(&rawtime));
355             }
356                 fclose(In);                             //close file
357                 printf("\n\tPress X to exit\n");
```

```
357                 printf("\n\tPress X to exit\n");
358                 while(choice != 'x' && choice != 'X'){  //condition (while) for exit
359                     choice = getch();}
360             }
361     }while(choice != 'x' && choice != 'X');
362     time (&end);                                        //stop timer
363     }
364     return 'e';                                         //return for exit
365 } //playname
366 float process_time(double a)                            //receviev time form game ("a" = second)
367 {
368         float min;                                      //declare for convert second to minutes
369         if (a > 60 || a == 60){                         //condition (if) for check second or mintutes
370             min = a / 60;                               //if "a" > 60 or "a" = 60 set min = a / 60
371             return min;                                 //return min (minutes)
372         } else {
373             return a;                                   //return a (second)
374         }
375 } //process_time
376 void enter_name(char name[])                            //receviev array from game (name == player[] in game)
377 {
378     FILE *in;                                           //declare for open file
379     int result;
380     char text;                                          //declare for keep charector
381     in = fopen("entername.txt","r");
382     result = fscanf(in,"%c",&text);
383     while (result != EOF){
384         printf("%c",text);
385         result = fscanf(in,"%c",&text);
386     }
387     fclose(in);                                         //close file
388     printf("\n\t++ Please enter your name(10 char):  ");
389     scanf("%s",name);
390 } //enter_name
391 int check(int size,int a[][size], int b[][size])
392 {
393     int i,j,count=0;                                    //declare "count" for check table
394     for(i=0;i<size;i++){
395         for(j=0;j<size;j++){
396             if (a[i][j] == b[i][j]){                    //condition (if) for check table
397                 count++;                                //if it match count +1
```

```
397                 count++;                                    //if it match count +1
398             }
399         }
400     }
401         return count;                                       //return count to game
402 } //check
403 char ready()                                                //use this for wait function random and return for exit loop
404 {
405     FILE *in;                                               //declare for open file
406     int result;                                             //declare for check EOF!
407     char text,ans[3] = "111";                               //declare "text" for show text art "ans" for check exit
408     in = fopen("ready.txt","r");                            //open file "read"
409     result = fscanf(in,"%c",&text);                         //check EOF! and keep charector
410     while (result != EOF){                                  //condition (while) for check EOF!
411         printf("%c",text);                                  //show charector
412         result = fscanf(in,"%c",&text);                     //check EOF! and keep charector
413     }
414     printf("\n");
415     fclose(in);                                             //close file
416     while(strcmp(ans,"yes") != 0 && strcmp(ans,"no") != 0){
417         printf("\t Are you ready? (yes/no) : ");
418         scanf("%s",ans);
419     }
420     if(strcmp(ans,"yes") != 0){                         //condition (if) for check "ans"
421         return 'a';
422     } else return 'b';
423 }
424 void rd_puzzle(int size,int a[][size])
425 {
426     int random,i;                                           //declare "random" for random choice
427     srand(time(NULL));                                      //function random
428     for(i=0;i<50;i++){                                      //condition (for loop) for random
429         random = rand() % 9;                                //random 0 - 9
430         switch(random){                                     //condition (switch case) for random
431             case 0 : slideright(size,a);                    //switch case random slide
432                 slideleft(size,a);
433                 slideup(size,a);
434                 slideright(size,a);
435                 slideunder(size,a); break;
436             case 1 : slideleft(size,a);                     //switch case random slide
437                 slideleft(size,a);
```

```
437                 slideleft(size,a);
438                 slideup(size,a);
439                 slideright(size,a);
440                 slideunder(size,a); break;
441             case 2 : slideup(size,a);                       //switch case random slide
442                 slideleft(size,a);
443                 slideup(size,a);
444                 slideright(size,a);
445                 slideunder(size,a); break;
446             case 3 : slideunder(size,a);                    //switch case random slide
447                 slideleft(size,a);
448                 slideup(size,a);
449                 slideright(size,a);
450                 slideunder(size,a); break;
451             case 4 : slideright(size,a);                    //switch case random slide
452                 slideleft(size,a);
453                 slideup(size,a);
454                 slideright(size,a);
455                 slideunder(size,a); break;
456             case 5 : slideup(size,a);                       //switch case random slide
457                 slideleft(size,a);
458                 slideup(size,a);
459                 slideright(size,a);
460                 slideunder(size,a); break;
461             case 6 : slideunder(size,a);                    //switch case random slide
462                 slideleft(size,a);
463                 slideup(size,a);
464                 slideright(size,a);
465                 slideunder(size,a); break;
466             case 7 : slideleft(size,a);                     //switch case random slide
467                 slideleft(size,a);
468                 slideup(size,a);
469                 slideright(size,a);
470                 slideunder(size,a); break;
471             case 8 : slideright(size,a);                    //switch case random slide
472                 slideleft(size,a);
473                 slideup(size,a);
474                 slideright(size,a);
475                 slideunder(size,a); break;
476             case 9 : slideup(size,a);                       //switch case random slide
477                 slideleft(size,a);
```

```
                slideleft(size,a);
                slideup(size,a);
                slideright(size,a);
                slideunder(size,a); break;
            }                                           //why many case? because it will not match between two table
        }
    } //rd_puzzle
    void rd_check(int size,int a[][size],int b[][size])
    {
        int i,j;
            for(i=0;i<size;i++){
                for(j=0;j<size;j++){
                    if (a[i][j] == b[i][j]){            //condition (if) for when  two table is match it will random
                        rd_puzzle(size,a);              //calling function random
                    }
                }
            }
    }
    char getch(){                                       //this function  for Linux
        //#include <unistd.h>   //_getch
        //#include <termios.h>  //_getch
        char buf=0;
        struct termios old={0};
        fflush(stdout);
        if(tcgetattr(0, &old)<0)
            perror("tcsetattr()");
        old.c_lflag&=~ICANON;
        old.c_lflag&=~ECHO;
        old.c_cc[VMIN]=1;
        old.c_cc[VTIME]=0;
        if(tcsetattr(0, TCSANOW, &old)<0)
            perror("tcsetattr ICANON");
        if(read(0,&buf,1)<0)
            perror("read()");
        old.c_lflag|=ICANON;
        old.c_lflag|=ECHO;
        if(tcsetattr(0, TCSADRAIN, &old)<0)
            perror ("tcsetattr ~ICANON");
        printf("%c\n",buf);
        return buf;
    } //getchlinux
```

```
    } //getchlinux
    void clear_data()
    {
        FILE *in;                                       //declare for open file
        int i,check=0;                                  //declare "check" for check amount of login admin
        char ch,ans[3] = "11",use[10]="11",pass[10]="11";
        while(strcmp(ans,"exit")!=0 && strcmp(ans,"edit")!=0 ){ //condition (while) for check "edit" and "exit"
            printf("Do you want to exit or edit? (exit/edit): ");
            scanf("%s",ans);
        }
        if(strcmp(ans,"edit") == 0){                    //condition (if) for check "edit"
                strcmp(ans,"11");
                while(strcmp(ans,"yes")!=0 && strcmp(ans,"no")!=0 ){
                printf("Do you want to clear data? (yes/no): ");
                scanf("%s",ans);
                }
            if(strcmp(ans,"yes") == 0){                 //condition (if) for check "ues"
                while(check < 3){
                    while(strcmp(use,"admin")!=0){
                        check++;                        //check +1
                        printf("\t\tUsername: ");
                        scanf("%s",use);
                        if (check > 3) break;           //if check > 3 --> break
                    }
                    if (check > 3) {
                        break;
                    }else check = 0;
                    while(strcmp(pass,"manzaza")!=0){
                            check++;                    //check +1
                            printf("\t\tPassword: ");
                            scanf("%s",pass);
                            if (check > 3) break;       //if check > 3 --> break
                    }
                    if (strcmp(use,"admin") == 0 && strcmp(pass,"manzaza") == 0)
                        check = 0;  break;
                }
                if(check < 3){
                    in = fopen("Score_data.txt","w");
                    printf("Clear data completely...\n");
                    printf("\n\nPress X to menu: ");
                    while(ch != 'x' && ch != 'X'){
```

```
556                 printf("\n\nPress X to menu: ");
557                 while(ch != 'x' && ch != 'X'){
558                     ch = getch();
559                 }
560             }else {                                         //else check > 3 show try again later
561                 printf("Please try again later...");
562                 printf("\n\nPress X to menu: ");
563                 while(ch != 'x' && ch != 'X'){
564                     ch = getch();
565                 }
566             }
567         }else printf("Data not clear...\n");
568     }
569 } //clear_data
570 void puzzle()
571 {
572     FILE *in;                                               //declare for open file
573     int result;
574     char text;                                              //declare for keep charector
575     in = fopen("puzzle.txt","r");
576     result = fscanf(in,"%c",&text);
577     while (result != EOF){
578         printf("%c",text);
579         result = fscanf(in,"%c",&text);
580     }
581     printf("\n\n");
582     fclose(in);                                             //close file
583 } //pluzzle
584 void mickey()
585 {
586     FILE *in;                                               //declare for open file
587     int result;
588     char text;                                              //declare for keep charector
589     in = fopen("mickey.txt","r");
590     result = fscanf(in,"%c",&text);
591     while (result != EOF){
592         printf("%c",text);
593         result = fscanf(in,"%c",&text);
594     }
595     fclose(in);                                             //close file
596 } //mickey
```

ส่วนของไฟล์ Puzzle.c

```
1  /*Program name : Puzzle.c
2  Student name : Namchok Singhachai
3  Student No. 60160169 Section 02
4  The projects of Software Engineering #7 BUU60*/
5  #include"puzzle.h"
6  int main(int argc, char *argv[])
7  {
8      char choice;                                         //declare variable
9      do{                                                  //condition (do_while) for menu
10         system("clear");                                        //clear screen
11         puzzle();                                        //calling function
12         main_menu();
13         time_t rawtime;                                  //search form Internet
14         time (&rawtime);
15         printf("\t\t       %s\n",ctime(&rawtime));
16         choice = getch();                                //function getch(); //choose choice
17         switch(choice){                                  //condition (switch case) for choose choice
18             case '1': system("clear"); play(); break;            //calling function
19             case '2': system("clear"); how2play(); break;
20             case '3': system("clear"); high_score(); break;
21             case '4': system("clear"); credit(); break;
22             default : printf("Please again...\n");       //another case
23         }
24     }while(choice != '5' );
25     system("clear");
26     mickey();                                            //calling function mickey
27     return 0;
28 } //main
```