ใบงานที่ 4

เรื่อง Program INSERT/DELETE
Function of Circular Queue

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นายธีรเดช ประเสริฐวงศ์พนา
65543206016-9

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี
หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา
ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

* ให้ระบุคำสั่ง/คำชี้แจง/คำอธิบาย ตามที่กำหนดให้

1. สร้างโปรแกรม INSERT/DELETE function of Circular Queue ตามตัวอย่างในเอกสารประกอบการสอน 3
2. แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
3. แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
4. สรุปผลการทดลอง

ลำดับขั้นการทดลอง

* ให้ระบุขั้นตอนการทดลอง ผลลัพธ์ที่ได้ โดยใช้รูปภาพประกอบ และอธิบายอย่างละเอียด

แสดงโค้ดโปรแกรมเป็นส่วน ๆ พร้อมอธิบาย

ส่วนที่ 1

```c
#include <stdio.h> //use printf()
#include <conio.h> //use getch()
#define N 5 //Set Max Queue
int Q[N]; //Prepare Queue 0..N-1
int x, Qnumber = 0, F = 0,R = 0; //Declare x and initial Qnumber/Front/Rear
char status = 'N'; //Initial Status = NORMAL
char ch; //KBD Read variable
```

ส่วนที่ 1 เป็นที่ประการเรียกใช้ library และประการตัวแปร ในโค้ดดังนี้

library 2 ตัวคือ <stdio.h> และ <conio.h>

ตัวแปร constants 1 ตัว คือ N ค่า 5 เพื่อบอกค่าสูงสุดของคิว

ตัวแปร int 4 ตัว คือ Q[N] เป็นอาเรย์เอาไว้เก็บข้อมูลตามตำแหน่ง, x เก็บค่าค่าตัวเลข,

Qnumber เอาไว้เก็บหมายเลขคิวที่รันสะสม(ไม่ใช่ current), F คือ front pointer, R คือ rear pointer

ตัวแปร char 2 ตัวคือ status เพื่อบอกสถานะ, ch เก็บและอ่านค่าจากคีย์บอร์ด

ส่วนที่ 2 insertCQ

```
void insertCQ(int y) //INSERT Function
{
if((R==F-1) || (R==N-1 && F==1) ) //Check Queue FULL?
{
printf("!!!OVER FLOW!!!...\n");
status='O'; //set status = OVER FLOW
}
else
{
if(R==N-1) // Loop back of R to 1 if it maximum
{
R=1;
}
else
{
R++; // increase R if Normal
if(F==0) //if F is zero set it to 1
F=1;
}
Qnumber++; //Increase queue number
printf("Youe are queue number : %d\n", Qnumber);
//Display queue number
Q[R]=y; //Put data into Queue
status = 'N'; //Set status to "NORMAL"
}
}
```

ส่วนที่ 2 ฟังก์ชั่น void insertCQ(int y) มีหน้าที่นำข้อมูลเข้าสู่คิว โดยใน parameter มี int y ที่รับข้อมูลมาจาก main()

-ในกรอบสีแดงจะมี if ((R == F - 1) || (R == N - 1 && F == 1)) คือ ถ้า R ชี้ชิดติดหลัง F หรือ R ชี้ตำแหน่งสุดท้ายและ F ชี้ตำแหน่งที่ 1 เมื่อตรงตามเงื่อนไข จะทำการแสดงข้อความคำว่า “ !!!OVER FLOW!!!... ” และ set ให้ตัวแปร status = “O”

-ในกรอบสีส้มจะมี if (R == N - 1) คือ ถ้า R ชี้ตำแหน่งสุดท้ายให้ R กลับไปชี้ตำแหน่งที่ 1

-ในกรอบสีเขียว R++ คือชี้ไปด้าน 1 ตำแหน่ง, if(F == 0) {F=1} คือ ถ้า F อยู่ตำแหน่งที่ 0 ให้F มาชี้ตำแหน่งที่ 1

Qnumber++ คือเพิ่มหมายคิวไปทีละ1

printf("Your are queue number : %d\n", Qnumber) คือแสดงหมายเลขคิว

Q[R] = y คือ นำข้อมูลตัวเลขที่อยู่ในตัวแปร y เก็บไว้ในคิวตามตำแหน่ง R

status = ‘N’ คือ set ให้ status เท่ากับ N

ส่วนที่ 3 deleteCQ

```
int deleteCQ() //DELETE Function
{
 int y;
 if (F == 0)
 {
 printf("\n!!!UNDER FLOW!!!...\n");
 status = 'U'; //set STATUS = "UNDER FLOW"
 }
 else
 {
 y=Q[F]; //Get data from Queue
 if (F==R) //Set both to 0 if F and R are same value
 {
 F=0; R=0;
 }
 else
 {
 if(F==N-1) //Set F to 1 if F is maximum, otherwise increase F
 F=1;
else
 F++;
 }
 status = 'N'; //Set status to "NORMAL"
 return(y); //Return data
 }
```

ส่วนที่ 3 ฟังก์ชั่น int deleteCQ() เป็นการลบคิว โดยการลบคิวจะอ้างอิงตำแหน่งจาก F

Int y คือ ประกาศตัวแปลใหม่ชื่อ y เอาไว้เก็บค่า

-**ในกรอบสีแดง**จะมีif (F == 0) คือ ถ้า F ชี้ตำแหน่งที่ 0 ให้ แสดงข้อความคำว่า !!!UNDER FLOW!!!... และ set status = ‘U’

-**ในกรอบสีเหลือง** y = Q[F] คือ ให้ y รับข้อมูลจาก Q[F], if(F == R) คือ ถ้า F ชี้ตำแหน่งเดียวกันกับ R ให้ F และ R เท่ากับ 0

-**ในกรอบเขียว** if (F == N - 1) คือ ถ้า F ชี้ตำแหน่งสุดท้าย ให้F กลับไปชี้ตำแหน่งที่ 1 แต่ถ้าไม่ใช่ให้ F ชี้ไปข้างหน้า 1 ตำแหน่ง

status = ‘N’ คือ set ให้ status เท่ากับ N

return (y) คือ รีเทิร์นตัวแปร y ออกไป

ส่วนที่ 4 DataInQueue

```
int DataInQueue() // Calculate Data waiting in queue
{
 int y=0;
 if (F!= 0 && R!=0) //if not equal then can calculate
 {
 if (F<=R)
 y=R-F+1; //Normal F and R
 else
 y=(N-1)-F+1+R; //incase loop of R
 }
 return(y);
}
```

ส่วนที่ 4 ฟังก์ชั่น int DataInQueue() เป็นคำนวณข้อมูลทั้งหมด ว่ามีกี่จำนวนข้อมูลที่ยังค้างอยู่ในคิว

Int y = 0 คือ ประกาศตัวแปลใหม่ชื่อ y มีค่าเริ่มต้นเท่า 0 เอาไว้เก็บค่า

if (F != 0 && R != 0) คือ ถ้า F และ R ไม่ได้ชี้ตำแหน่งที่ 0 ให้ทำตามเงื่อนไขตามกรอบสีแดง

-กรอบสีแดงจะมี if (F != 0 && R != 0) คือ ถ้า F มากกว่าหรือเท่า R ให้ y = R - F + 1 แต่ถ้าไม่ใช่ให้ y = (N - 1) - F + 1 + R

return (y) คือ รีเทิร์นตัวแปร y ออกไป

ส่วนที่ 5 ShowAllQueue()

```
void ShowAllQueue() //Display Function
{
 int i; //Counter variable
 printf("N : %d\n",N-1);
 printf("Status = %c \n",status); //Display STATUS
 printf("Data waiting in queue = %d\n",DataInQueue());
//Display Data waitting in queue
 printf(" F = %d / R = %d\n",F,R); //Display F R
 for ( i = 1; i < N; i++ )
 {
 printf("%d:%d / ",i, Q[i]); //Display all of data in QUEUE
 }
 printf("\n-----------------------------------------------------------\n");
}
```

ส่วนที่ 5 ฟังก์ชั่น void ShowAllQueue() เป็นการนำข้อมูลที่อยู่ในคิวทั้งหมดมาแสดง

ประกาศตัวแปรใหม่ int i เพื่อใช้ในการ loop

printf(" N : %d\n ", N - 1); คือ แสดงจำนวนคิวที่สูงสุงที่คิวเก็บได้

printf("Status : %c\n ", status); คือ แสดงสถานะ

printf("Data waiting in queue = %d\n", DataInQueue()); คือ แสดงจำนวนข้อมูลที่ยังค้างอยู่ในคิว

-กรอบสีแดงคือ for loop เพื่อนำค่าที่อยู่ในคิวหรือตัวแปร Q[ ] มาแสดงทั้งหมดโดยอ้างอิงตำแหน่งจากตัวแปร I ตามเงื่อนไขใน for

ส่วนที่ 6 main()

```
int main()
{
 printf("CICULAR QUEUE PROGRAM...\n");
 printf("======================\n");
 ch=' ';
 while (ch != 'E')
 {
 printf("\n[1=INSERT : 2=DELETE E:Exit] : "); //Show MENU
 ch = getch(); //Wait and read KBD with out ENTER Press
 switch(ch) //Check ch
 {
 case '1' : printf("\nInsert Number : ");
 scanf("%d", &x); //Read data from KBD
 insertCQ(x); //Call INSERTNQ Function
 ShowAllQueue(); //Display all data in Queue
 break;
 case '2' : x=deleteCQ(); //Delete data
 printf("\nData from Queue = %d\n",x); //Display it
 ShowAllQueue(); //Display all data in Queue
 break;
 } //End SWITCH CASE
 } //End WHILE Loop
 printf("\n"); //line feed
 return(0);
}//End MAIN Fn.
```

ส่วนที่ 6 ฟังก์ชั่น int main() เป็นฟังก์ชั่นหลักหรือฟังก์ชั่นที่โปรแกรมจะเริ่มทำงาน

ch = ‘ ‘ คือ ให้ch เท่ากับค่าว่าง

โดย while (ch != ‘E’) เพื่อเช็ค ch ต้องมีค่าไม่เท่ากับ E ถ้าเข้าเงื่อนไขก็เข้า loop ทำงานต่อไปนี้

ch = getch() คือ getch() รับค่าจากคีย์บอร์ดแล้วเก็บไว้ในตัวแปร ch และ auto enter

switch (ch) คือ ถ้า ch เป็นอะไรก็จะเข้าตาม case นั้นๆ

-กรอบสีแดง ถ้า ch = 1 จะเข้า case 1

scanf("%d", &x) คือรับค่าจากคีย์บอร์ดเป็นตัวเลขแล้วตัวไว้ในตัวแปร x

insertCQ(x) คือ เรียกใช้ฟังก์ชั่น insertCQ(x) ใน parameter แนบ x เข้าไปด้วย

ShowAllQueue () คือ เรียกใช้ฟังก์ชั่น ShowAllQueue () เพื่อแสดงค่าที่อยู่ใน Q[ ] ทั้งหมด

-กรอบสีเหลือง ถ้า ch = 2 จะเข้า case 2

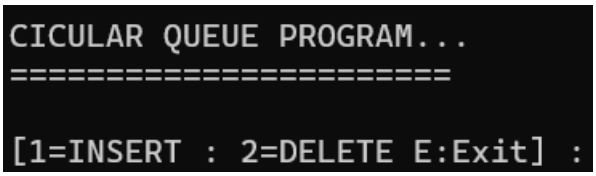
x = deleteCQ() คือ เมื่อฟังก์ชั่น deleteCQ() ทำงานเสร็จให้เก็บค่าไว้ใน x

printf("\nData from Queue = %d\n", x) คือ ปริ้นค่าตัวแปร x ออกมาแสดง

ShowAllQueue () คือ เรียกใช้ฟังก์ชั่น ShowAllQueue () เพื่อแสดงค่าที่อยู่ใน Q[ ] ทั้งหมด

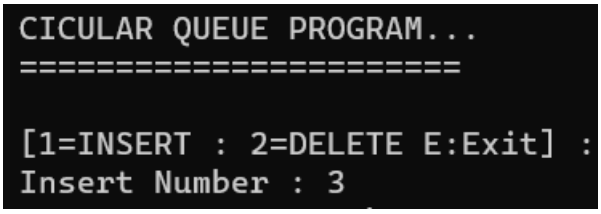
แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน

1. เมื่อผู้ใช้งานรันโปรแกรมจะขึ้นดังภาพที่ 1 ให้ผู้ใช้เลือกพิมพ์หมายเลขตามเมนูถ้าเลือกเลข 1 จะเป็นการใส่ข้อมูลเพิ่มเข้าไปแต่ถ้าเลือกเลข 2 จะเป็นที่นำข้อมูลล่าสุดออก

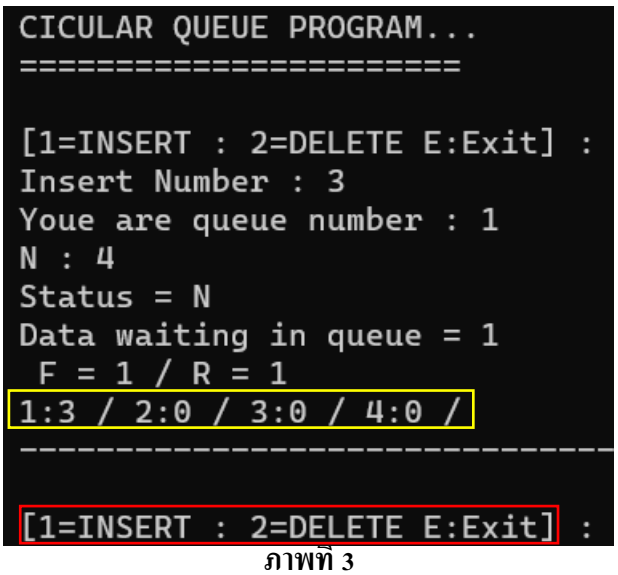


ภาพที่ 1

2. ถ้าผู้ใช้เลือกหมายเลข 1 จะขึ้นให้ใส่ข้อมูลดังภาพที่ 2 เมื่อใส่ข้อมูลเสร็จแล้ว Enter จะแสดง ดังภาพที่ 3 (Your are queue number : 1 คือจำนวนคิวที่รันสะสม, N : 4 คือจำนวนคิวของข้อมูลที่รองรับสูงสุด, Status : N คือสถานะปกติ, F = 1 คือตำแหน่งของ front pointer ปัจจุบัน, R = 1 คือตำแหน่งของ rear pointer ปัจจุบัน, Data waiting in queue คือ จำข้อมูลที่รออยู่ในคิว) จะสังเกตว่า โปรแกรมจะแสดงคิวที่เก็บข้อมูลตั้งแต่ 1 – 4 ดังกรอบสีเหลือง ดังนั้นคิวที่ 1 เป็นจะ 1:5 เพราะเราใส่ข้อมูลครั้งแรกคือ 3 (ตำแหน่งคิว : ข้อมูล) จากนั้นโปรแกรมจะขึ้นให้เลือกคำสั่งอีกครั้ง ดังกรอบสีแดง

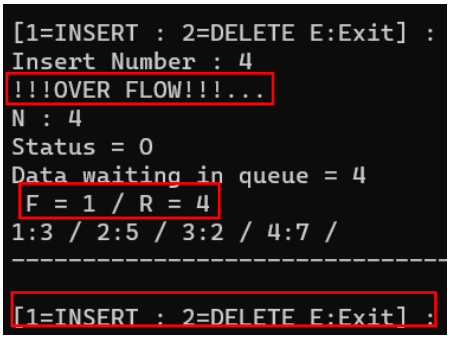


ภาพที่ 2



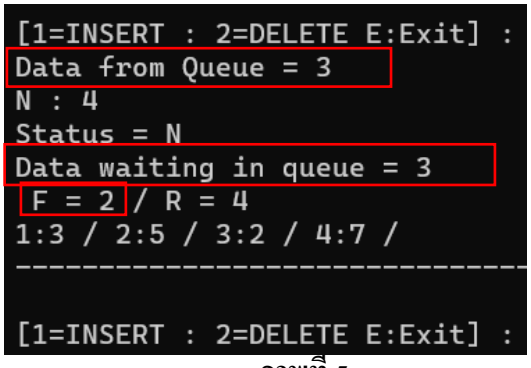
ภาพที่ 3

3. เมื่อเราทำการ INSERT จน R ถึงตำแหน่งคิวเดียวกับ F แล้วยังทำการ INSERT อีก โปรแกรมจะขึ้นข้อความคำว่า “ !!!OVER FLOW!!!... ” และ Status เปลี่ยนเป็น O ดังภาพที่ 4 จากนั้นโปรแกรมจะขึ้นให้เลือกคำสั่งอีกครั้ง



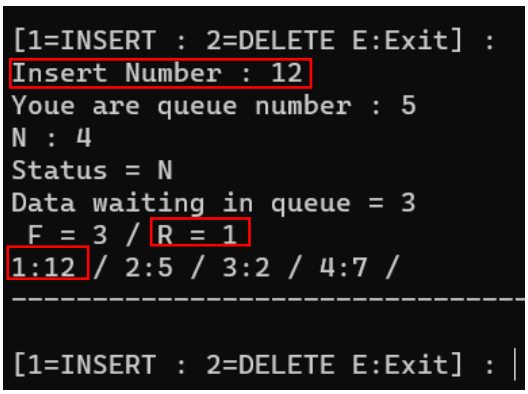
ภาพที่ 4

4. เมื่อเราเลือกหมายเลข 2 จะเป็นคำสั่ง DELETE จะเป็นการลบข้อมูลหรือคิวที่ INSERT ไปแรกสุดออกมาและ F จะเลื่อนตำแหน่งเพิ่ม 1 ดังภาพที่ 5 เมื่อ F = N แล้วยัง DELETE ต่อ F จะกลับมาชี้คิวที่ 1



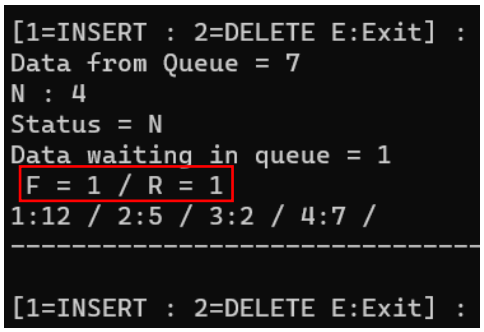
ภาพที่ 5

5. ต่อจากข้อก่อนหน้านี้ ดังนั้นเมื่อกับทำการ INSERT ต่อตัวโปรแกรมจะไม่ OVER FLOW แล้วแต่จะ เป็นการวนคิวแทนโดยที่ R จากตำแหน่งคิวสุดท้ายจะกลับไปตำแหน่งคิวที่ 1 ดังภาพที่ 6 ดังนั้น ถ้า เราไม่ทำให้ F < 0 หรือ F กับ R ชี้ตำแหน่งคิวเดียวกันก็จำสามารถวนเพิ่มข้อมูลได้ตลอด

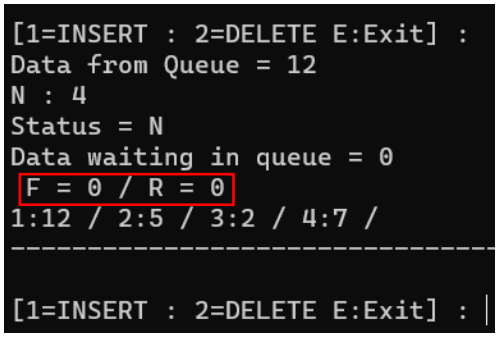


ภาพที่ 6

6. เมื่อเรา DELETE จน F ชี้ตำแหน่งคิวเดียวกันกับ Rดังภาพที่7แล้วยัง DELETEต่อจะทำให้ F และ R ถูก reset ค่าเป็น 0 รวมถึงข้อมูลที่อยู่ในคิวด้วย ดังภาพที่ 8

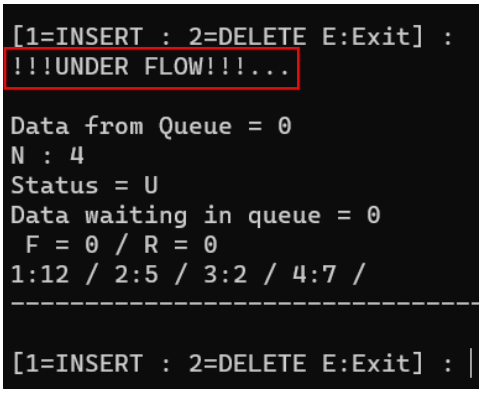


ภาพที่ 7



ภาพที่ 8

7. ต่อจากข้อ 6 ภาพที่ 8 ถ้ายังทำการ DELETE อีก โปรแกรมจะขึ้นข้อความคำว่า “ !!!UNDER FLOW!!!... ” และ Status เปลี่ยนเป็น U ดังภาพที่ 9



ภาพที่ 9

สรุปผลการทดลอง

* ให้ระบุข้อมูลสรุปผลการดำเนินงาน สิ่งที่ได้เรียนรู้ ปัญหา และวิธีการแก้ไข

สรุปผล : โปรแกรมนี้เป็นการ เพิ่ม/ลบ คิว และสามารถวนคิวได้หรือก็คือ Circular Queue โดยจะมี F (front pointer) และ R (rear pointer) ในชี้ตำแหน่งและจัดการคิว ทำให้คิวไม่เกิดช่องว่างที่ศูนย์เปล่าในกรณีที่ DELETE ไปแล้วทำให้F เลื่อนตำแหน่ง R ก็จะสามรถวนมาต่อหลัง F ได้นั่นเอง

ปัญหา : จากโปรแกรมจะสังเกตว่าตอนที่เรา DELETE ข้อมูลที่อยู่ใน Q[ ] ไม่ได้ถูกเอาออกจริงหรือไม่ได้ set 0 ดังภาพที่ 6 กับ 7 และในกรณี F = R ไม่ reset Q[ ] ดังภาพที่ 11

วิธีการแก้ไข : เราจะเพิ่มโค้ดที่เข้าไปใน function deleteCQ() เพื่อให้ข้อมูลถูกนำออกจริงโดยการ set 0 ที่ตำแหน่งของ F ของตัวแปร Q[ ] ดังภาพที่ 10 จากนั้นเราจะทำการทดสอบ ผลลัพธ์จากการแก้ไขจะแสดงดังภาพที่ 11 และ 1

```
int deleteCQ() //DELETE Function
{
 int y;
 if (F == 0)
 {
 printf("\n!!!UNDER FLOW!!!...\n");
 status = 'U'; //set STATUS = "UNDER FLOW"
 }
 else
 {
 y=Q[F]; //Get data from Queue
 Q[F] = 0; //Set data 0
 if (F==R) //Set both to 0 if F and R are same value
 {
 F=0; R=0;
 }
 else
 {
 if(F==N-1) //Set F to 1 if F is maximum, otherwise increase F
 F=1;
else
 F++;
 }
 status = 'N'; //Set status to "NORMAL"
 return(y); //Return data
 }
}
```

ภาพที่ 10

```
[1=INSERT : 2=DELETE E:Exit] :
Data from Queue = 3
N : 4
Status = N
Data waiting in queue = 3
 F = 2 / R = 4
1:0 / 2:5 / 3:2 / 4:7 /
------------------------------

[1=INSERT : 2=DELETE E:Exit] :
Data from Queue = 5
N : 4
Status = N
Data waiting in queue = 2
 F = 3 / R = 4
1:0 / 2:0 / 3:2 / 4:7 /
------------------------------

[1=INSERT : 2=DELETE E:Exit] :
Data from Queue = 2
N : 4
Status = N
Data waiting in queue = 1
 F = 4 / R = 4
1:0 / 2:0 / 3:0 / 4:7 /
------------------------------

[1=INSERT : 2=DELETE E:Exit] :
```

ภาพที่ 11

```
[1=INSERT : 2=DELETE E:Exit] :
Data from Queue = 7
N : 4
Status = N
Data waiting in queue = 0
 F = 0 / R = 0
1:0 / 2:0 / 3:0 / 4:0 /
------------------------------

[1=INSERT : 2=DELETE E:Exit] :
```

ภาพที่ 12

ตอบคำถามท้ายการทดลอง

* ตอบคำถามท้ายการทดลอง (ถ้ามี)

.....

.....

.....

.....

สื่อ / เอกสารอ้างอิง

* ให้ระบุที่มาของข้อมูลที่ใช้ประกอบการทดลอง เช่น หนังสือ เว็บไซต์ youtube เป็นต้น

-ในเอกสารประกอบการสอนสัปดาห์ที่ 3 ของอาจารย์ปิยพล ยืนยงสถาวร