

เรื่อง การคำนวณ และคำสั่งภาษาเบื้องต้น

รายวิชาการเขียนโปรแกรมภาษาซี 1
กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี
ชั้นมัธยมศึกษาปีที่ 5

```
int i = 0;  
loop: // label for goto statement  
do {  
    if (i == 3) {
```

หน่วยการเรียนรู้ที่ 8

เรื่อง คำสั่งวนรอบการทำงาน

```
    ++i;  
    goto loop;
```

```
std::cout << i <<  
    ++i;
```

กัณษดา พานิชเจริญ

ตำแหน่ง ครู วิทยฐานะครูชำนาญการพิเศษ

โรงเรียนวชิรปราการวิทยาคม

สำนักงานเขตพื้นที่การศึกษามัธยมศึกษากำแพงเพชร

สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน กระทรวงศึกษาธิการ

คู่มือสำหรับนักเรียน

มีคำแนะนำสำหรับนักเรียนผู้ใช้เอกสารประกอบการเรียน เรื่องการคำนวณและคำสั่งภาษาเบื้องต้น รายวิชาการเขียนโปรแกรมภาษาซี 1 กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5 หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน ใช้เวลา 4 ชั่วโมง ให้นักเรียน ปฏิบัติกิจกรรมตามขั้นตอนดังต่อไปนี้

1. การเตรียมตัวของนักเรียน

- 1.1 ให้นักเรียนอ่านทำความเข้าใจคู่มือสำหรับนักเรียน
- 1.2 ศึกษาเอกสารประกอบการเรียนตามลำดับของเนื้อหา

2. บทบาทของนักเรียน

- 2.1 ศึกษาเอกสารประกอบการเรียนด้วยความตั้งใจ และฝึกปฏิบัติ อย่างเต็มความสามารถของตนเอง
- 2.2 ควรร่วมแสดงความคิดเห็น อธิบาย อภิปราย ซักถามกันภายในกลุ่ม รวมถึง มีการวางแผนการทำงาน เพื่อให้งานเสร็จภายในเวลาที่กำหนด
- 2.3 ในขณะที่ทำแบบทดสอบหรือแบบฝึกหัด ต้องมีความซื่อสัตย์ ไม่ควรดูคำตอบเพื่อน หรือดูเฉลยคำตอบก่อน
- 2.4 เมื่อมีข้อสงสัยนักเรียนสามารถขอคำปรึกษาหรือคำแนะนำจากครูผู้สอนทันที

3. ส่วนประกอบของบทเรียนเอกสารประกอบการเรียน หน่วยการเรียนรู้ที่ 8

เรื่อง คำสั่งวนรอบการทำงาน

3.1 ส่วนที่ 1 ประกอบด้วย

- 3.1.1 ปก
- 3.1.2 คำนำ
- 3.1.3 สารบัญ
- 3.1.4 คู่มือสำหรับนักเรียน
- 3.1.5 ลำดับขั้นตอนการใช้เอกสารประกอบการเรียน หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน
- 3.1.6 มาตรฐานการเรียนรู้/ผลการเรียนรู้



- 3.1.7 ฟังมโนทัศน์สาระการเรียนรู้ หน่วยการเรียนรู้ที่ 8
เรื่อง คำสั่งวนรอบการทำงาน
- 3.1.8 แบบทดสอบก่อนเรียน
- 3.1.9 กระจายคำตอบแบบทดสอบก่อนเรียน
- 3.1.10 ใบความรู้ที่ 1 เรื่อง คำสั่ง While loop
- 3.1.11 ใบกิจกรรมที่ 1 เรื่อง คำสั่ง While loop
- 3.1.12 ใบความรู้ที่ 2 เรื่อง คำสั่ง Do-while loop
- 3.1.13 ใบกิจกรรมที่ 2 คำสั่ง Do-while loop
- 3.1.14 ใบความรู้ที่ 3 เรื่อง คำสั่ง For loop
- 3.1.15 ใบกิจกรรมที่ 3 เรื่อง คำสั่ง For loop
- 3.1.16 ใบความรู้ที่ 4 เรื่อง คำสั่งวนรอบซ้อนกัน
- 3.1.17 ใบกิจกรรมที่ 4 เรื่อง คำสั่งวนรอบซ้อนกัน
- 3.1.18 แบบทดสอบหลังเรียน
- 3.1.19 กระจายคำตอบแบบทดสอบหลังเรียน
- 3.2 ส่วนที่ 2 ประกอบด้วย
 - 3.2.1 แบบบันทึกการประเมินผล หน่วยการเรียนรู้ที่ 8
เรื่อง คำสั่งวนรอบการทำงาน
 - 3.2.2 เฉลยแบบทดสอบก่อนเรียน
 - 3.2.3 เฉลยใบกิจกรรมที่ 1 – 4
 - 3.2.4 เฉลยแบบทดสอบหลังเรียน
- 4. ขั้นตอนการเรียนรู้ด้วยเอกสารประกอบการเรียน หน่วยการเรียนรู้ที่ 8
เรื่อง คำสั่งวนรอบการทำงาน
 - 4.1 ศึกษาคู่มือสำหรับนักเรียน
 - 4.2 ฟังคำแนะนำของครูในการปฏิบัติกิจกรรมการเรียนรู้จากเอกสารประกอบการเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน
 - 4.3 ศึกษาหัวข้อเนื้อหาจากฟังมโนทัศน์สาระการเรียนรู้ หน่วยการเรียนรู้ที่ 8
เรื่อง คำสั่งวนรอบการทำงาน

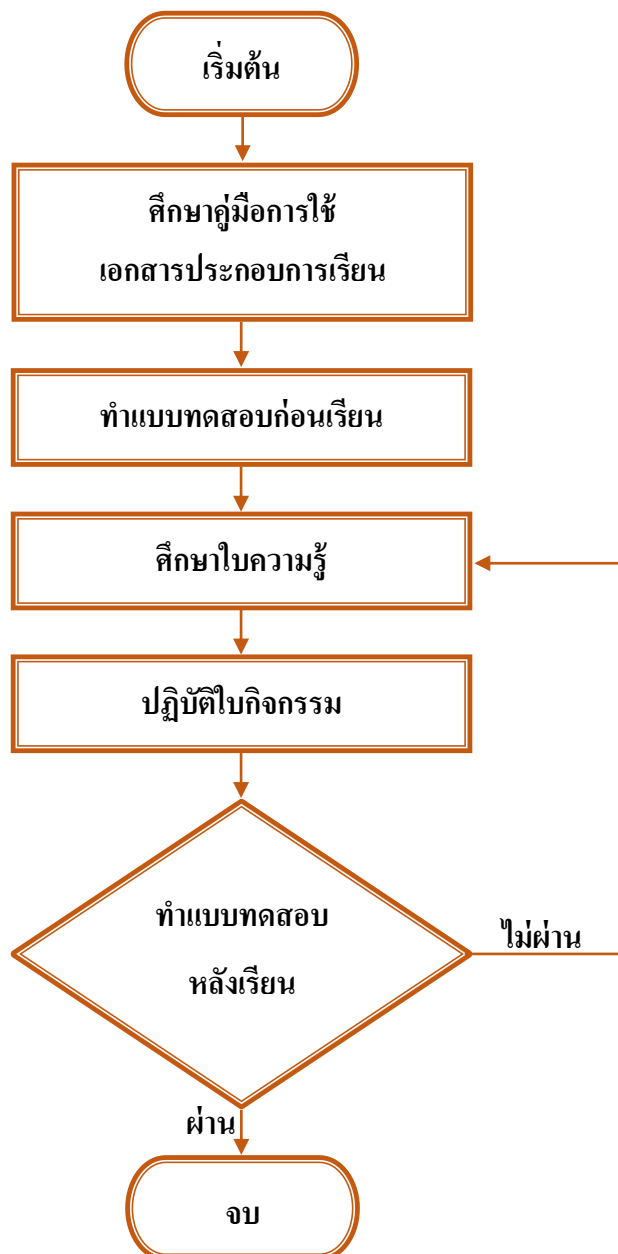


- 4.4 ทำแบบทดสอบก่อนเรียน แบบเลือกตอบ หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน แบบเลือกตอบ จำนวน 10 ข้อ และอัตนัย จำนวน 1 ข้อ เมื่อทำเสร็จให้นักเรียนเปลี่ยนกันตรวจ โดยดูเฉลยจากภาคผนวกด้วยความซื่อสัตย์สุจริต บันทึกผลและส่งให้ครูได้ตรวจสอบความถูกต้องอีกครั้ง
- 4.5 ศึกษาใบความรู้ด้วยความตั้งใจ ทำกิจกรรมกลุ่ม กลุ่มละ 4 – 5 คน ร่วมอภิปรายในหัวข้อการเรียนรู้ที่ได้มอบหมาย สรุปความรู้ร่วมกันและฝึกปฏิบัติใบกิจกรรม
- 4.6 ทำแบบทดสอบหลังเรียนแบบเลือกตอบ จำนวน 10 ข้อ และอัตนัย จำนวน 1 ข้อ เมื่อทำเสร็จให้นักเรียนเปลี่ยนกันตรวจ โดยดูเฉลยจากภาคผนวกด้วยความซื่อสัตย์สุจริต บันทึกผลและส่งให้ครูได้ตรวจสอบความถูกต้องอีกครั้ง
 - 4.6.1 หากได้คะแนนตั้งแต่ 14 คะแนนขึ้นไปจึงจะผ่านเกณฑ์
 - 4.6.2 หากหากคะแนนต่ำกว่า 14 คะแนน นักเรียนต้องศึกษาใบความรู้และทำแบบทดสอบหลังเรียนจนกว่าจะผ่านเกณฑ์
- 4.7 นักเรียนสามารถทบทวนหรือศึกษาจากเอกสารประกอบการเรียนหรือ E Book
5. การใช้เอกสารประกอบการเรียน หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน
 - 5.5 ไม่ขีดเขียนข้อความใด ๆ ลงในเอกสารประกอบการเรียน
 - 5.6 ไม่ฉีก หรือตัดหน้าเอกสารประกอบการเรียน จะทำให้ข้อความบางตอนไม่สมบูรณ์ ผู้ที่มาอ่านภายหลังไม่ได้ข้อมูลในส่วนที่ขาดหายไป
 - 5.7 เมื่ออ่านยังไม่จบเล่ม ควรค้นหาคำด้วยวัสดุที่มีความบาง เช่น กระดาษหรือที่คั่นไม่ควรพับมุมหรือวางเอกสารประกอบการเรียนคว่ำหน้าลง ซึ่งจะทำให้เอกสารประกอบการเรียนขาดหรือหลุดง่าย
6. การส่งงาน
ให้นักเรียนส่งงานที่โต๊ะครูท้ายชั่วโมง

เด็ก ๆ อ่านคำชี้แจงและลำดับขั้นตอน
การเรียนรู้ให้เข้าใจก่อนลงมือปฏิบัตินะคะ



ลำดับขั้นตอนการเรียนรู้โดยใช้เอกสารประกอบการเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน



มาตรฐานการเรียนรู้/ผลการเรียนรู้

เรื่อง การคำนวณ และคำสั่งภาษาเบื้องต้น

รายวิชาการเขียนโปรแกรมภาษาซี 1 รหัสวิชา ง30249 (รายวิชาเพิ่มเติม)

กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5

หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน เวลา 4 ชั่วโมง

สาระที่ 3 เทคโนโลยีสารสนเทศและการสื่อสาร

สาระสำคัญ

การทำงานแบบวนรอบการทำงาน การทำซ้ำ หรือลูปนั้น จะตรวจสอบเงื่อนไขว่าให้ทำซ้ำขั้นตอนใดบ้าง และ เมื่อใดให้หยุดการทำงานลง ในภาษาซีมีคำสั่งลูปการทำงานอยู่สามประเภท คือ คำสั่ง while loop คำสั่ง do-while loop คำสั่ง for loop และการประยุกต์ใช้คำสั่งวนรอบการทำงานซ้อนกัน เพื่อให้ได้โปรแกรมการทำงานที่เหมาะสม

มาตรฐานการเรียนรู้

มาตรฐาน ง 3.1 เข้าใจ เห็นคุณค่า และใช้กระบวนการเทคโนโลยีสารสนเทศ ในการสืบค้นข้อมูล การเรียนรู้ การสื่อสาร การแก้ปัญหา การทำงาน และอาชีพอย่างมีประสิทธิภาพ ประสิทธิผล และมีคุณธรรม

ตัวชี้วัด ง 3.1 ม.4-6/6 เขียนโปรแกรมภาษา

ผลการเรียนรู้

1. เข้าใจรูปแบบและเลือกใช้ประโยคคำสั่งวนรอบการทำงานได้อย่างถูกต้องเหมาะสม
2. สามารถวิเคราะห์โจทย์ปัญหา ตรวจสอบผลลัพธ์การทำงานและเขียนโปรแกรมวนรอบการทำงานได้



จุดประสงค์การเรียนรู้

1. นักเรียนเข้าใจรูปแบบของประโยคคำสั่งวนรอบการทำงาน
2. นักเรียนสามารถเลือกใช้ประโยคคำสั่งวนรอบการทำงานได้อย่างถูกต้องเหมาะสม
3. นักเรียนสามารถวิเคราะห์โจทย์ปัญหา และตรวจสอบผลลัพธ์ของคำสั่งวนรอบการทำงานได้
4. นักเรียนสามารถเขียนโปรแกรมคำสั่งวนรอบการทำงานที่กำหนดได้





ผังมโนทัศน์สาระการเรียนรู้

เรื่องการคำนวณและคำสั่งภาษาเบื้องต้น

รายวิชาการเขียนโปรแกรมภาษาซี 1 รหัสวิชา ง30249 (รายวิชาเพิ่มเติม)

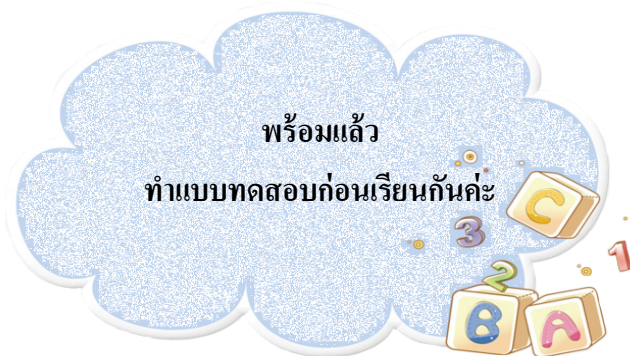
กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5

หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน เวลา 4 ชั่วโมง



พร้อมแล้ว

ทำแบบทดสอบก่อนเรียนกันค่ะ





แบบทดสอบก่อนเรียน หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

แบบทดสอบฉบับนี้มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

ตอนที่ 2 เป็นแบบอัตนัย เต็มคำจำนวน 1 ข้อ 10 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

1. ข้อใด กล่าวถึง รูปแบบของประโยคคำสั่งวนรอบการทำงานในภาษาซีได้ถูกต้อง
 - ก. มีเพียงรูปแบบเดียวเท่านั้น โดยใช้กับการแก้ปัญหาเฉพาะอย่าง
 - ข. มีเพียงรูปแบบเดียวเท่านั้น แต่สามารถประยุกต์ใช้ได้หลากหลาย
 - ค. มีหลายรูปแบบ โดยแต่ละรูปแบบมีการใช้งานที่จำกัด
 - ง. มีหลายรูปแบบ และสามารถประยุกต์ใช้ได้หลากหลาย
2. ข้อใดเป็นรูปแบบของการใช้คำสั่ง while ได้อย่างถูกต้อง
 - ก. มีการตรวจสอบเงื่อนไขก่อนการทำงาน
 - ข. การคำสั่งเพิ่มค่าตัวแปรได้ครั้งละ หนึ่งค่าเท่านั้น
 - ค. คำสั่งจะต้องทำงานในลูปมาแล้วอย่างน้อย 1 รอบเสมอ
 - ง. คำสั่งจะใช้ได้ก็ต่อเมื่อรู้จำนวนรอบที่แน่นอนเท่านั้น
3. รูปแบบการวนรอบการทำงานที่ทำงานตามคำสั่งก่อนแล้วจึงตรวจสอบเงื่อนไขภายหลัง คือข้อใด

ก. for	ข. while
ค. break	ง. do while
4. หากโจทย์มีการกำหนดจำนวนรอบที่แน่นอน ควรใช้คำสั่งการวนรอบในข้อใด

ก. for	ข. while
ค. break	ง. do while





5. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int i = 5;
while (i<=7)
{
    printf("A");
    i++;
}
```

ก. AAAAA

ข. AAAA

ค. AAA

ง. AA

6. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int i=1,j=0;
while(i<=2)
{
    for(j=1; j<=3; j++)
    {
        printf("%d", j);
    }
    i++;
    printf("\n");
}
```

ก. 123

ข. 321

123

321

123

321

ค. 123

ง. 321

123

321



7. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int A, B;  
for(A=3; A>0; A--)  
{  
    for(B=1; B<=A; B++)  
    {  
        printf("%d", A);  
    }  
    printf("\n");  
}
```

ก. 123

12

1

ก. 1

22

333

ข. 1

12

123

ง. 333

22

1

8. ให้นักเรียนเขียนโปรแกรมเพื่อแสดงเลขคี่ ระหว่าง 1 – 100 โดยใช้คำสั่ง while loop

```
int i=1;
while (①.....)
{
    print("%d ",i);
    (②.....);
}
```

- ก. ① $i \leq 100$ และ ② $i++$
ข. ① $i \geq 100$ และ ② $i++$
ค. ① $i \leq 100$ และ ② $i+=2$
ง. ① $i \geq 100$ และ ② $i+=2$

9. จากโปรแกรมควรเติมการตรวจสอบเงื่อนไขในข้อใด เพื่อหาผลรวมของเลขคี่ ระหว่าง 1 – 20 โดยใช้คำสั่ง do while loop

```
int num = 1, sum = 0;
do
{
    sum += num;
    (①.....);
}while (②.....);
print("sum of odd num: %d", sum);
```

- ก. ① $num++$ และ ② $num \leq 20$
ข. ① $num++$ และ ② $num \geq 20$
ค. ① $num+=2$ และ ② $num \leq 20$
ง. ① $num+=2$ และ ② $num \geq 20$

10. จากโปรแกรมควรเติมการตรวจสอบเงื่อนไขในข้อใด เพื่อให้ได้ “Good”
แสดงทางหน้าจอจำนวน 3 ครั้ง

for (.....)

print("Good");

ก. $i = 1; i \leq 4; i++$

ข. $i = 1; i \leq 3; i++$

ค. $i = 1; i \leq 4;$

ง. $i = 1; i \leq 3;$

ตอนที่ 2 ข้อสอบแบบอัตนัย (10 คะแนน)

คำสั่ง ให้นักเรียนเขียนโปรแกรมหาผลบวกของเลขจำนวนเต็มที่มีค่าอยู่ในช่วงตั้งแต่ x ถึง y โดยที่ค่า x และ y ถูกผู้ใช้กำหนดขึ้น โดยให้ค่า $y > x$ แล้วพิมพ์ผลลัพธ์ทางจอภาพ

ตอบ.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



กระดาษคำตอบแบบทดสอบก่อนเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

เอกสารประกอบการเรียน เรื่องการคำนวณและคำสั่งภาษาเบื้องต้น
รายวิชาการเขียนโปรแกรมภาษาซี 1 รหัสวิชา ง30249 (รายวิชาเพิ่มเติม)
กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5

ชื่อ - สกุล.....ชั้น.....เลขที่.....

ข้อ	ก	ข	ค	ง
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

คะแนนแบบทดสอบก่อนเรียน

คะแนนตอนที่ 1

คะแนนตอนที่ 2

คะแนนรวมที่ได้

ผลการประเมิน

☐ ดีมาก ★★★★★

☐ ดี ★★★

☐ พอใช้ ★★

☐ ปรับปรุง ★

เกณฑ์การประเมิน

คะแนน 17 – 20 อยู่ในเกณฑ์ ดีมาก

คะแนน 14 – 16 อยู่ในเกณฑ์ ดี

คะแนน 11 – 13 อยู่ในเกณฑ์ พอใช้

คะแนน 0 – 10 อยู่ในเกณฑ์ ปรับปรุง

เสร็จแล้ว!!!

รีบทำกิจกรรมต่อไปกัน



ใบความรู้ที่ 1

เรื่อง คำสั่ง While loop

เวลาเรียน 1 ชั่วโมง

การเขียนโปรแกรมแบบวนรอบ หรือ loop นั้น หมายถึง การทำงานกับคำสั่งหรือกลุ่มคำสั่งแบบวนซ้ำหรือทำซ้ำไปจนกว่าเงื่อนไขจะเป็นไปตามที่กำหนดไว้ โดยประโยชน์ของการใช้ loop นั้น ทำให้เขียนโปรแกรมที่ต้องทำงานเดิม ๆ สั้น กระชับ และยืดหยุ่นมากขึ้น

คำสั่งการวนรอบในภาษาซี มี 3 รูปแบบหลัก ได้แก่ 1) while loop 2) do while loop และ 3) for loop ซึ่งทุกรูปแบบเป็นการทำงานซ้ำ ๆ กับกลุ่มคำสั่งทั้งสิ้น แต่มีรายละเอียดการทำงานการใช้งานและเงื่อนไขในการเริ่มต้นหรือหยุดการทำงานแตกต่างกันไป

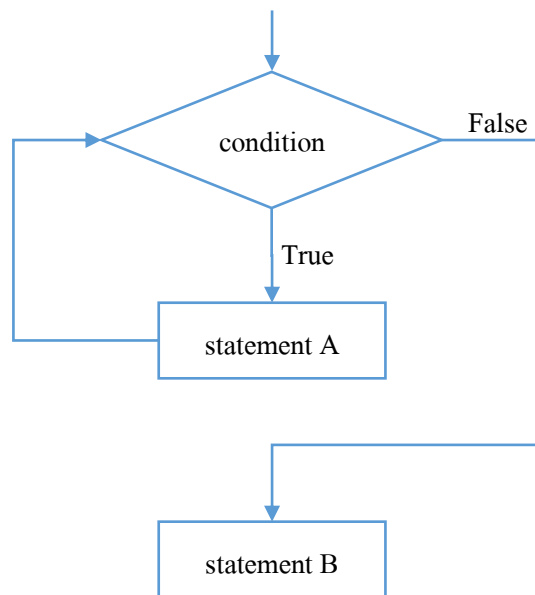
บางโจทย์ปัญหา การใช้งานคำสั่งวนรอบทั้ง 3 แบบ มีรายละเอียดแตกต่างกันเพียงเล็กน้อย แต่บางโจทย์ปัญหาสามารถแก้ไขได้ง่ายกว่าเมื่อใช้คำสั่งการทำงานแบบวนรอบบางอย่าง

ในใบความรู้ที่ 1 นี้ จะอธิบายถึงคำสั่ง while loop เพียงคำสั่งเดียวก่อน

คำสั่ง While loop

การทำงานแบบวนรอบด้วยคำสั่ง while loop จะตรวจสอบเปรียบเทียบข้อมูลหรือค่าของตัวแปรกับเงื่อนไขที่กำหนด โดยการวนรอบการทำงานจะเกิดขึ้นเมื่อผลลัพธ์การเปรียบเทียบนั้นมีค่าเป็นจริง จากนั้นจะทำงานกับกลุ่มคำสั่งที่กำหนด เมื่อสิ้นสุดการทำงานในแต่ละรอบจะเปรียบเทียบเงื่อนไขทุกครั้ง หากผลลัพธ์การเปรียบเทียบยังคงเป็นจริง จะทำงานต่อไปเช่นนี้จนกว่าผลจากการทดสอบข้อมูลกับเงื่อนไขนั้นมีค่าเป็นเท็จ





รูปที่ 1 ฟังก์ชันแสดงการทำงานของ while loop

รูปแบบ

```

while (condition)
{
    statement A
}
statement B
    
```

โดยที่

condition	หมายถึง นิพจน์เงื่อนไขที่ต้องการตรวจสอบ
statement A	หมายถึง ประโยคคำสั่งต่าง ๆ ที่กำหนดให้ทำเมื่อเงื่อนไขเป็นจริง
statement B	หมายถึง ประโยคคำสั่งต่าง ๆ ที่กำหนดให้ทำเมื่อเงื่อนไขเป็นเท็จ

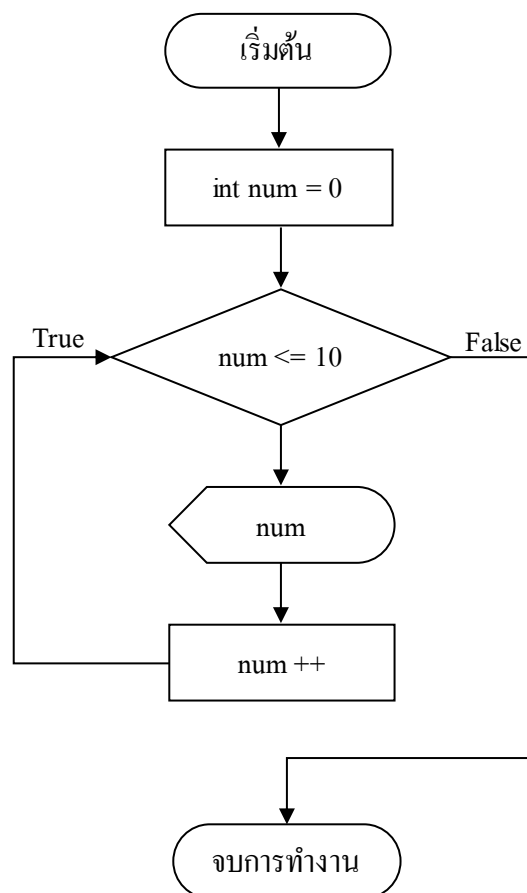


ข้อควรรู้

การทำซ้ำโดย while loop จะไม่มีตัวแปรสำหรับนับรอบ หากต้องการนับรอบ จะต้องสร้างตัวแปรขึ้น และกำหนดการเพิ่มหรือลดค่า



ตัวอย่างที่ 1 แสดงเลขจำนวนเต็ม 1 – 10



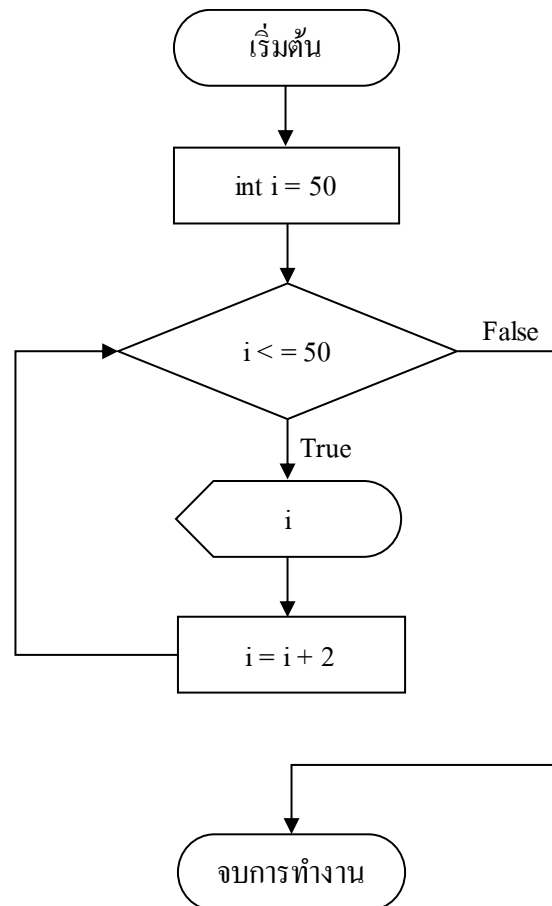
ตัวอย่างที่ 1 โปรแกรม

```
(1) int num = 1;           // ประกาศตัวแปรชนิดจำนวนเต็ม num มีค่าเท่ากับ 1
(2) while (num <= 10) /* ตรวจสอบเงื่อนไขตัวแปร num ว่ามีค่าน้อยกว่าหรือ
                        เท่ากับ 10 หรือไม่ ถ้าเป็นจริง ให้ทำบรรทัดที่ (3) และ (4) */
{
(3)     printf("%d\n", num); // พิมพ์ ค่าตัวแปร num ทางจอภาพ
(4)     num++;              // เพิ่ม ค่าตัวแปร num ขึ้นอีก 1
}                               /* เมื่อเสร็จสิ้นบรรทัดที่ (4) ให้กลับไปบรรทัดที่ (2)
                                เพื่อตรวจสอบเงื่อนไขอีกครั้ง */
}
```

ผลลัพธ์ที่ได้จากการประมวลผล

1
2
3
4
5
6
7
8
9
10

ตัวอย่างที่ 2 ให้แสดงเฉพาะเลขคู่ ระหว่าง 50 – 59



ตัวอย่างที่ 2 โปรแกรม

```

(1) int i = 50;           // ประกาศตัวแปรชนิดจำนวนเต็ม i มีค่าเท่ากับ 50
(2) while (i <= 59)      /* ตรวจสอบเงื่อนไขตัวแปร i ว่ามีค่าน้อยกว่าหรือ
                           เท่ากับ 59 หรือไม่ ถ้าเป็นจริง ให้ทำบรรทัดที่ (3) และ (4)
                           ถ้าเป็นเท็จให้หยุดการทำงาน */
{
  (3) printf("%d\n", i);  // พิมพ์ค่าตัวแปร i แล้วขึ้นบรรทัดใหม่
  (4) i = i+2;            // เพิ่มค่าตัวแปร i ขึ้นอีก 2
}                          /* เมื่อเสร็จสิ้นบรรทัดที่ (4) ให้กลับไปบรรทัดที่ (2)
                           เพื่อตรวจสอบเงื่อนไขอีกครั้ง */
  
```



ผลลัพธ์ที่ได้จากการประมวลผล

50

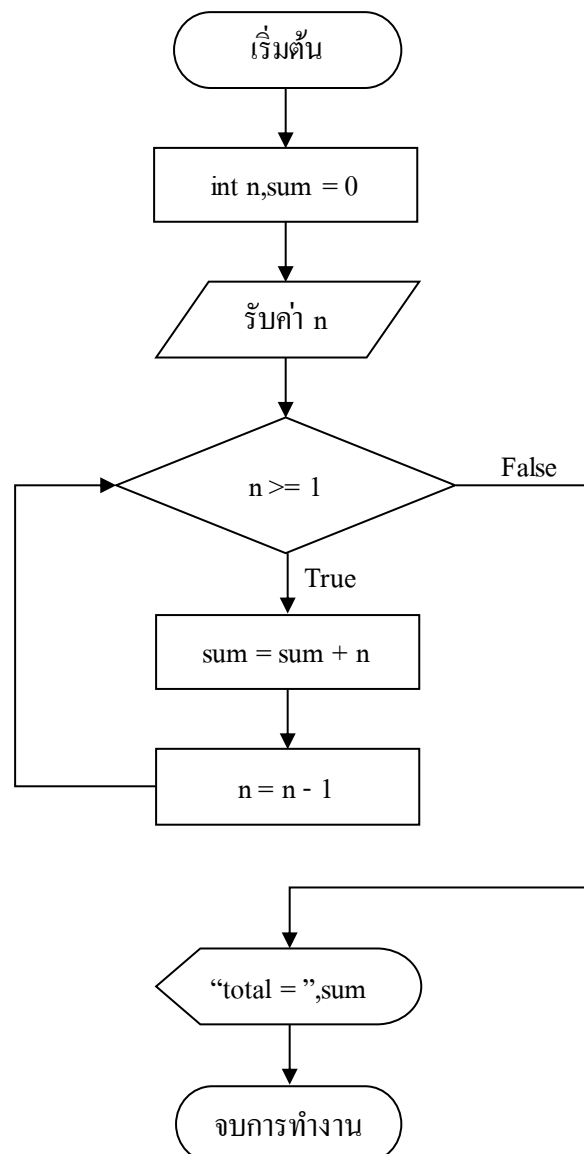
52

54

56

58

ตัวอย่างที่ 3 ให้รับค่าเป็นจำนวนเต็ม จากนั้นหาผลรวมของตัวเลขตั้งแต่ค่านั้น ๆ และลดลงทีละ 1 จนมีค่าเท่ากับ 1 เช่น กรอกค่า 5 ให้หาผลรวมของ $5 + 4 + 3 + 2 + 1$



ตัวอย่างที่ 3 โปรแกรม

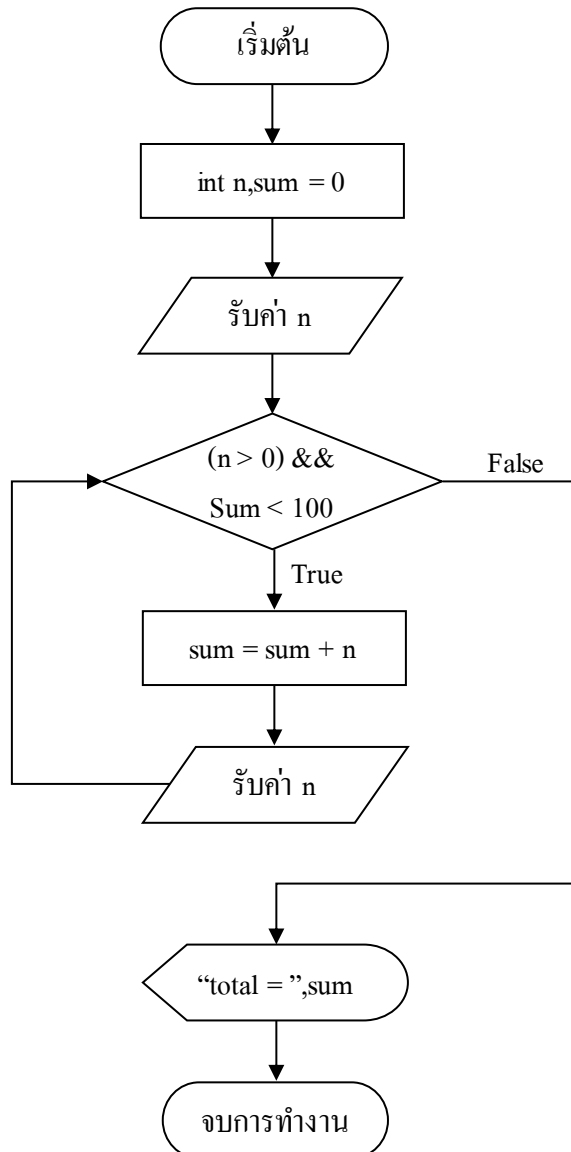
```
(1) int n; // ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ n
(2) int sum = 0; // ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ sum มีค่าเท่ากับ 0
(3) scanf("%d",&n); // รับค่าตัวแปร n เป็นเลขจำนวนเต็ม
(4) while (n >= 1) /* ตรวจสอบเงื่อนไขตัวแปร n ว่ามีค่ามากกว่าหรือเท่ากับ 1
ถ้าเป็นจริง ให้ทำคำสั่งบรรทัดที่ (5) และ (6)
ถ้าเป็นเท็จ ให้ข้ามไปทำคำสั่งบรรทัดที่ (7) */
{
(5) sum = sum+n; /* นำค่าของตัวแปร sum บวกด้วยค่าตัวแปร n จากนั้นบันทึก
ผลลัพธ์ไว้ในตัวแปร sum เช่นเดิม */
(6) n = n - 1; /* ลบค่าตัวแปร n ด้วย 1 จากนั้นบันทึกผลลัพธ์ที่ตัวแปร n
เช่นเดิม */
}
(7) printf("total = %d ",sum); /* เมื่อเสร็จสิ้นคำสั่ง while ทำคำสั่งต่อไป
หลังจาก ช่วง {} ซึ่งกำหนดให้ พิมพ์ข้อความ
total = ค่าตัวแปร sum แบบเลขจำนวนเต็ม
ทางจอภาพ */
```

ผลลัพธ์ที่ได้จากการประมวลผล หาก ถ้า $n = 5$ จะเป็นการคำนวณ $sum = 5+4+3+2+1$

5

total = 15

ตัวอย่างที่ 4 ให้รับค่าเป็นจำนวนเต็ม แล้วหาผลรวมของตัวเลขทั้งหมดที่กรอก โดยให้หยุดการรับค่า เมื่อผลรวมของตัวเลขมากกว่า 100 หรือ เมื่อผู้ใช้กรอกจำนวนที่น้อยกว่าหรือเท่ากับ 0



ตัวอย่างที่ 4 โปรแกรม

```

(1) int n;           // ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ n
(2) int sum = 0;      // ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ sum มีค่าเท่ากับ 0
(3) scanf("%d", &n);  // รับค่าตัวแปร n เป็นเลขจำนวนเต็ม
(8) while ((n > 0) && (sum < 100)) /* ตรวจสอบเงื่อนไขตัวแปร n ว่า
                                   มีค่ามากกว่า 0 และค่าของตัวแปร sum
                                   มากกว่า 100
                                   ถ้าเป็นจริง ให้ทำคำสั่งบรรทัดที่ (5) และ (6)
                                   ถ้าเป็นเท็จ ให้ข้ามไปทำคำสั่งบรรทัดที่ (8) */
(4) {
(5)     sum = sum+n; /* นำค่าของตัวแปร sum บวกด้วยค่าตัวแปร n จากนั้นบันทึก
                                   ผลลัพธ์ไว้ในตัวแปร sum เช่นเดิม */
(6)     scanf("%d", &n); // รับค่าตัวแปร n ใหม่
(7) }
(8) printf("total = %d ",sum); /* เมื่อเสร็จสิ้นคำสั่ง while ทำคำสั่งต่อไป
                                   หลังจาก ช่วง {} ซึ่งกำหนดให้ พิมพ์ข้อความ
                                   total = ค่าตัวแปร sum แบบเลขจำนวนเต็ม
                                   ทางจอภาพ */

```

ผลลัพธ์ที่ได้จากการประมวลผล ตามตัวอย่างการกรอกข้อมูล เช่น

```

15
50
24
37
-2
total = 126

```

สรุปเนื้อหา

1. ชุดคำสั่งภายใน while loop จะถูกทำงานเมื่อตรวจสอบเงื่อนไขแล้วเป็นจริง
2. while loop จบการทำงานเมื่อตรวจสอบเงื่อนไขแล้วเป็นเท็จ
3. เงื่อนไขหรือนิพจน์ที่นำมาตรวจสอบ สามารถใช้ตัวดำเนินการเปรียบเทียบ หรือตัวดำเนินการทางตรรกะได้

ใบกิจกรรมที่ 1

เรื่อง คำสั่ง While loop

คำชี้แจง ใบกิจกรรมที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

1. ข้อใด คือ ลักษณะการทำงานของคำสั่ง while loop
 - ก. ทำงานและตรวจสอบเงื่อนไขพร้อมกัน
 - ข. ทำงานตามคำสั่งก่อนจึงตรวจสอบเงื่อนไข
 - ค. ตรวจสอบเงื่อนไขก่อนแล้วจึงทำงานต่อไป
 - ง. ตรวจสอบเงื่อนไขก่อนก็ได้ หรือทำงานก่อนก็ได้
2. ข้อใดกล่าวถึงคำสั่ง while loop ได้ถูกต้อง
 - ก. การทำงานซ้ำ ๆ ที่มีจำนวนรอบที่แน่นอน
 - ข. ลูปจะทำงานวนไปจนกว่าเงื่อนไขจะเป็นเท็จ
 - ค. หากเงื่อนไขของลูปเป็นเท็จ โปรแกรมจะทำงานตามคำสั่งที่อยู่ภายในลูป
 - ง. หากเงื่อนไขของลูปจะเป็นจริงในครั้งแรก จะไม่มีการทำงานตามคำสั่งใด ๆ
3. หาก while loop มีคำสั่งวนซ้ำ หลายบรรทัด ต้องใช้เครื่องหมายใด

ก. []	ข. < >
ค. ()	ง. { }
4. จากโปรแกรมต่อไปนี้ จะเกิดการซ้ำของ while loop จำนวนกี่ครั้ง


```
int n = 3;
while (n > 3)
{
    printf("%d", n);
}
```

ก. 3	ข. 4
ค. 5	ง. ไม่เกิดการซ้ำ



10. จากโปรแกรมจะได้ผลลัพธ์ตรงกับข้อใด

```
int a = 1, b = 2, c = 3;
while(a <= 10)
{
    if((a<=5) && (c>=2))
    {
        a = b + 8;
    }
    b = a * 2;
    c = b - 3;
    a++;
}
printf("A = %d, B = %d, C = %d",a,b,c);
```

- ก. A = 1, B = 2, C = 3
- ข. A = 10, B = 20, C = 17
- ค. A = 11, B = 20, C = 17
- ง. A = 11, B = 22, C = 19

คะแนนในกิจกรรมที่ 1

คะแนนเต็ม	10
คะแนนที่ได้	



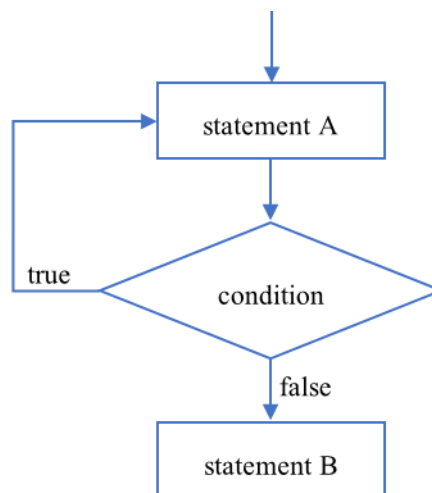
ใบความรู้ที่ 2

เรื่อง คำสั่ง Do-while loop

เวลาเรียน 1 ชั่วโมง

คำสั่ง Do-while loop

คำสั่ง do while loop จะทำการตรวจสอบเงื่อนไขภายหลังจากการทำงานในลูป โดยโปรแกรมจะทำงาน ในลูปซ้ำไปเรื่อย ๆ ถ้าตรวจสอบเงื่อนไขแล้วพบว่าเป็นจริงจะทำงานซ้ำต่อไป จนกระทั่งผลลัพธ์ของการเปรียบเทียบเงื่อนไขนั้นเป็นเท็จ จึงหยุดการทำงานของลูป ดังนั้นจึงทำให้มีการทำคำสั่งในลูป อย่างน้อย 1 ครั้งเสมอ กล่าวคือ การทำงานวนรอบแบบ do while จะทำคำสั่งในลูปแล้วจึงทำการตรวจสอบเงื่อนไขภายหลัง



รูปที่ 2 ฟังก์ชัน คำสั่ง do while loop



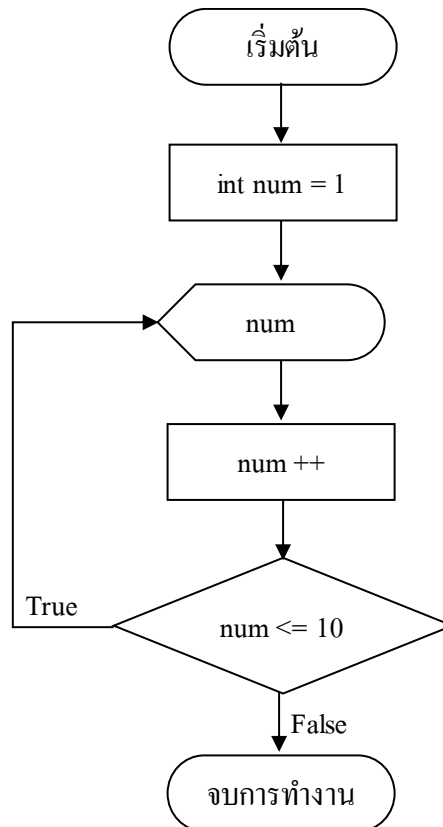
รูปแบบ

```
do
{
    statement A
}
while (condition);
    statement B
```

โดยที่

condition	หมายถึง นิพจน์เงื่อนไขที่ต้องการตรวจสอบ
statement A	หมายถึง ประโยคคำสั่งต่าง ๆ ที่กำหนดให้ทำเมื่อเงื่อนไขเป็นจริง
statement B	หมายถึง ประโยคคำสั่งต่าง ๆ ที่กำหนดให้ทำเมื่อเงื่อนไขเป็นเท็จ

ตัวอย่างที่ 1 ให้แสดงผลตัวเลข 1 - 10



ตัวอย่างที่ 1 โปรแกรม

```

(1) int num = 1;           // ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num มีค่าเท่ากับ 1
(2) do
    {
(3)     printf("%d\n", num); /* พิมพ์ค่าตัวแปร num ทางจอภาพ
                             แล้วขึ้นบรรทัดใหม่ */
(4)     num ++;             // เพิ่ม ค่าตัวแปร num ขึ้นอีก 1
    }
(5) while(num<=10);        /* ตรวจสอบเงื่อนไขตัวแปร num มีค่า
                             น้อยกว่าหรือเท่ากับ 10
                             ถ้าเป็นจริง ให้ทำซ้ำบรรทัดที่ (3) และ (4)
                             ถ้าเป็นเท็จ ให้จบการทำงาน */
  
```



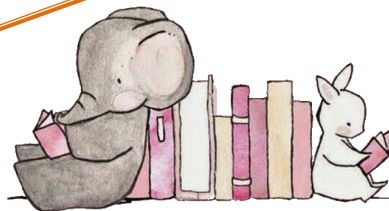
จากตัวอย่างที่ 1 โปรแกรมจะทำการประมวลผลคำสั่งภายในเครื่องหมาย {} ก่อนหนึ่งครั้ง แล้วจึงทำการตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่ ถ้าเป็นจริงก็จะวนกลับไปทำคำสั่งอีกครั้ง ทำอย่างนี้ไปจนกว่า การตรวจสอบเงื่อนไขหลังคำสั่ง while จะเป็นเท็จ จึงจบการทำงาน

ผลลัพธ์ที่ได้จากการประมวลผล

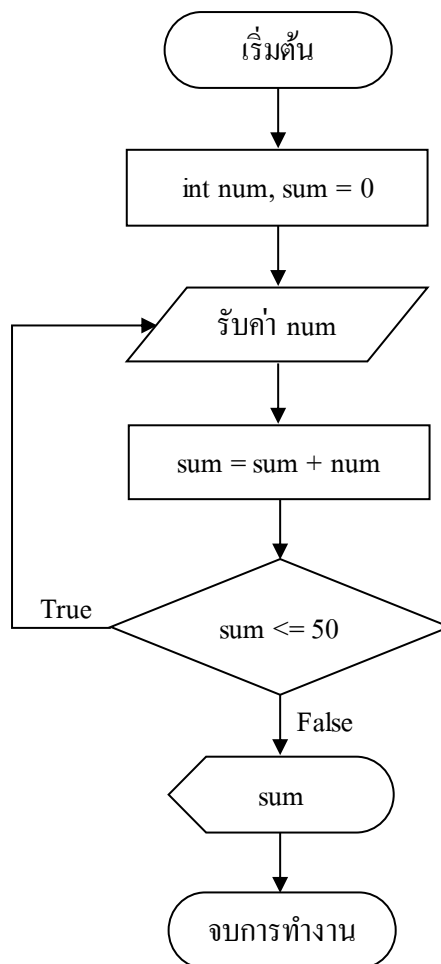
1
2
3
4
5
6
7
8
9
10

ข้อควรระวัง

การใช้ while กับ do while หากลืมใส่คำสั่งที่ทำให้เงื่อนไขเป็นเท็จ โปรแกรมจะทำงานไม่หยุด และหากต้องการหยุดการทำงานระหว่างที่โปรแกรมทำงาน นักเรียนต้องกด Ctrl + Break บนแป้นพิมพ์



ตัวอย่างที่ 2 ให้รับค่าจำนวนเต็มจากแป้นพิมพ์ จากนั้นหาผลรวมของตัวเลข จนกว่าผลรวมจะมีค่ามากกว่า 50



ตัวอย่างที่ 2 โปรแกรม

```
(1) int num, sum = 0;    /* ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num และ ชื่อ sum
                           ให้มีค่าเท่ากับ 0 */

(2) do
{
(3)     scanf("%d",&num);    // รับค่าตัวแปร num เป็นชนิดเลขจำนวนเต็ม
(4)     sum = sum+num;        /* นำค่าตัวแปร sum บวกค่าตัวแปร num
                           แล้วบันทึกค่าเก็บไปยังตัวแปร sum เช่นเดิม */
}

(5) while(sum<=50);    /* ตรวจสอบเงื่อนไขตัวแปร num มีค่าน้อยกว่าหรือเท่ากับ
                           ค่า 50 หรือไม่ ถ้าเป็นจริง ให้กลับไปทำคำสั่งในบรรทัดที่ (3)
                           และ (4) ถ้าเป็นเท็จ ให้ออกจากลูปแล้วประมวลผลคำสั่งใน
                           บรรทัดที่ (6) ต่อ */

(6) printf("value of sum is %d", sum);    /* พิมพ์ค่าตัวแปร sum แบบเลขจำนวนเต็ม
                                           */
```

จากตัวอย่างที่ 2 โปรแกรมจะทำการประมวลผลคำสั่ง รับค่าให้กับตัวแปร num และคำสั่ง `sum = sum+num` ก่อน แล้วจึงทำการตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่ ถ้าเป็นจริง ก็จะวนกลับไปทำคำสั่ง (statement) ที่อยู่ในลูปอีกครั้ง ทำเช่นนี้ไปจนกว่าการตรวจสอบเงื่อนไขจะเป็นเท็จ จึงจะจบการทำงาน



วิธีคิดหาค่าของตัวแปร num และตัวแปร sum ของการทำงานในลูปแต่ละครั้ง เมื่อรับค่า num จากแป้นพิมพ์

num	sum ก่อนการบวก	การคำนวณ sum+num	sum หลังการบวก	เงื่อนไข sum<=50
5	0	$0 + 5$	5	จริง
15	5	$15 + 5$	20	จริง
10	20	$20 + 10$	30	จริง
25	30	$30 + 25$	55	เท็จ

ผลลัพธ์ที่ได้จากการประมวลผล ตามตัวอย่างการรับค่าตัวแปร num

5

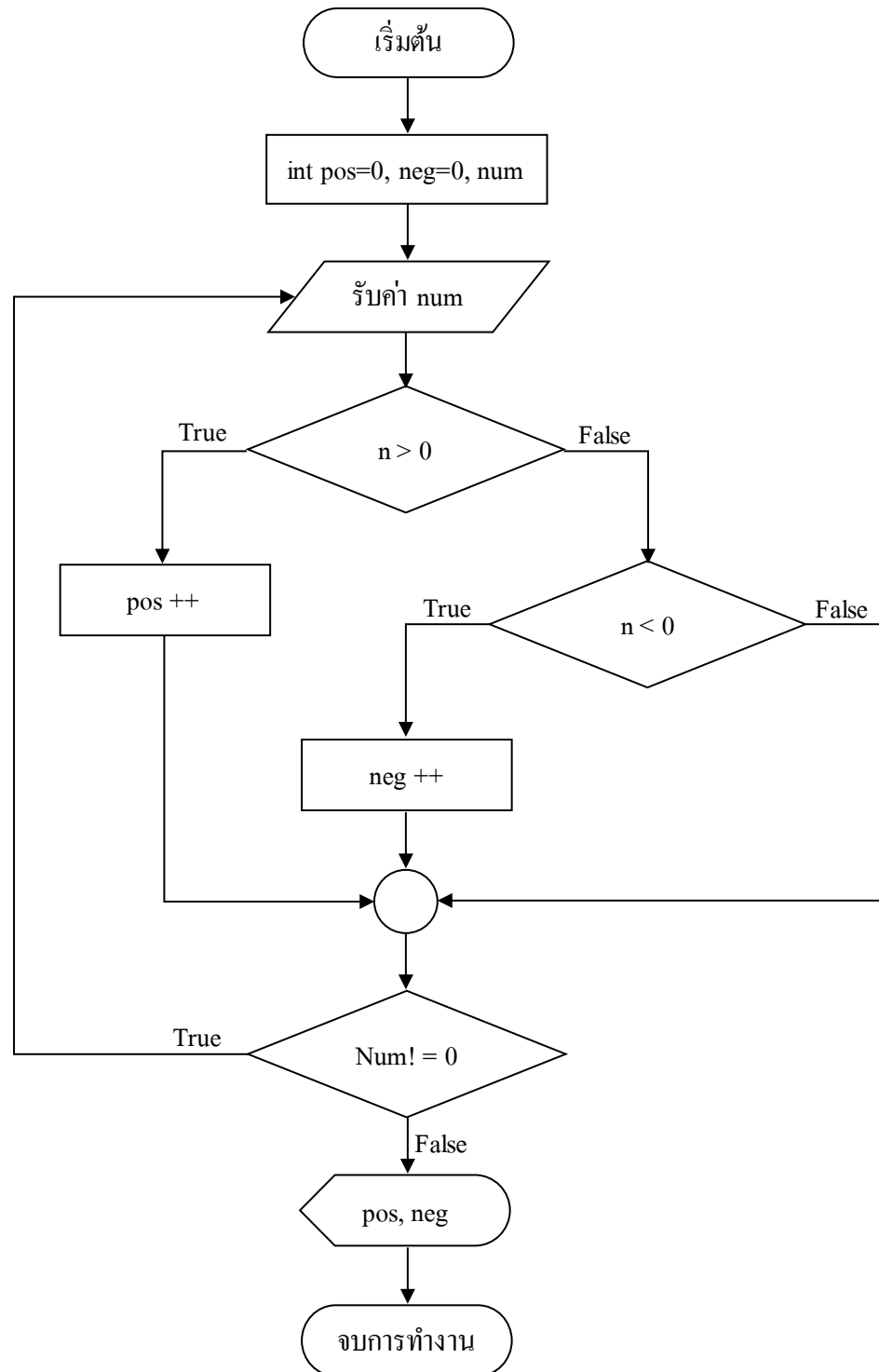
15

10

25

value of sum is 55

ตัวอย่างที่ 3 ให้รับค่าจำนวนเต็มจากแป้นพิมพ์ จนกว่าผู้ใช้จะกรอกตัวเลข 0 จากนั้นให้แสดงว่ามีจำนวนเต็มบวกกี่จำนวน และจำนวนเต็มลบกี่จำนวน



ตัวอย่างที่ 3 โปรแกรม

```

(1) int pos=0, neg=0, num;      /* ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ pos และ
                                ชื่อ neg กำหนดค่าเริ่มต้นเป็น 0 สำหรับนับ
                                จำนวนเต็ม และตัวแปรชื่อ num ใช้ในการรับค่า
                                จากผู้ใช้ */

(2) do
    {
(3)     scanf("%d",&num);      // รับค่าเป็นจำนวนเต็ม เก็บไว้ที่ตัวแปร num
(4)     if (num>0)              // ถ้าค่าที่รับมามากกว่า 0 ให้ทำบรรทัดที่ (5)
(5)         pos++;              // เพิ่มค่าในตัวแปร pos ขึ้นอีก 1
(6)     else if (num<0)         // ถ้าค่าที่รับมาน้อยกว่า 0 ให้ทำบรรทัดที่ (7)
(7)         neg++;              // เพิ่มค่าในตัวแปร neg ขึ้นอีก 1
(8) }while(num != 0);           /* ตรวจสอบเงื่อนไขตัวแปร num ว่ามีค่าไม่เท่ากับ 0
                                หรือไม่ ถ้าค่าไม่เท่ากับ 0 เป็นจริง ให้ทำซ้ำตั้งแต่
                                บรรทัดที่ 3 ถ้าค่าเท่ากับ 0 ให้จบการทำงาน
                                แล้วทำบรรทัดที่ (9) ต่อไป */

(9) printf("positive number =%d\n", pos);    /* พิมพ์ว่าผู้ใช้ได้กรอกจำนวนเต็ม
                                                ที่มีค่าบวกไปทั้งหมดกี่จำนวน */

(10)printf("negative number =%d\n", neg);    /* พิมพ์ว่าผู้ใช้ได้กรอกจำนวนเต็ม
                                                ที่มีค่าลบไปทั้งหมดกี่จำนวน */

```

จากตัวอย่างที่ 3 โปรแกรมจะทำการประมวลผลคำสั่ง ตั้งแต่บรรทัดที่ (3) จนถึงบรรทัดที่ (7) จากนั้นจึงตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่ ถ้าเป็นจริงก็จะวนกลับไปทำคำสั่ง (statement) ที่อยู่ในลูป อีกครั้ง ทำอย่างนี้ไปจนกว่าการตรวจสอบเงื่อนไขจะเป็นเท็จจึงจบการทำงานในส่วน ของลูปแล้วทำงานในส่วนต่อไป คือตั้งแต่บรรทัดที่ (9) เป็นต้นไป



วิธีคิดคำนวณของการทำงานในลูปแต่ละครั้ง เมื่อรับค่า num จากแป้นพิมพ์ ตามตัวอย่าง

num	pos	neg	เงื่อนไข num != 0
5	1	0	จริง
-9	1	1	จริง
-26	1	2	จริง
14	2	2	จริง
378	3	2	จริง
0	3	2	เท็จ

ผลลัพธ์ที่ได้จากการประมวลผล

5

-9

-26

14

378

0

positive number =3

negative number ==2

สรุปเนื้อหา คำสั่ง do while loop

คำสั่ง do while loop จะมีการทำงานคล้ายกับ while loop มีความแตกต่างตรงที่ คำสั่ง do..while loop จะตรวจสอบเงื่อนไขหลังการทำงานตามคำสั่ง (statement) นั่นคือ โปรแกรมจะประมวลผลคำสั่งภายในลูปอย่างน้อย 1 ครั้งเสมอ แล้วจึงตรวจสอบเงื่อนไขทีหลัง



ศึกษาจนเข้าใจแล้ว!!!

ไปทำกิจกรรมกันนะคะ



ใบกิจกรรมที่ 2

เรื่อง คำสั่ง Do-while loop

คำชี้แจง ใบกิจกรรมที่ 2 เป็นแบบทดสอบอัตรัยการเขียนโปรแกรมโดยใช้คำสั่ง Do-while loop จำนวน 2 ข้อ 10 คะแนน เวลา 10 นาที

ข้อที่ 1

คำสั่ง ให้นักเรียนเขียนผลลัพธ์ที่ได้จากส่วนของโปรแกรมนี้ เมื่อผู้ใช้กรอกข้อมูล 7000

```
int i=0;
float pop;
scanf("%f", &pop);
do{
    pop *= 1.1;
    i++;
    printf("in %d year, population = %.2f\n", i, pop);
}while(pop<10000);
```

ตอบ

.....

.....

.....

.....

.....

.....

ข้อที่ 2

คำสั่ง ให้นักเรียนเขียนโปรแกรมเพื่อรับค่าตัวเลขจำนวนเต็มไปจนกว่าผู้ใช้จะตอบ 'N'
แล้วแสดงผลรวมของเลขทั้งหมดที่กรอกนั้น

ตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คะแนนในกิจกรรมที่ 2		
รายการ	คะแนนเต็ม	คะแนนที่ได้
ตอนที่ 1	5	
ตอนที่ 2	5	



ใบความรู้ที่ 3 เรื่อง คำสั่ง For loop

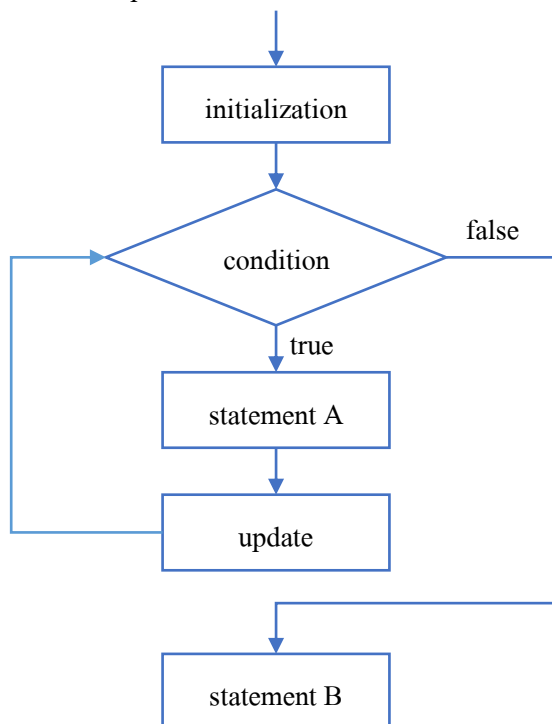
เวลาเรียน 1 ชั่วโมง

คำสั่ง For loop

การทำงานวนรอบแบบ for loop เหมาะกับโจทย์ปัญหาที่ทราบจำนวนรอบที่แน่นอน โดยจำนวนรอบอาจเป็นค่าคงที่ หรือค่าของตัวแปรชนิดจำนวนเต็มก็ได้ ซึ่งเงื่อนไขในการทำงาน จะทำในขณะที่ผลลัพธ์การเปรียบเทียบเงื่อนไขเป็นจริง

นอกจากนี้ยังทำงานได้โดยใช้ตัวแปรนับรอบ ซึ่งสามารถกำหนดค่าเริ่มต้นของตัวแปร ที่ใช้ในการนับรอบได้ โดย for loop มีส่วนประกอบอยู่ 3 ส่วน แต่ละส่วน คั่นด้วยเครื่องหมาย semicolon (;) ได้แก่

1. ส่วนเริ่มต้น (Initialization)
2. ส่วนเงื่อนไข (Condition)
3. ส่วนปรับค่า (Update)



รูปที่ 3 แสดงผังงานแสดงการทำงานคำสั่ง for loop



รูปแบบ

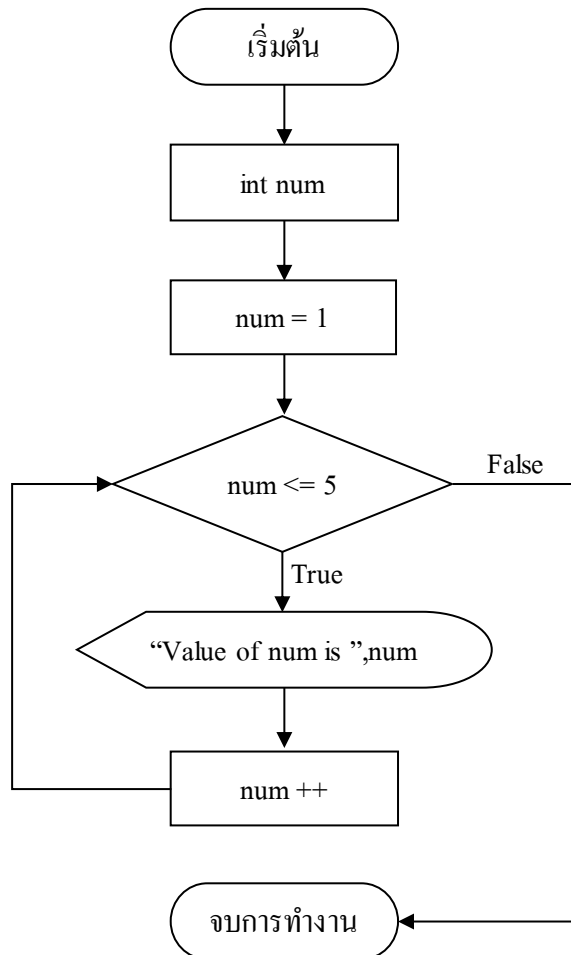
```
for(initialization; condition; update)
{
    statement A;
    ....
}
statement B;
```

โดยที่

initialization	หมายถึง	ส่วนกำหนดค่าให้กับตัวแปร ซึ่งจะทำเพียงครั้งเดียว ก่อนที่จะทำงานกับกลุ่มคำสั่งใน for loop
condition	หมายถึง	นิพจน์เงื่อนไขที่ต้องการตรวจสอบ
update	หมายถึง	การเปลี่ยนค่าของตัวแปร โดยจะทำทุกครั้งหลังจากเสร็จสิ้นกลุ่มคำสั่งใน for loop และก่อนที่จะเปรียบเทียบเงื่อนไขที่กำหนด
statement A	หมายถึง	ประโยคคำสั่งต่าง ๆ ที่ต้องการให้ทำซ้ำ
statement B	หมายถึง	ประโยคคำสั่งที่ให้ทำต่อหลังจากเสร็จสิ้นการทำงานแบบวนรอบ



ตัวอย่างที่ 1 โปรแกรมแสดงเลขจำนวนเต็ม 1 - 5



```

(1) int num; // ประกาศตัวแปรชนิดเลขจำนวนเต็ม ชื่อ num
(2) for(num =1;num<=5;num++)/* ค่าเริ่มต้น คือ ตัวแปร num มีค่าเท่ากับ 1
                               เงื่อนไข คือ ตัวแปร num มีค่าน้อยกว่าหรือเท่ากับ 5
                               การเพิ่มค่าตัวแปร คือ ตัวแปร num เพิ่มค่าครั้งละ 1*/
{
(3)   printf("Value of num is %d\n", num); /*พิมพ์ข้อความ Value of num is
                                           และค่าตัวแปร num ทางจอภาพ*/
}
  
```



ผลลัพธ์ที่ได้จากการประมวลผล

Value of num is 1

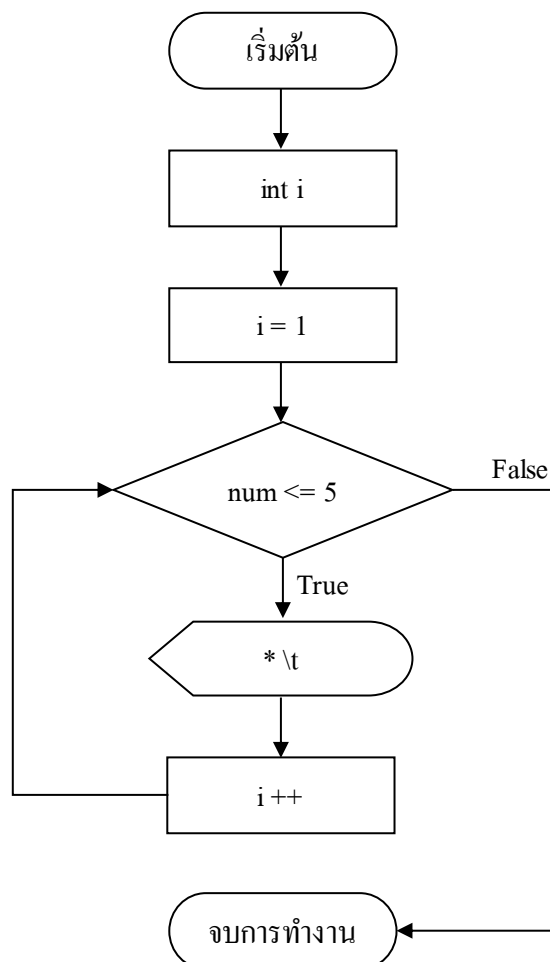
Value of num is 2

Value of num is 3

Value of num is 4

Value of num is 5

ตัวอย่างที่ 2 โปรแกรมแสดงตัวอักษร (*) ทางจอภาพ 5 ตัว โดยเว้นระยะเท่าๆ
สำหรับแต่ละอักขระ



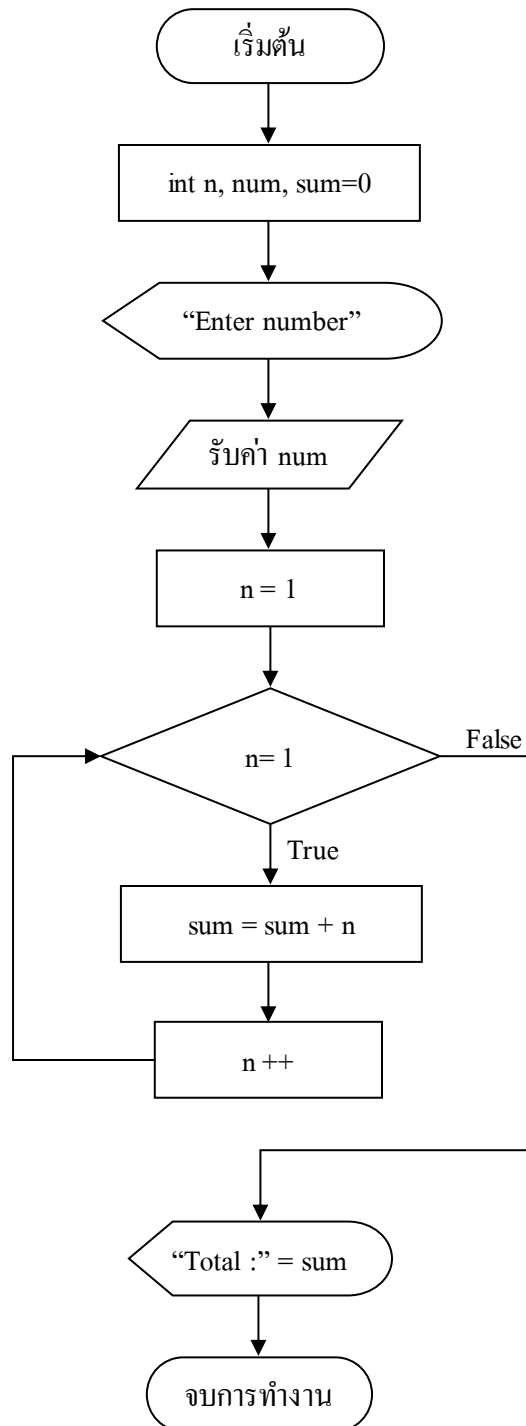
ตัวอย่างที่ 2 โปรแกรม

```
(1) int i;           // ประกาศตัวแปร ชนิดเลขจำนวนเต็ม ชื่อ i
(2) for (i =1; i <= 5; i++) /* ค่าเริ่มต้น คือ ตัวแปร i มีค่าเท่ากับ 1
                           เจื่อนไข คือ ตัวแปร i มีค่าน้อยกว่าหรือเท่ากับ 5
                           การเพิ่มค่าตัวแปร คือ ตัวแปร i เพิ่มค่าครั้งละ 1*/
{
(3)     printf("%t"); //พิมพ์ข้อความ * ทางจอภาพ แล้วเว้นระยะ 1 แท็บ
}
```

ตัวอย่างที่ 2 ผลลัพธ์ที่ได้จากการประมวลผล

```
*      *      *      *      *
```

ตัวอย่างที่ 3 ให้รับค่า n จากแป้นพิมพ์ จากนั้นคำนวณหาผลรวมของ $1+2+3+\dots+n$ แล้วแสดงผลลัพธ์ทางจอภาพ



ตัวอย่างที่ 3 โปรแกรม

```

(1) int num, n, sum = 0; /* ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num และ n
                           และตัวแปร sum กำหนดให้มีค่าเท่ากับ 0

(2) printf("Enter number : ") // แสดงข้อความเพื่อให้กรอกตัวเลข

(3) scanf("%d",&num); // รับค่าตัวแปร num เป็นชนิดเลขจำนวนเต็ม

(4) for(n = 1; n<=num; n++) /* ค่าเริ่มต้น คือ ตัวแปร n มีค่าเท่ากับ 1
                              เงื่อนไข คือ ตัวแปร n มีค่าน้อยกว่าหรือเท่ากับ num
                              การเพิ่มค่าตัวแปร คือ ตัวแปร n เพิ่มค่าครั้งละ 1 */

{
(5)     sum = sum+n; /* นำค่าตัวแปร sum บวกค่าตัวแปร n
                     แล้วบันทึกผลลัพธ์ที่ได้ไปยังตัวแปร sum ดังเดิม */

}

(6) printf("Total: %d", sum); // พิมพ์ผลลัพธ์ค่าตัวแปร sum แบบเลขจำนวนเต็ม
  
```

ในกำหนดค่าเริ่มต้นของตัวแปร n ให้เป็น 1 โดยจะทำก่อนเริ่มการทำงานของ loop เพียงครั้งเดียว จากนั้นตรวจสอบเงื่อนไขตัวแปร num มีค่าน้อยกว่าหรือเท่ากับ n หรือไม่ ถ้าเป็นจริงให้ทำคำสั่งในบรรทัดที่ (5) ถ้าเป็นเท็จให้จบการทำงานของ loop แล้วทำงานคำสั่งในบรรทัดที่ (6) ต่อไป โดยทุกครั้งที่ทำงานใน loop เสร็จสิ้น และก่อนที่จะตรวจสอบเงื่อนไขครั้งใหม่จะปรับค่าของตัวแปร n เพิ่มขึ้นทีละ 1

วิธีคิดหาค่าของตัวแปร n และตัวแปร sum ของการทำงานในลูปแต่ละครั้ง เมื่อรับค่า num เป็น 5 จากแป้นพิมพ์

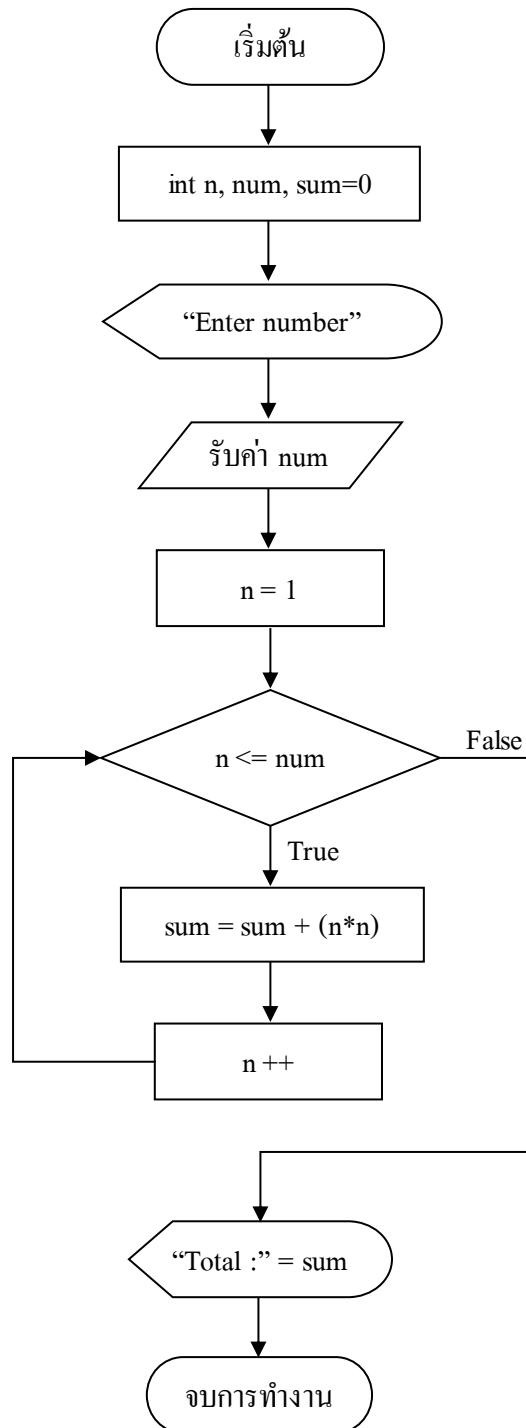
n	การคำนวณ sum	sum	ปรับค่า n	เงื่อนไข $n \leq num$
1	$0+1$	1	$1+1$	จริง
2	$1+2$	3	$2+1$	จริง
3	$3+3$	6	$3+1$	จริง
4	$6+4$	10	$4+1$	จริง
5	$10+5$	15	$5+1$	เท็จ
6	เงื่อนไขเป็นเท็จ จบการทำงาน			

ผลลัพธ์ที่ได้จากการประมวลผล เมื่อรับค่า num เป็น 5

Enter number : 5

Total : 15

ตัวอย่างที่ 4 การคำนวณหาค่า $1^2+2^2+3^2+...+n^2$ โดยรับค่า num จากแป้นพิมพ์



ตัวอย่างที่ 4 โปรแกรม

```

(1) int num, n, sum = 0; /* ประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num และ n
                        และตัวแปร sum กำหนดให้มีค่าเท่ากับ 0
(2) printf("Enter number : ") // แสดงข้อความเพื่อให้กรอกตัวเลข
(3) scanf("%d",&num); // รับค่าตัวแปร num เป็นชนิดเลขจำนวนเต็ม
(4) for(n = 1; n<=num; n ++ ) /* ค่าเริ่มต้น คือ ตัวแปร n มีค่าเท่ากับ 1
                                เงื่อนไข คือ ตัวแปร n มีค่าน้อยกว่าหรือเท่ากับ num
                                การเพิ่มค่าตัวแปร คือ ตัวแปร n เพิ่มค่าครั้งละ 1 */
{
(5)     sum = sum+(n*n); /* นำค่าตัวแปร n ยกกำลัง 2 แล้วนำผลที่ได้มาบวก
                        ค่าตัวแปร sum แล้วบันทึกผลลัพธ์ที่ได้ไปยัง
                        ตัวแปร sum ดังเดิม */
}
(6) printf("Total: %d", sum); // พิมพ์ผลลัพธ์ค่าตัวแปร sum แบบเลขจำนวนเต็ม
  
```

จากตัวอย่างที่ 4 โปรแกรมจะทำการประมวลผลคำสั่ง $sum = sum + (n * n)$ และ $n ++$; แล้วทำการตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่ ถ้าเป็นจริงก็จะวนกลับไปทำคำสั่ง (statement) ที่อยู่ในลูป อีกครั้ง ทำอย่างนี้ไปจนกว่าการตรวจสอบเงื่อนไขจะเป็นเท็จจึงจบการทำงาน

วิธีคิดหาค่าตัวแปร n และตัวแปร sum ของการทำงานในลูปแต่ละครั้ง เมื่อรับค่า num เป็น 5 จากแป้นพิมพ์

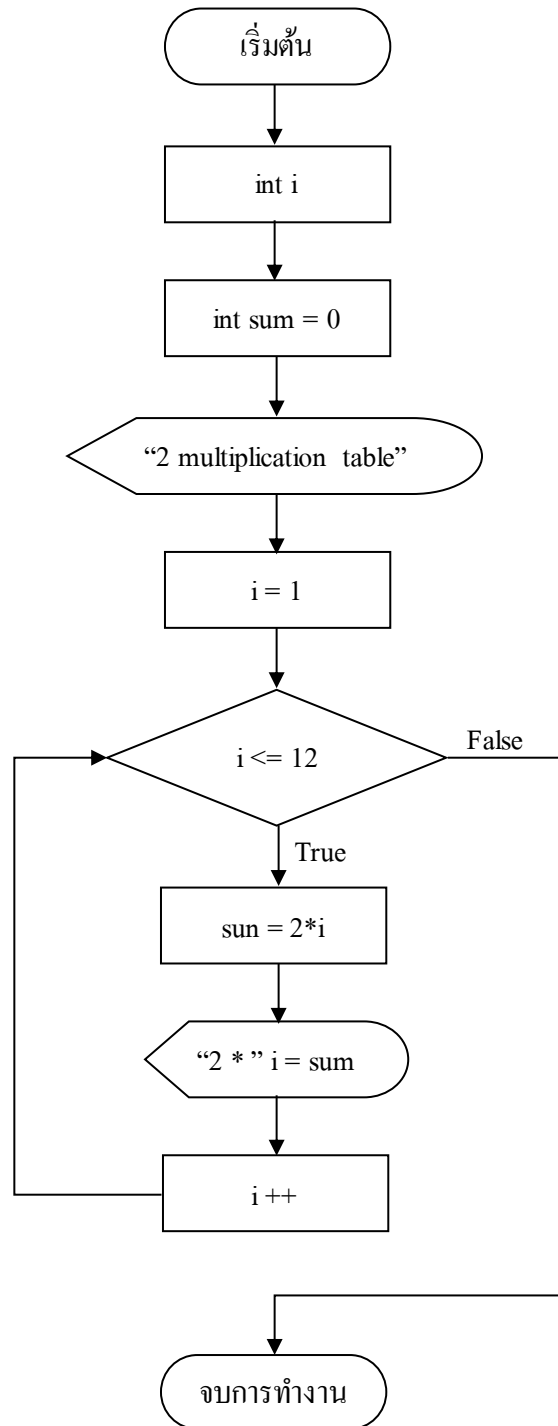
n	การคำนวณ sum	sum	การคำนวณ n	เงื่อนไข $n \leq \text{num}$
1	$0 + (1 * 1)$	1	$1 + 1$	จริง
2	$1 + (2 * 2)$	5	$2 + 1$	จริง
3	$5 + (3 * 3)$	14	$3 + 1$	จริง
4	$14 + (4 * 4)$	30	$4 + 1$	จริง
5	$30 + (5 * 5)$	55	$5 + 1$	เท็จ
6	เงื่อนไขเท็จจบการทำงาน			

ผลลัพธ์ที่ได้จากการประมวลผล

Enter Number : 5

Total : 55

ตัวอย่างที่ 5 โปรแกรมแสดงสูตรคูณแม่ 2



ตัวอย่างที่ 5 โปรแกรม

```

(1) int i, sum=0;          /* ประกาศตัวแปร ชนิดจำนวนเต็ม ชื่อ i และ sum
                           กำหนดตัวแปร sum มีค่าเท่ากับ 0 */

(2) printf("2 Multiplication Table");    /* พิมพ์ข้อความ 2 Multiplication Table
                                           ทางจอภาพ */

(3) for (i=1; i<=12; i++)    /* ค่าเริ่มต้น คือ ตัวแปร i มีค่าเท่ากับ 1
                               เงื่อนไข คือ ตัวแปร i มีค่าน้อยกว่าหรือเท่ากับ 12
                               การเพิ่มค่าตัวแปร คือ ตัวแปร i เพิ่มค่าครั้งละ 1 */

(4) {

(5)     sum = 2*i;          /* 2 คูณค่าตัวแปร i แล้วส่งค่าเก็บไปยังตัวแปร sum
                               ทางซ้าย */

(6)     printf("2 * %d = %d\n",i,sum); /* พิมพ์ 2 ค่าตัวแปร i, sum เป็นเลขจำนวนเต็ม
                                           ในรูปแบบ 2 * i = sum ทางจอภาพ */

(7) }

```

ผลลัพธ์ที่ได้จากการประมวลผล

2 Multiplication Table

```

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
2 * 11 = 22
2 * 12 = 24

```

สรุปเนื้อหา คำสั่ง For loop

คำสั่ง for loop จะมีการทำงานเหมือนกับคำสั่ง while คือตรวจสอบเงื่อนไข หรือทำการเปรียบเทียบก่อน เมื่อเป็นจริงจึงทำคำสั่งภายในลูป แตกต่างตรงที่ for loop นิยมใช้ในกรณีที่ทราบจำนวนครั้งแน่นอนที่ต้องการวนซ้ำ

ไม่ยากเลยใช่ไหมคะ
ไปทำกิจกรรมกันเลยคะ



ใบกิจกรรมที่ 3

เรื่อง คำสั่ง For loop

คำชี้แจง ใบกิจกรรมที่ 1 มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 5 ข้อ 5 คะแนน เวลา 5 นาที

ตอนที่ 2 เป็นแบบเติมคำตอบแบบสั้น จำนวน 5 ข้อ 5 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 5 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

ให้นักเรียนพิจารณาโปรแกรมแสดงสูตรคูณ โดย รับค่า n แทนแม่สูตรคูณต่อไปนี้
เพื่อตอบในข้อที่ 1 - 3

```
(1) int i,n,sum=0;
(2) printf("Enter mulitplication number : ");
(3) .....1.....;
(4) .....2.....
(5) {
(6)     printf("%d * %d = %d\n",.....3.....);
(7) }
```

1. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 1

- ก. scanf("%d", &n)
- ข. scanf("%d", &sum)
- ค. printf("%d", n)
- ง. printf("%d", sum)

2. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 2

ก. for (i<=12; i++)

ข. for (i<=12; i=1)

ค. for (i=1; i<=12; i++)

ง. for (i++; i<=12; i=1)

3. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 3

ก. i, n, i*i

ข. i, n, n*i

ค. n, i, n*n

ง. n, i, n²

4. จากผลลัพธ์ที่ได้จากการประมวลผล ควรเลือกคำสั่งในข้อใด

1 + 5 = 6

2 + 4 = 6

3 + 3 = 6

4 + 2 = 6

ก. for(i=1, j=5; i<5; i++, j--)

printf("%d + %d = %d\n", i, j, i+j);

ข. for(i=1, j=5; i<=5; i++, j++)

printf("%d + %d = %d\n", i, j, i+j);

ค. for(i=0, j=5; i<5; i++, j--)

printf("%d + %d = %d\n", i, j, i+j);

ง. for(i=0, j=5; i<=5; i++, j++)

printf("%d + %d = %d\n", i, j, i+j);



5. จากคำสั่งต่อไปนี้ได้ผลลัพธ์ตรงกับข้อใด

```
int num, sum=0;
for(num=1; num<=5; num=num+2){
    sum = sum +num;
}
printf("sum = %d\n",sum);
```

ก. 3

ข. 7

ค. 9

ง. 15

ตอนที่ 2 เป็นแบบเติมคำตอบแบบสั้น จำนวน 5 ข้อ

คำสั่ง จากโปรแกรมด้านล่างนี้ ให้พิจารณาค่าต่าง ๆ ตามที่โจทย์กำหนดในแต่ละข้อ

```
(1) int i, num, sum=0;
(2) scanf("%d", &num);
(3) if(num>5&&num<10)
(4)     for(i=1; i<=5; i=i+2){
(5)         sum = sum +i;    }
(6) else
(7)     for(i=1; i<=3; i++){
(8)         sum = sum-i;    }
(9) printf("sum = %d\n", sum);
```

1. ค่าของ sum ภายหลังการประมวลผลบรรทัดที่ (5) ในรอบแรก หากกำหนดให้บรรทัดที่ (2) รับค่า 6

.....

2. ค่าของ sum ที่ภายหลังการประมวลผลบรรทัดที่ (7) ในรอบแรก หากกำหนดให้บรรทัดที่ (2) รับค่า 5

.....

3. ค่าของ sum ที่ในบรรทัดที่ (9) หากกำหนดให้บรรทัดที่ (2) รับค่า 8

.....

4. ค่าของ sum ที่ในบรรทัดที่ (9) หากกำหนดให้บรรทัดที่ (2) รับค่า 5

.....

5. โปรแกรมจะทำงานบรรทัดที่ (7) เมื่อ num มีค่าในช่วงใด

.....

คะแนนในกิจกรรมที่ 3

รายการ	คะแนนเต็ม	คะแนนที่ได้
ตอนที่ 1	5	
ตอนที่ 2	5	

ใบความรู้ที่ 4

เรื่อง คำสั่งวนรอบซ้อนกัน

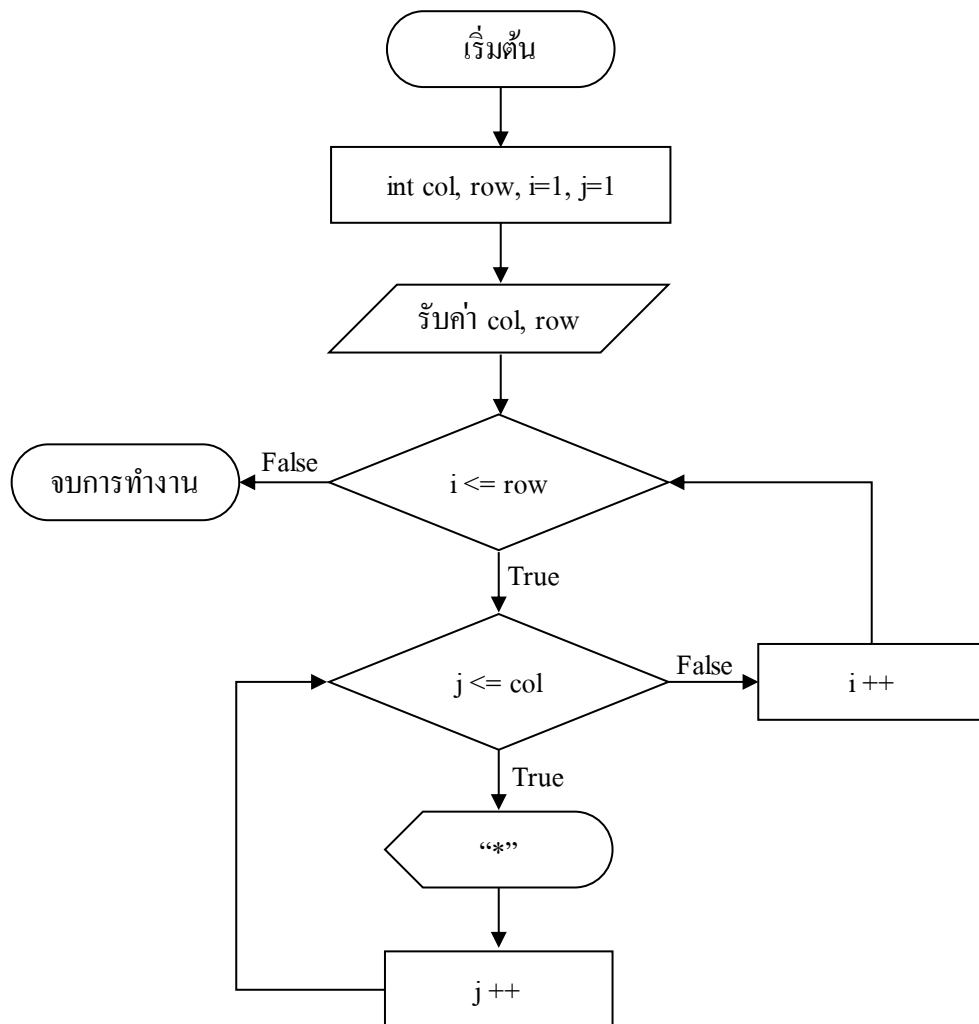
เวลาเรียน 1 ชั่วโมง

คำสั่งวนรอบซ้อนกัน

คำสั่งวนรอบซ้อนกัน คือ คำสั่งทำงานแบบวนรอบที่มีคำสั่งทำงานแบบวนรอบอีกอันหนึ่ง หรือมากกว่าเป็นคำสั่งย่อยอยู่ภายใน โดยคำสั่งวนรอบซ้อนกัน อาจเป็นคำสั่งทำซ้ำประเภทเดียวกัน หรือแตกต่างกันก็ได้ เช่น คำสั่ง while ซ้อนอยู่ในคำสั่ง while หรือ คำสั่ง for ซ้อนอยู่ในคำสั่ง while หรือคำสั่ง for ซ้อนอยู่ในคำสั่ง for เป็นต้น

ในการประมวลผลคำสั่งวนรอบซ้อนกัน ในแต่ละรอบของการประมวลผลคำสั่งทำซ้ำนั้น จะประมวลผลคำสั่งแบบวนรอบที่อยู่ชั้นในสุดจนกระทั่งสิ้นสุดเงื่อนไขการทำซ้ำของลูปนั้นๆ ก่อน แล้วจึงทำซ้ำคำสั่งวนซ้ำชั้นนอกในรอบถัดไปจนกระทั่งเงื่อนไขการทำซ้ำชั้นนอกจะสิ้นสุดลง จึงจะทำให้คำสั่งวนรอบซ้อนกันสิ้นสุดการทำงาน

ตัวอย่างที่ 1 โปรแกรมแสดงอักขระ (*) ออกมาทางจอภาพ n แถว แถวละ n ตัว
ด้วยคำสั่ง for loop ซ้อนกัน



ตัวอย่างที่ 1 โปรแกรม

```

(1) int i, j, col, row;      // ประกาศตัวแปร ชนิดจำนวนเต็ม ชื่อ i, j, col และ row
(2) printf("Enter columns : "); // พิมพ์ข้อความ Enter number : ทางจอภาพ
(3) scanf("%d", &col);      // รับค่าเป็นจำนวนเต็ม เก็บไว้ที่ตัวแปร col
(4) printf("Enter rows : "); // พิมพ์ข้อความ Enter number : ทางจอภาพ
(5) scanf("%d", &row);      // รับค่าเป็นจำนวนเต็ม เก็บไว้ที่ตัวแปร row
(6) for (i=1; i<=row; i++)  /* ค่าเริ่มต้น คือ ตัวแปร i มีค่าเท่ากับ 1
                             เงื่อนไข คือ ตัวแปร i มีค่าน้อยกว่าหรือเท่ากับ row
                             การเพิ่มค่าตัวแปร คือ ตัวแปร i เพิ่มค่าครั้งละ 1 */
(7) {
(8)     for(j=1; j<=col; j++){ /* ค่าเริ่มต้น คือ ตัวแปร j มีค่าเท่ากับ 1
                              เงื่อนไข คือ ตัวแปร j มีค่าน้อยกว่าหรือเท่ากับ col
                              การเพิ่มค่าตัวแปร คือ ตัวแปร j เพิ่มค่าครั้งละ 1 */
                                printf("*"); // พิมพ์ * ทางจอภาพ
                                }
(9)     printf("\n");        // ขึ้นบรรทัดใหม่
(10) }

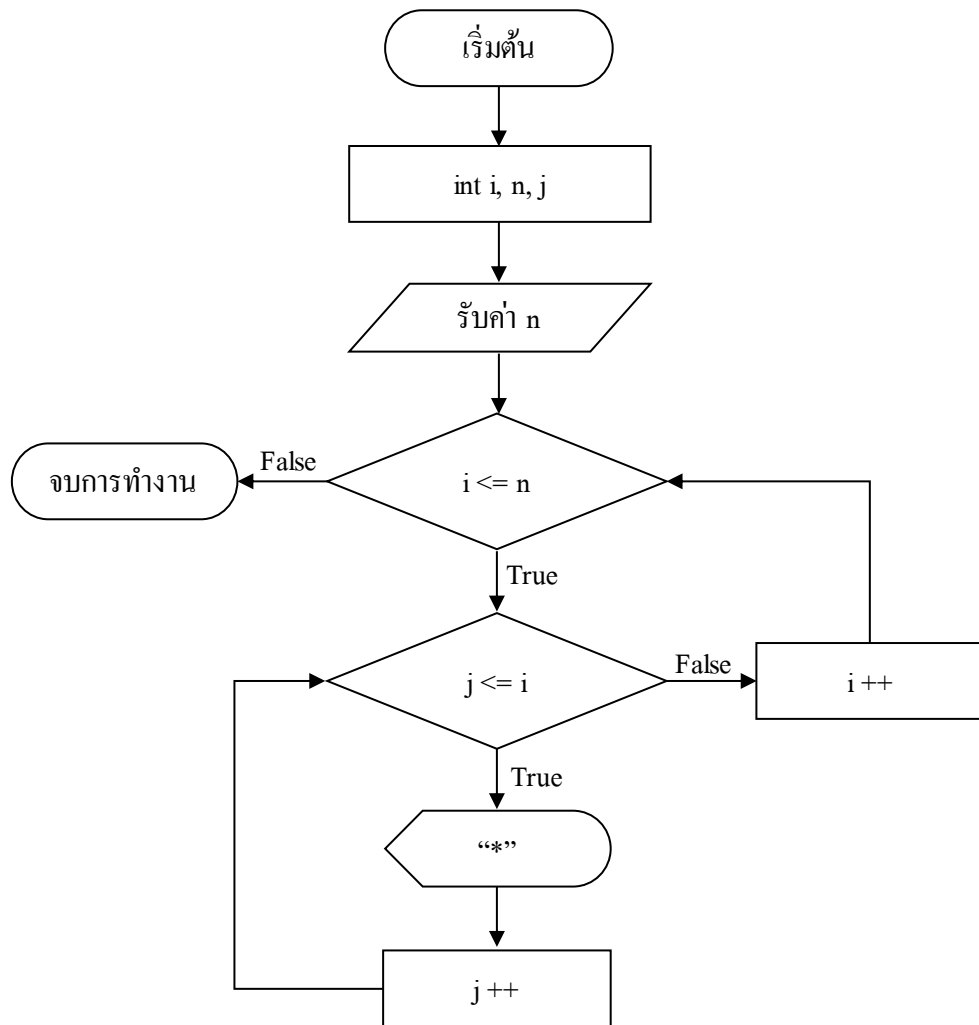
```

ผลลัพธ์ที่ได้จากการประมวลผล

Enter columns : 3

Enter rows : 4

ตัวอย่างที่ 2 โปรแกรมแสดงตัวอักษร (*) ออกมาทางจอภาพ n แถว โดยแต่ละแถว
จะเพิ่มจำนวนครั้งละ 1 ด้วยคำสั่ง for loop ซ้อนกัน



ตัวอย่างที่ 2 โปรแกรม

```

(1) int i, n, j;           // ประกาศตัวแปร ชนิดจำนวนเต็ม ชื่อ i , n และ j
(2) printf("Enter number : "); // พิมพ์ข้อความ Enter number : ทางจอภาพ
(3) scanf("%d", &n);       // รับค่าเป็นจำนวนเต็ม เก็บไว้ในที่ตัวแปร n
(4) for (i=1; i<=n; i++)   /* ค่าเริ่มต้น คือ ตัวแปร i มีค่าเท่ากับ 1
                             เงื่อนไข คือ ตัวแปร i มีค่าน้อยกว่าหรือเท่ากับ n
                             การเพิ่มค่าตัวแปร คือ ตัวแปร n เพิ่มค่าครั้งละ 1 */

(5) {
(6)     for(j=1; j<=i; j++) /* ค่าเริ่มต้น คือ ตัวแปร j มีค่าเท่ากับ 1
                             เงื่อนไข คือ ตัวแปร j มีค่าน้อยกว่าหรือเท่ากับ i
                             การเพิ่มค่าตัวแปร คือ ตัวแปร j เพิ่มค่าครั้งละ 1 */

(7)     {
(8)         printf("*");   // พิมพ์ * ทางจอภาพ
(9)     }
(10)    printf("\n");      // ขึ้นบรรทัดใหม่
(11) }

```

ผลลัพธ์ที่ได้จากการประมวลผล

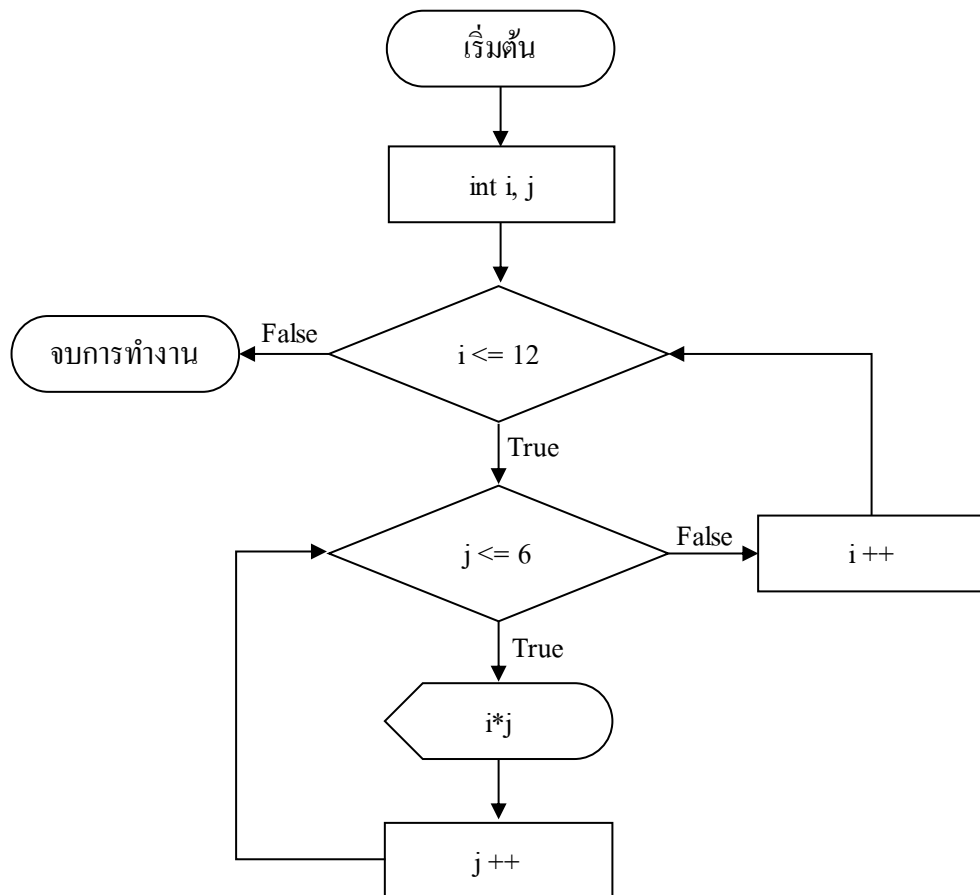
Enter number : 5

```

*
**
***
****
*****

```

ตัวอย่างที่ 3 โปรแกรมพิมพ์ผลการคูณแม่ 1 ถึง 6 ด้วยคำสั่ง for loop ซ้อนกัน



ตัวอย่างที่ 3 โปรแกรม

```
(1) int i, j; // ประกาศตัวแปร ชนิดจำนวนเต็ม ชื่อ i และ j
(2) for(i=1; i<=12; i++) /* คำเริ่มต้น คือ ตัวแปร i มีค่าเท่ากับ 1
                          เงื่อนไข คือ ตัวแปร i มีค่าน้อยกว่าหรือเท่ากับ 12
                          การเพิ่มค่าตัวแปร คือ ตัวแปร i เพิ่มค่าครั้งละ 1 */
(3) {
(4)     for(j=1; j<=6; j++) /* คำเริ่มต้น คือ ตัวแปร j มีค่าเท่ากับ 1
                           เงื่อนไข คือ ตัวแปร j มีค่าน้อยกว่าหรือเท่ากับ 6
                           การเพิ่มค่าตัวแปร คือ ตัวแปร j เพิ่มค่าครั้งละ 1 */
(5)     {
(6)         printf("%d\t", i*j); // พิมพ์ค่า i คูณ j ในรูปแบบเลขจำนวนเต็ม
(7)     }
(8)     printf("\n"); // ขึ้นบรรทัดใหม่
(9) }
```

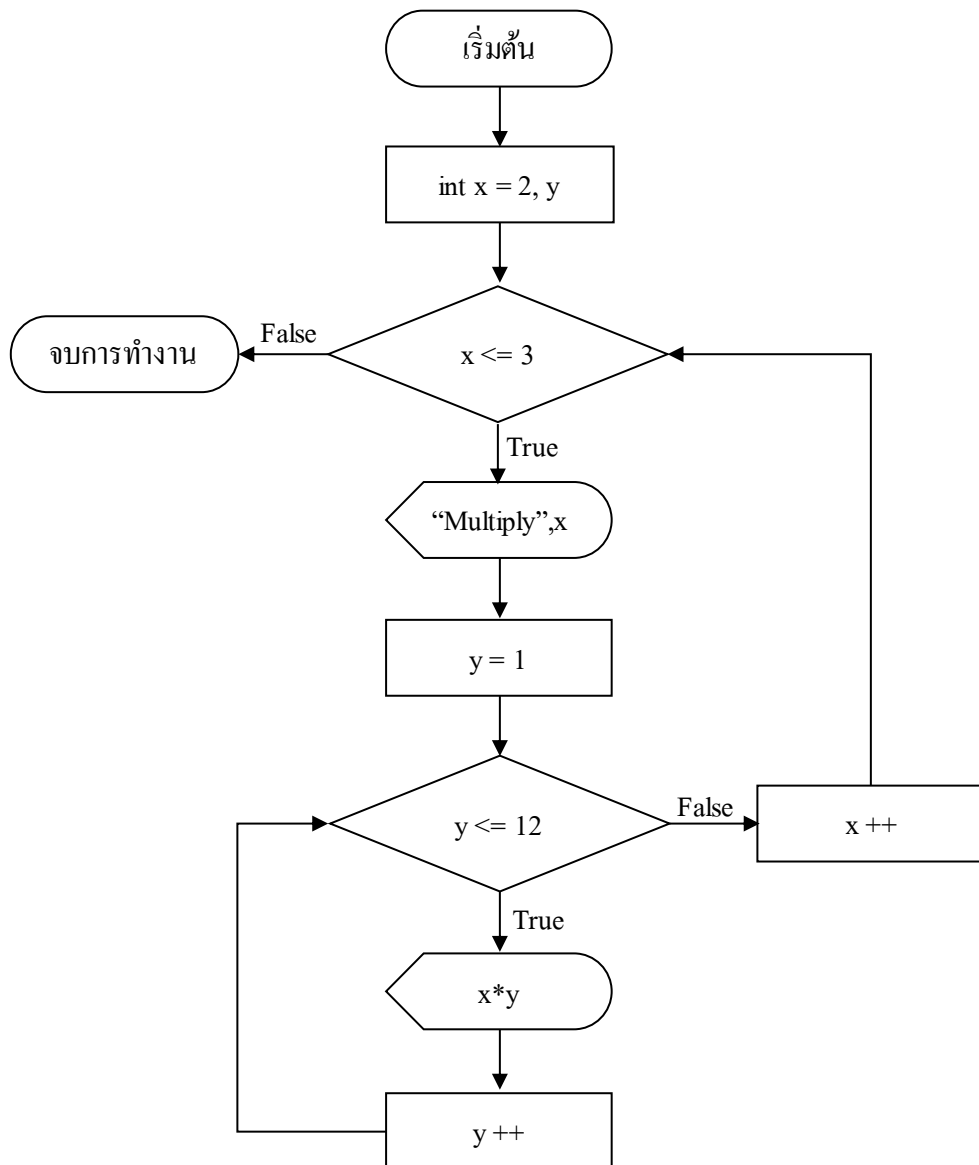


ผลลัพธ์ที่ได้จากการประมวลผล

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36
7	14	21	28	35	42
8	16	24	32	40	48
9	18	27	36	45	54
10	20	30	40	50	60
11	22	33	44	55	66
12	24	36	48	60	72



ตัวอย่างที่ 4 โปรแกรมพิมพ์สูตรคูณแม่ 2 ถึง 3 ด้วยคำสั่ง for loop และ while loop



ตัวอย่างที่ 4 โปรแกรม

```

(1) int x,y;           // ประกาศตัวแปร ชนิดจำนวนเต็ม ชื่อ x และ y
(2) for(x=2; x<=3; x++) /* ค่าเริ่มต้น คือ ตัวแปร x มีค่าเท่ากับ 2
                        เงื่อนไข คือ ตัวแปร x มีค่าน้อยกว่าหรือเท่ากับ 3
                        การเพิ่มค่าตัวแปร คือ ตัวแปร x เพิ่มค่าครั้งละ 1 */
(3) {
(4)     printf("\nMultiply %d",x); /* พิมพ์ค่า x ในรูปแบบเลขจำนวนเต็ม
                                    ทางจอภาพ */
(5)     y=1;           // กำหนดค่าตัวแปร y ให้เท่ากับ 1
(6)     while(y<=12)   /* ตรวจสอบเงื่อนไขตัวแปร y มีค่าน้อยกว่า
                        หรือเท่ากับ 12 */
(7)     {
(8)         printf("\n %d * %d = %d",x, y, x*y); /* พิมพ์ค่า x ค่า y และ
                                                    x คูณ y ในรูปแบบเลข
                                                    จำนวนเต็มทางจอภาพ */
(9)         y++;
(10)    }
(11)    printf("\n+++++++"); // พิมพ์ ++++++++ ทางจอภาพ
(12) }

```

ผลลัพธ์ที่ได้จากการประมวลผล

Multiply 2

$$2 * 1 = 2$$

$$2 * 2 = 4$$

$$2 * 3 = 6$$

$$2 * 4 = 8$$

$$2 * 5 = 10$$

$$2 * 6 = 12$$

$$2 * 7 = 14$$

$$2 * 8 = 16$$

$$2 * 9 = 18$$

$$2 * 10 = 20$$

$$2 * 11 = 22$$

$$2 * 12 = 24$$

+++++

Multiply 3

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

$$3 * 4 = 12$$

$$3 * 5 = 15$$

$$3 * 6 = 18$$

$$3 * 7 = 21$$

$$3 * 8 = 24$$

$$3 * 9 = 27$$

$$3 * 10 = 30$$

$$3 * 11 = 33$$

$$3 * 12 = 36$$

+++++



สรุปเนื้อหา คำสั่งวนรอบซ้อนกัน

คำสั่งทำซ้ำมีคำสั่งย่อยเป็นการทำซ้ำซ้อนอยู่ภายใน อาจเป็นคำสั่งทำซ้ำประเภทเดียวกัน หรือแตกต่างกันก็ได้ โดยการประมวลผลคำสั่งวนรอบซ้อนกันจะทำงานเป็นชั้น ๆ จากชั้นในสุดสู่ชั้นนอกตามลำดับ

ศึกษาจนเข้าใจแล้ว

ไปทำกิจกรรมกันเลยคะ



ใบกิจกรรมที่ 4

เรื่อง คำสั่งวนรอบซ้อนกัน

คำชี้แจง แบบทดสอบอัตนัยการเขียนโปรแกรม จำนวน 2 ข้อ 10 คะแนน เวลา 10 นาที

ข้อที่ 1

คำสั่ง ให้เขียนโปรแกรมด้วยคำสั่ง For loop รับค่าจำนวนเต็มจากแป้นพิมพ์ จากนั้นให้แสดงอักขระ * จำนวนบรรทัดเท่ากับที่กรอก โดยแต่ละบรรทัดให้แสดงเพียงหนึ่งอักขระ ดังตัวอย่าง เมื่อกรอกตัวเลข 5

```
*
 *
  *
   *
    *
```

ตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....



ข้อที่ 2

คำสั่ง ให้นักเรียนเขียนโปรแกรมด้วยคำสั่ง While loop และ For loop แสดงผลลัพธ์ต่อไปนี้

```
123456
12345
1234
123
12
1
```

ตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คะแนนในกิจกรรมที่ 4		
รายการ	คะแนนเต็ม	คะแนนที่ได้
ตอนที่ 1	5	
ตอนที่ 2	5	





แบบทดสอบหลังเรียน เล่ม 8 เรื่อง คำสั่งวนรอบการทำงาน

แบบทดสอบฉบับนี้มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

ตอนที่ 2 เป็นแบบอัตนัย เต็มคำจำนวน 1 ข้อ 10 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

1. ประโยคคำสั่งวนรอบการทำงาน มีรูปแบบการทำงานดังข้อใด
 - ก. คำสั่งวนรอบการทำงานสามารถประยุกต์ใช้เพื่อแก้ปัญหาได้ทุกอย่าง
 - ข. ปัญหาบางอย่าง ไม่สามารถแก้ไขได้ด้วยคำสั่งวนรอบ
 - ค. คำสั่งวนรอบการทำงาน แก้ปัญหาได้เฉพาะการทำงานที่มีจำนวนรอบที่แน่นอนเท่านั้น
 - ง. คำสั่งวนรอบการทำงาน ใช้งานกับอุปกรณ์เฉพาะอย่างเท่านั้น
2. คำสั่ง while ในข้อใดมีรูปแบบถูกต้อง
 - ก. while (x <=10)
 - ข. while {x >=10}
 - ค. while [x <=10]
 - ง. while "x >=10"
3. ข้อใดคือคำสั่งวนรอบการทำงานที่ตรวจสอบเงื่อนไขเป็นจริงก่อนจึงจะทำงานตามคำสั่ง

ก. for	ข. while
ค. break	ง. do while
4. ข้อใดคือคำสั่งที่สามารถกำหนดจำนวนครั้งการวนรอบการทำงานได้

ก. if	ข. for
ค. while	ง. do while



5. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int i = 1;
do
{
    printf("A");
    i++;
}
while (i<=3);
```

ก. AAAAA

ข. AAAA

ค. AAA

ง. AA

6. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int i=1,j;
while(i<=2)
{
    for(j=3; j>0; j--)
    {
        printf("%d",j);
    }
    i++;
    printf("\n");
}
```

ก. 123

ข. 321

123

321

123

321

ค. 123

ง. 321

123

321

7. จากโปรแกรม จะได้ผลลัพธ์ในข้อใด

```
int A, B;
for(A=3; A>0; A--)
{
    for(B=1; B<=A; B++)
    {
        printf("%d",B);
    }
}
```

- | | |
|--------|--------|
| ก. 123 | ข. 1 |
| 12 | 12 |
| 1 | 123 |
| ค. 1 | ง. 333 |
| 22 | 22 |
| 333 | 1 |

8. ให้นักเรียนเขียนโปรแกรมเพื่อแสดงเลขคู่ ระหว่าง 0 – 100 โดยใช้คำสั่ง while loop

```
int i=0;
while (① ..... )
{
    print("%d ",i);
    (② ..... );
}
```

- ก. ① i=0 และ ② i++
 ข. ① i=0 และ ② i=i+1
 ค. ① i<=100 และ ② i++
 ง. ① i<=100 และ ② i+=2



9. ให้นักเรียนเขียนโปรแกรมเพื่อหาผลรวมของเลขคู่ ระหว่าง 1 – 20 โดยใช้คำสั่ง do while loop

```
int num = 2, sum = 0;

do
{
    sum += num;
    (① .....);
} while (② .....);

print("sum of even num: %d", sum);
```

- ก. ① num++ และ ② num<=20
- ข. ① num++ และ ② num>=20
- ค. ① num+=2 และ ② num<=20
- ง. ① num+=2 และ ② num>=20

10. ให้นักเรียนเขียนโปรแกรมเพื่อพิมพ์เลขระหว่าง 1 – 100 โดยใช้คำสั่ง for loop

```
int i;

for (.....)

    print("%d",i);
```

- ก. i=0; i=1-100; i++;
- ข. i=0; i=1-100; i++
- ค. i=1; i<=100; i++
- ง. i=1; i<100; i++

ตอนที่ 2 ข้อสอบแบบอัตนัย (10 คะแนน)

คำสั่ง ให้นักเรียนเขียนโปรแกรมให้มีผลลัพธ์ดังรูป

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

```

ตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



อย่าลืม!!!
เปลี่ยนกันตรวจนะคะ



กระดาษคำตอบแบบทดสอบหลังเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

เอกสารประกอบการเรียน เรื่องการคำนวณและคำสั้งภาษาเบื้องต้น
รายวิชาการเขียนโปรแกรมภาษาซี 1 รหัสวิชา ง30249 (รายวิชาเพิ่มเติม)
กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5

ชื่อ – สกุล.....ชั้น.....เลขที่.....

ข้อ	ก	ข	ค	ง
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

คะแนนแบบทดสอบก่อนเรียน

คะแนนตอนที่ 1	
คะแนนตอนที่ 2	
คะแนนรวมที่ได้	

ผลการประเมิน

- ☐ ดีมาก ★★★★★
- ☐ ดี ★★★
- ☐ พอใช้ ★★
- ☐ ปรับปรุง ★

เกณฑ์การประเมิน

คะแนน 17 – 20 อยู่ในเกณฑ์ ดีมาก

คะแนน 14 – 16 อยู่ในเกณฑ์ ดี

คะแนน 11 – 13 อยู่ในเกณฑ์ พอใช้

คะแนน 0 – 10 อยู่ในเกณฑ์ ปรับปรุง

เสร็จแล้ว!!!

รีบทำกิจกรรมต่อไปกัน



แบบบันทึกการประเมิน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

เอกสารประกอบการเรียน เรื่องการคำนวณและคำสั่งภาษาเบื้องต้น
รายวิชาการเขียนโปรแกรมภาษาซี 1 รหัสวิชา ง30249 (รายวิชาเพิ่มเติม)
กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 5

ชื่อ - สกุล.....ชั้น.....เลขที่.....

กิจกรรมการเรียนรู้	คะแนน		ผลการประเมิน	
	เต็ม	ได้	ผ่าน	ไม่ผ่าน
กิจกรรมที่ 1 คำสั่ง While loop	10			
กิจกรรมที่ 2 คำสั่ง Do-while loop	10			
กิจกรรมที่ 3 คำสั่ง For loop	10			
กิจกรรมที่ 4 คำสั่งวนรอบซ้อนกัน	10			
รวมคะแนนทั้งหมด	40			

เกณฑ์การผ่านกิจกรรม ได้คะแนนตั้งแต่ร้อยละ 70 ขึ้นไป (28 คะแนนขึ้นไป)

บรรณานุกรม

- กิตินันท์ พลสวัสดิ์. (2554). รวมโจทย์และแบบฝึกหัดภาษา C+ Java. นนทบุรี: ไอดีซี พรีเมียร์.
- ไกรสร ตั้งโอภากุล และ กิตินันท์ พลสวัสดิ์. (2554). คู่มือเรียนเขียนโปรแกรมภาษา C. นนทบุรี: ไอดีซี พรีเมียร์.
- ธีรวัฒน์ ประกอบผล. (2550). การเขียนโปรแกรมบนระบบปฏิบัติการเท็กซ์โหมด. กรุงเทพฯ: ชัคเชส มีเดีย.
- พัฒนพงษ์ อมรวงศ์. (2554). การเขียนโปรแกรมภาษาซี. ปทุมธานี: มีเดีย อินเทลลิเจนซ์ เทคโนโลยี.
- ศุภชัย สมพานิช. (2557). คู่มือเรียนและเขียนโปรแกรมด้วยภาษา C. กรุงเทพฯ: สวัสดิ์ ไอที.
- สมโภชน์ ชื่นเอี่ยม และคณะ. (ม.ป.ป.). การเขียนโปรแกรมเบื้องต้น (ภาษาซี). กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- โอภาส เอี่ยมสิริวงศ์. (2552). การเขียนโปรแกรมด้วยภาษา C. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.



ภาคผนวก



เฉลยแบบทดสอบก่อนเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

คำชี้แจง แบบทดสอบฉบับนี้มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

ตอนที่ 2 เป็นแบบอัตนัย เต็มคำจำนวน 1 ข้อ 10 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

ข้อ	คำตอบ
1	ง
2	ก
3	ง
4	ก
5	ค
6	ค
7	ง
8	ก
9	ค
10	ข

ตอนที่ 2 ข้อสอบแบบอัตนัย (10 คะแนน)

คำสั่ง ให้นักเรียนเขียนโปรแกรมหาผลบวกของเลขจำนวนเต็มที่มีค่าอยู่ในช่วงตั้งแต่ x ถึง y โดยที่ค่า x และ y ถูกผู้ใช้กำหนดขึ้น โดยให้ค่า $y > x$ แล้วพิมพ์ผลลัพธ์ทางจอภาพ

ตอบ ตอบ ตัวอย่างของเฉลยโปรแกรม เช่น

```
int i, x, y, sum = 0;
printf("Enter x : ");
scanf("%d", &x);
printf("Enter y : ");
scanf("%d", &y);
i = x;
while(i <= y)
{
    sum = sum + i;
    i++;
}
printf("sum between %d and %d is %d", x, y, sum);
```

หรือตอบ

```
int x, y, sum = 0;
printf("Enter x : ");
scanf("%d", &x);
printf("Enter y : ");
scanf("%d", &y);
while(x <= y)
{
    sum = sum + x;
    x++;
}
printf("result is %d", sum);
```

เฉลยใบกิจกรรมที่ 1

เรื่อง คำสั่ง While loop

คำชี้แจง ใบกิจกรรมที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

1. ข้อใด คือ ลักษณะการทำงานของคำสั่ง while loop
 - ก. ทำงานและตรวจสอบเงื่อนไขพร้อมกัน
 - ข. ทำงานตามคำสั่งก่อนจึงตรวจสอบเงื่อนไข
 - ค. ตรวจสอบเงื่อนไขก่อนแล้วจึงทำงานต่อไป
 - ง. ตรวจสอบเงื่อนไขก่อนก็ได้ หรือทำงานก่อนก็ได้
2. ข้อใดกล่าวถึงคำสั่ง while loop ได้ถูกต้อง
 - ก. การทำงานซ้ำ ๆ ที่มีจำนวนรอบที่แน่นอน
 - ข. ลูปจะทำงานวนไปจนกว่าเงื่อนไขจะเป็นเท็จ
 - ค. หากเงื่อนไขของลูปเป็นเท็จ โปรแกรมจะทำงานตามคำสั่งที่อยู่ภายในลูป
 - ง. หากเงื่อนไขของลูปจะเป็นจริงในครั้งแรก จะไม่มีการทำงานตามคำสั่งใด ๆ
3. หาก while loop มีคำสั่งวนซ้ำ หลายบรรทัด ต้องใช้เครื่องหมายใด

ก. []	ข. < >
ค. ()	ง. { }
4. จากโปรแกรมต่อไปนี้ จะเกิดการซ้ำของ while loop จำนวนกี่ครั้ง


```
int n = 3;
while (n > 3)
{
    printf("%d", n);
}
```

ก. 3	ข. 4
ค. 5	ง. ไม่เกิดการซ้ำ



5. จากโปรแกรมต่อไปนี้ จะเกิดการซ้ำของ while loop จำนวนกี่ครั้ง

```
int n = 1;
while (n <= 2)
{
    printf("%d    %d\n", n, n*2);
}
```

ก. 2

ข. 4

ค. ไม่มีจุดสิ้นสุด

ง. ไม่เกิดการทำงาน

จากโปรแกรมต่อไปนี้จึงตอบคำถามข้อ 6 – 7

1	int i =3;
2	while(i<=5)
3	{
4	printf("Hello...\n");
5	i++;
6	}

6. จากโปรแกรมจะได้ Hello.... จำนวนกี่บรรทัด

ก. 5

ข. 3

ค. 2

ง. 1

7. เมื่อสิ้นสุดการทำงานของ while loop ตัวแปร i มีค่าเท่าใด

ก. 6

ข. 5

ค. 3

ง. 1



10. จากโปรแกรมจะได้ผลลัพธ์ตรงกับข้อใด

```
int a = 1, b = 2, c = 3;
while(a <= 10)
{
    if((a<=5) && (c>=2))
    {
        a = b + 8;
    }
    b = a * 2;
    c = b - 3;
    a++;
}
printf("A = %d, B = %d, C = %d",a,b,c);
```

ก. A = 1, B = 2, C = 3

ข. A = 10, B = 20, C = 17

ค. A = 11, B = 20, C = 17

ง. A = 11, B = 22, C = 19

เฉลยใบกิจกรรมที่ 2

เรื่อง คำสั่ง Do-while loop

คำชี้แจง ใบกิจกรรมที่ 2 เป็นแบบทดสอบอัตรัยการเขียนโปรแกรมโดยใช้คำสั่ง Do-while loop จำนวน 2 ข้อ 10 คะแนน เวลา 10 นาที

ข้อที่ 1

คำสั่ง ให้นักเรียนเขียนผลลัพธ์ที่ได้จากส่วนของโปรแกรมนี้ เมื่อผู้ใช้กรอกข้อมูล 7000

```
int i=0;
float pop;
scanf("%f", &pop);
do{
    pop *= 1.1;
    i++;
    printf("in %d year, population = %.2f\n", i, pop);
}while(pop<10000);
```

ตอบ

.....in 1 year, population = 7700.00.....

.....in 2 year, population = 8470.00.....

.....in 3 year, population = 9317.00.....

.....in 4 year, population = 10248.70.....

.....

ข้อที่ 2

คำสั่ง ให้นักเรียนเขียนโปรแกรมเพื่อรับค่าตัวเลขจำนวนเต็มไปจนกว่าผู้ใช้จะตอบ 'N'
แล้วแสดงผลรวมของเลขทั้งหมดที่กรอกนั้น

ตอบตัวอย่างโปรแกรม.....

```
char ch;  
int n, sum=0;  
do  
{  
    printf("\nEnter integer : ");  
    scanf("%d", &n);  
    sum = sum+n;  
    printf("Do you want to continue (N to stop) : ");  
    scanf(" %c", &ch);  
}  
while (ch != 'N');  
printf("\nSum : %d", sum);
```

เฉลยใบกิจกรรมที่ 3

เรื่อง คำสั่ง For loop

คำชี้แจง ใบกิจกรรมที่ 1 มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 5 ข้อ 5 คะแนน เวลา 5 นาที

ตอนที่ 2 เป็นแบบเติมคำตอบแบบสั้น จำนวน 5 ข้อ 5 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 5 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

ให้นักเรียนพิจารณาโปรแกรมแสดงสูตรคูณ โดย รับค่า n แทนแม่สูตรคูณต่อไปนี้ เพื่อตอบในข้อที่ 1 - 3

```
(1) int i,n,sum=0;
(2) printf("Enter mulitplication number : ");
(3) .....1.....;
(4) .....2.....
(5) {
(6)     printf("%d * %d = %d\n",.....3.....);
(7) }
```

1. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 1

- ก. scanf("%d", &n)
- ข. scanf("%d", &sum)
- ค. printf("%d", n)
- ง. printf("%d", sum)

2. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 2

ก. for (i<=12; i++)

ข. for (i<=12; i=1)

ค. for (i=1; i<=12; i++)

ง. for (i++; i<=12; i=1)

3. จากโจทย์ดังกล่าวให้เติมคำสั่งภาษาซีที่เหมาะสมในตำแหน่งเลข 3

ก. i, n, i*i

ข. i, n, n*i

ค. n, i, n*n

ง. n, i, n²

4. จากผลลัพธ์ที่ได้จากการประมวลผล ควรเลือกคำสั่งในข้อใด

1 + 5 = 6

2 + 4 = 6

3 + 3 = 6

4 + 2 = 6

ก. for(i=1, j=5; i<5; i++, j--)

printf("%d + %d = %d\n", i, j, i+j);

ข. for(i=1, j=5; i<=5; i++, j++)

printf("%d + %d = %d\n", i, j, i+j);

ค. for(i=0, j=5; i<5; i++, j--)

printf("%d + %d = %d\n", i, j, i+j);

ง. for(i=0, j=5; i<=5; i++, j++)

printf("%d + %d = %d\n", i, j, i+j);



5. จากคำสั่งต่อไปนี้ได้ผลลัพธ์ตรงกับข้อใด

```
int num, sum=0;
for(num=1; num<=5; num=num+2){
    sum = sum +num;
}
printf("sum = %d\n",sum);
```

ก. 3

ข. 7

ค. 9

ง. 15

ตอนที่ 2 เป็นแบบเติมคำตอบแบบสั้น จำนวน 5 ข้อ

คำสั่ง จากโปรแกรมด้านล่างนี้ ให้พิจารณาค่าต่าง ๆ ตามที่โจทย์กำหนดในแต่ละข้อ

```
(1) int i, num, sum=0;
(2) scanf("%d", &num);
(3) if(num>5&&num<10)
(4)     for(i=1; i<=5; i=i+2){
(5)         sum = sum +i;    }
(6) else
(7)     for(i=1; i<=3; i++){
(8)         sum = sum-i;    }
(9) printf("sum = %d\n", sum);
```

1. ค่าของ sum ภายหลังการประมวลผลบรรทัดที่ (5) ในรอบแรก หากกำหนดให้บรรทัดที่ (2) รับค่า 6

.....1.....

2. ค่าของ sum ที่ภายหลังการประมวลผลบรรทัดที่ (7) ในรอบแรก หากกำหนดให้บรรทัดที่ (2) รับค่า 5

.....-1.....

3. ค่าของ sum ที่ในบรรทัดที่ (9) หากกำหนดให้บรรทัดที่ (2) รับค่า 8

.....9.....

4. ค่าของ sum ที่ในบรรทัดที่ (9) หากกำหนดให้บรรทัดที่ (2) รับค่า 5

.....-6.....

5. โปรแกรมจะทำงานบรรทัดที่ (7) เมื่อ num มีค่าในช่วงใด

.....มี 2 ช่วง คือ ค่าน้อยกว่าหรือเท่ากับ 5 หรือ ค่ามากกว่าหรือเท่ากับ 10.....



เฉลยใบกิจกรรมที่ 4

เรื่อง คำสั่งวนรอบซ้อนกัน

คำชี้แจง แบบทดสอบอัตนัยการเขียนโปรแกรม จำนวน 2 ข้อ 10 คะแนน เวลา 10 นาที

ข้อที่ 1

คำสั่ง ให้เขียนโปรแกรมด้วยคำสั่ง For loop รับค่าจำนวนเต็มจากแป้นพิมพ์ จากนั้นให้แสดงอักขระ * จำนวนบรรทัดเท่ากับที่กรอก โดยแต่ละบรรทัดให้แสดงเพียงหนึ่งอักขระ ดังตัวอย่าง เมื่อกรอกตัวเลข 5

```
*
 *
  *
   *
    *
```

ตอบ

```
int i, j, num;
scanf("%d", &num);
for(i=1; i<=num; i++)
{
    for(j=1; j<=i-1; j++)
        printf(" ");
    printf("*\n");
}
```



ข้อที่ 2

คำสั่ง ให้นักเรียนเขียนโปรแกรมด้วยคำสั่ง While loop และ For loop แสดงผลลัพธ์ต่อไปนี้

```
123456
12345
1234
123
12
1
```

ตอบ

```
int A, B;
```

```
for(A=6; A>0; A--)
```

```
{
```

```
for(B=1; B<=A; B++)
```

```
{
```

```
printf("%d",B);
```

```
}
```

```
printf("\n");
```

```
}
```



เฉลยแบบทดสอบหลังเรียน
หน่วยการเรียนรู้ที่ 8 เรื่อง คำสั่งวนรอบการทำงาน

คำชี้แจง

แบบทดสอบฉบับนี้มี 2 ตอน

ตอนที่ 1 เป็นแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ 10 คะแนน เวลา 10 นาที

ตอนที่ 2 เป็นแบบอัตนัย เต็มคำจำนวน 1 ข้อ 10 คะแนน เวลา 5 นาที

ตอนที่ 1 ข้อสอบแบบเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ

คำสั่ง ให้นักเรียนเลือกคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว ทำเครื่องหมาย (X) ลงในกระดาษคำตอบ

ข้อ	คำตอบ
1	ข
2	ก
3	ข
4	ข
5	ค
6	ง
7	ก
8	ง
9	ค
10	ค



ตอนที่ 2 ข้อสอบแบบอัตนัย (10 คะแนน)

คำสั่ง ให้นักเรียนเขียนโปรแกรมให้มีผลลัพธ์ดังรูป

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

ตอบ

```
int i, j;
```

```
for(i=1; i<=5; i++)
```

```
{ for(j=1; j<=i; j++)
```

```
printf("* ");
```

```
printf("\n");
```

```
}
```

```
for(i=4; i>=1; i--)
```

```
{ for(j=i; j>=1; j--)
```

```
printf("* ");
```

```
printf("\n");
```

```
}
```

