

แบบฝึกทักษะ การเขียนโปรแกรมภาษาซี

สำหรับนักเรียนชั้นมัธยมศึกษาปีที่ 5



คำสั่งแบบวนรอบการทำงาน

โดย นางสาวประกายเพชร เอ้ยวัน
ตำแหน่งครู วิทยฐานะชำนาญการ

กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี
โรงเรียนวังชันวิทยา อำเภอวังชัน จังหวัดแพร่
สำนักงานเขตพื้นที่การศึกษามัธยมศึกษา เขต 37
สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน กระทรวงศึกษาธิการ





แบบฝึกทักษะการเขียนโปรแกรมภาษาซีนี้ จัดทำขึ้นเพื่อใช้ประกอบการเรียนรู้และ ฝึกปฏิบัติการ เขียนโปรแกรมภาษาซี รายวิชา ง32242 เทคโนโลยีสารสนเทศ 4 ชั้นมัธยมศึกษาปีที่ 5 กลุ่มสาระ การเรียนรู้การงานอาชีพและเทคโนโลยี ตามหลักสูตรแกนกลางการศึกษาขั้นพื้นฐาน พุทธศักราช 2551 ทั้งนี้ นักเรียนสามารถเรียนรู้ได้ด้วยตนเองทั้งในห้องเรียนและนอกห้องเรียน เป็นการกระตุ้นให้นักเรียน เกิดแนวคิดและเข้าใจหลักการเขียนโปรแกรมภาษาซี ตลอดจนมีทัศนคติ ที่ดีต่อการเรียนวิชาการเขียน โปรแกรม

แบบฝึกทักษะการเขียนโปรแกรมภาษาซี ในแต่ละเล่มจะประกอบด้วย แบบทดสอบ ก่อนเรียน- หลังเรียน เนื้อหาความรู้ และกิจกรรมให้นักเรียนได้ลงมือปฏิบัติ เพื่อให้เกิดความเข้าใจและเกิดทักษะ ในการเขียนโปรแกรมภาษาซี มีทั้งหมด 6 เล่ม ดังนี้

- เล่มที่ 1 เรื่อง ความรู้เบื้องต้นเกี่ยวกับภาษาซี
- เล่มที่ 2 เรื่อง ตัวดำเนินการและนิพจน์
- เล่มที่ 3 เรื่อง การแสดงผลข้อมูลและรับข้อมูล
- เล่มที่ 4 เรื่อง คำสั่งแบบเลือกตามเงื่อนไข
- เล่มที่ 5 เรื่อง คำสั่งแบบวนรอบการทำงาน
- เล่มที่ 6 เรื่อง การประยุกต์ใช้ในชีวิตประจำวัน

ขอขอบพระคุณ ผู้อำนวยการโรงเรียนวังชันวิทยา คณะครู นักเรียน โรงเรียนวังชันวิทยา ผู้เชี่ยวชาญ ผู้ทรงคุณวุฒิ ที่กรุณาให้คำแนะนำ ช่วยเหลือ ให้ข้อเสนอแนะในการพัฒนาแบบฝึกทักษะ ทั้ง 6 เล่มนี้ จนสำเร็จลุล่วงไปได้ด้วยดี

ผู้จัดทำหวังเป็นอย่างยิ่งว่าแบบฝึกทักษะนี้ จะเป็นประโยชน์ต่อนักเรียน ครูผู้สอน และผู้ที่สนใจ ฝึกทักษะการเขียนโปรแกรมภาษาซี

ประกายเพชร เอ้ยวัน





	หน้า
คำนำ	ก
สารบัญ	ข
คำชี้แจงในการใช้แบบฝึกทักษะสำหรับครู	ง
คำชี้แจงในการใช้แบบฝึกทักษะสำหรับนักเรียน	จ
แบบฝึกทักษะ เรื่อง คำสั่งแบบวนรอบการทำงาน	1
สาระสำคัญ	1
ผลการเรียนรู้	1
จุดประสงค์การเรียนรู้	2
สาระการเรียนรู้	2
แบบทดสอบก่อนเรียน	3
เกณฑ์การให้คะแนนแบบทดสอบก่อนเรียน	7
คำสั่งแบบวนรอบการทำงาน	8
➤ การใช้คำสั่ง for	8
➤ การใช้คำสั่ง while	14
➤ การใช้คำสั่ง do while	18
สรุปท้ายบท	23
แบบฝึกทักษะที่ 5.1	24
เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.1	27
แบบฝึกทักษะที่ 5.2	28
เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.2	30
แบบฝึกทักษะที่ 5.3	31
เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.3	33
แบบฝึกทักษะที่ 5.4	34
เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.4	37





	หน้า
แบบทดสอบหลังเรียน	38
เกณฑ์การให้คะแนนแบบทดสอบหลังเรียน	42
เฉลยแบบทดสอบก่อนเรียน	43
เฉลยแบบทดสอบหลังเรียน	43
เฉลยแบบฝึกทักษะที่ 5.1	47
เฉลยแบบฝึกทักษะที่ 5.2	44
เฉลยแบบฝึกทักษะที่ 5.3	49
เฉลยแบบฝึกทักษะที่ 5.4	51
แบบบันทึกคะแนน	57
บรรณานุกรม	58





คำชี้แจงในการใช้แบบฝึกทักษะสำหรับครู

แบบฝึกทักษะการเขียนโปรแกรมภาษาซี เล่มที่ 5 คำสั่งแบบวนรอบการทำงาน เป็นแบบฝึกทักษะที่ใช้ในการเรียนการสอน รายวิชา ง32242 เทคโนโลยีสารสนเทศ 4 ชั้นมัธยมศึกษาปีที่ 5 ที่เน้นให้ผู้เรียนได้ลงมือปฏิบัติด้วยตนเอง และครูควรปฏิบัติ ดังนี้

1. ครูควรศึกษาทำความเข้าใจเกี่ยวกับเนื้อหา แล้วปฏิบัติตามกิจกรรมในแผนการจัดการเรียนรู้
2. ครูแจกแบบฝึกทักษะการเขียนโปรแกรมภาษาซี เล่มที่ 5 คำสั่งแบบวนรอบการทำงาน ให้แก่นักเรียน
3. ชี้แจงการใช้แบบฝึกทักษะการเขียนโปรแกรมภาษาซี ให้นักเรียนทราบก่อนลงมือปฏิบัติ
4. ครูให้นักเรียนทำแบบทดสอบก่อนเรียน เพื่อวัดความรู้พื้นฐานของนักเรียน
5. ครูให้นักเรียนศึกษาแบบฝึกทักษะการเขียนโปรแกรมภาษาซี เล่มที่ 5 คำสั่งแบบวนรอบการทำงาน ปฏิบัติตามใบความรู้ และแบบฝึกทักษะตามลำดับ
6. ครูให้นักเรียนทำแบบทดสอบหลังเรียน เพื่อประเมินผลว่านักเรียนมีความรู้ ความเข้าใจมากขึ้นเพียงใด
7. ครูสังเกตพฤติกรรมคุณลักษณะที่พึงประสงค์ และบันทึกผล
8. ครูตรวจและบันทึกผลการทำงานของนักเรียนลงในแบบเก็บคะแนน
9. ครูแจ้งคะแนนให้นักเรียนทราบ ให้คำแนะนำเพิ่มเติม และรับฟังข้อเสนอแนะจากนักเรียน



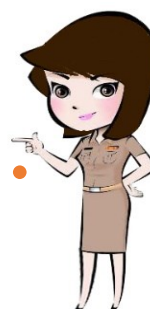


คำชี้แจงในการใช้แบบฝึกทักษะสำหรับนักเรียน

แบบฝึกทักษะการเขียนโปรแกรมภาษาซี เล่มที่ 5 คำสั่งแบบวนรอบการทำงาน จัดทำขึ้น โดยมีจุดประสงค์ เพื่อให้ให้นักเรียนได้เรียนรู้ ด้วยการใช้ทักษะกระบวนการการเขียนโปรแกรมภาษาซี ก่อนใช้แบบฝึกทักษะ นักเรียนควรอ่านคำชี้แจงในการใช้ตามลำดับขั้นตอน ดังนี้

1. อ่านและทำความเข้าใจคำชี้แจงในการใช้แบบฝึกทักษะ
2. ศึกษาผลการเรียนรู้และจุดประสงค์ เพื่อให้ทราบว่าเมื่อจบบทเรียนแล้ว นักเรียนสามารถเรียนรู้อะไรบ้าง
3. ทำแบบทดสอบก่อนเรียน แล้วตรวจคำตอบที่เฉลยไว้ เพื่อให้รู้ว่าคุณมีความรู้พื้นฐานเกี่ยวกับเรื่องที่จะศึกษามากน้อยเพียงใด
4. ศึกษาแบบฝึกทักษะการเขียนโปรแกรมภาษาซี เล่มที่ 1 ความรู้เบื้องต้นเกี่ยวกับภาษาซี ตั้งแต่หน้าแรกจนถึงหน้าสุดท้าย
5. เมื่อทำแบบฝึกทักษะเรียบร้อยแล้ว ครูและนักเรียนช่วยกันตรวจคำตอบละช่วยกันแก้ไข เมื่อมีคำตอบผิด
6. ทำแบบทดสอบหลังเรียนเพื่อประเมินผลการเรียนรู้
7. นักเรียนต้องมีสมาธิและความซื่อสัตย์ต่อตนเอง ไม่ข้ามขั้นตอน มิฉะนั้นผลงานที่ได้จะไม่สมบูรณ์ ไม่ผ่านตามเกณฑ์การประเมิน
8. นักเรียนรับฟังผลการทำแบบทดสอบ และข้อเสนอแนะจากครู
9. ถ้านักเรียนและผู้สนใจต้องการเนื้อหาเพิ่มเติม สามารถค้นคว้าได้จากเอกสารอ้างอิง ในบรรณานุกรม

ศึกษาคำชี้แจงแล้ว
ปฏิบัติตามอย่าง
เคร่งครัดนะคะ





แบบฝึกทักษะ เรื่องคำสั่งแบบวนรอบการทำงาน



สาระสำคัญ

การเขียนโปรแกรมคอมพิวเตอร์ขึ้นมาเพื่อใช้ในการแก้ไขปัญหาต่าง ๆ ที่มีความซับซ้อนซึ่งผู้เขียนโปรแกรมต้องสามารถควบคุมทิศทางการทำงานได้ ดังนั้น โปรแกรมที่สร้างขึ้นต้องมีความสามารถควบคุมการทำงานของโปรแกรม แบบวนรอบการทำงานเดิมๆตามเงื่อนไขที่กำหนด



ผลการเรียนรู้

มีความรู้ความเข้าใจในการใช้คำสั่งวนรอบการทำงาน





จุดประสงค์การเรียนรู้

1. อธิบายรูปแบบการใช้คำสั้ง for ได้
2. อธิบายรูปแบบการใช้คำสั้ง while ได้
3. อธิบายรูปแบบการใช้คำสั้ง do – while ได้
4. เขียนโปรแกรมโดยใช้คำสั้งแบบเลือกทำตามเงื่อนไขได้ถูกต้อง



สาระการเรียนรู้

1. การใช้คำสั้ง for
2. การใช้คำสั้ง while
3. การใช้คำสั้ง do – while



พร้อมแล้วอย่าลืม
ทำแบบทดสอบ
ก่อนเรียนนะคะ





- คำชี้แจง 1, แบบปรนัยเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ
2. ให้นักเรียนทำเครื่องหมาย × หน้าคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว

1. จากโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
#include<stdio.h>
main()
{
    int x = 1, y = 1;
    while (x < 5)
        while (y < 5)
        {
            printf("Thailand\n");
            x++;
            y++;
        }
}
```

- | | |
|--------------------------------------|--------------------------------------|
| ก. พิมพ์คำว่า Thailand ออกมา 3 ครั้ง | ข. พิมพ์คำว่า Thailand ออกมา 4 ครั้ง |
| ค. พิมพ์คำว่า Thailand ออกมา 5 ครั้ง | ง. พิมพ์คำว่า Thailand ออกมา 6 ครั้ง |
2. คำสั่งประเภทใดที่โปรแกรมต้องตรวจสอบเงื่อนไขก่อนการทำ Statement ภายในลูป
- | | |
|---------------|-----------------------|
| ก. do - while | ข. do - while และ for |
| ค. for | ง. for และ while |



- ```
int x = 0;
do
{
 printf("\tx = %d ", ++x);
}
while (x<5);
```

5. จากส่วนของโปรแกรมต่อไปนี้ จะพิมพ์คำว่า Hi ออกมากี่ครั้ง

ก. 15 ครั้ง                  ข. 13 ครั้ง                  ค. 0 ครั้ง                  ง. 17 ครั้ง

- ก. 15 ครั้ง                  ข. 14 ครั้ง                  ค. 12 ครั้ง                  ง. 17 ครั้ง

- x = 0;**



```

 }
 printf("*");
 x++;
}
}

```

ก. \* \* \* \* \*

ข. \* \* \* \* \*

ค. \* \* \* \* \*

ง. \* \* \* \* \*

8. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```

int a = 1;
while (a < 11)
{
 a++;
 printf("%d ", ++a);
}

```

ก. 0 2 4 6 8 10

ข. 2 4 6 8 10

ค. 1 3 5 7 9 11

ง. 3 5 7 9 11

9. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือ y มีค่าเท่ากับเท่าใด

```

int x = 1, y = 1;
while (x < 5)
{
 ++x; y += x;
}
printf("%d", y);

```

ก. 12

ข. 13

ค. 14

ง. 15

10. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int i = 0;
do
{
 switch (i++)
 {
 case 0: putchar('A');
 case 1: putchar('B');
 case 2: putchar('C');
 }
}
while (i < 3);
```

ก. ABCABC

ข. ABCBCC

ค. BCBCBC

ง. BCABCA



ทำแบบทดสอบก่อนเรียน  
เสร็จแล้วเริ่มศึกษา  
บทเรียนกันเลยนะคะ





## เกณฑ์การให้คะแนนแบบทดสอบก่อนเรียน



การให้คะแนน พิจารณาจากการตอบคำถามแต่ละข้อ ข้อละ 1 คะแนน

|                  |             |
|------------------|-------------|
| ตอบถูก           | ให้ 1 คะแนน |
| ตอบผิด           | ให้ 0 คะแนน |
| ไม่ตอบ           | ให้ 0 คะแนน |
| ตอบมากกว่า 1 ข้อ | ให้ 0 คะแนน |



# คำสั่งแบบวนรอบการทำงาน



โดยทั่วไปแล้ว การทำงานของโปรแกรมคอมพิวเตอร์จะทำงานเรียงตามลำดับตั้งแต่คำสั่งแรกถึงคำสั่งสุดท้าย แต่เราสามารถให้คอมพิวเตอร์ทำงานซ้ำๆ ที่ชุดคำสั่งใดชุดคำสั่งหนึ่งโดยใช้คำสั่งควบคุมให้ทำงานซ้ำได้ บางครั้งจะเรียกว่าคำสั่งลูป ในภาษาซีมีคำสั่งให้ทำงานซ้ำอยู่ 3 คำสั่ง คือ คำสั่ง for, คำสั่ง while และคำสั่ง do - while

## การใช้คำสั่ง for

คำสั่ง for เป็นคำสั่งที่สั่งให้โปรแกรมมีการทำงานซ้ำๆ วนรอบจนกว่าเงื่อนไขที่กำหนดไว้เป็นเท็จ จึงออกจากคำสั่ง for ไปทำคำสั่งถัดไป ควรใช้คำสั่ง for ในกรณีที่ทราบจำนวนรอบของการทำงาน

### รูปแบบการใช้คำสั่ง for

```
for (initial; condition; change)
 statement;
```

หรือ

```
for (initial; condition; change)
{
 statement_1;
 statement_2;
 statement_3;

 statement_n;
}
```





โดยที่ initial คือ นิพจน์ที่ใช้กำหนดค่าเริ่มต้นให้กับตัวแปรที่จะใช้วนรอบ เช่น  $y = 0$  หมายถึง กำหนดให้ตัวแปร  $y$  มีค่าเริ่มต้นเป็น 0

condition คือ นิพจน์ที่ใช้ทดสอบเงื่อนไข ซึ่งจะมีค่าจริงหรือเท็จอย่างใดอย่างหนึ่งเท่านั้น

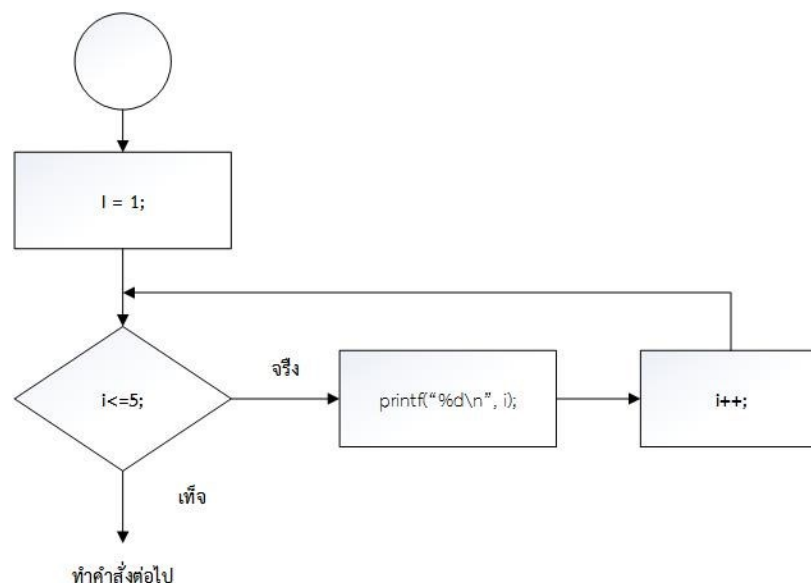
change คือ นิพจน์ที่ใช้เพิ่มหรือลดค่าตัวแปรที่จะใช้วนรอบ

statement คือ คำสั่งต่างๆ ถ้ามีมากกว่า 1 คำสั่ง จะต้องเขียนอยู่ภายในเครื่องหมาย {...}

ตัวอย่างเช่น ถ้าเขียนคำสั่ง for ดังต่อไปนี้

```
int i; //สร้างตัวแปร i เพื่อเก็บเลขจำนวนเต็ม
for(i = 1; i <= 5; i++) //กำหนดค่า i เท่ากับ 1 และกำหนดเงื่อนไขให้วนรอบ 5 รอบ
 printf("%d\n", i); //แต่ละรอบให้แสดงค่าของ i
```

เมื่อรันโปรแกรม โปรแกรมจะพิมพ์ค่าของ  $i$  ตั้งแต่ 1 ถึง 5 โดยการทำงานจะเริ่มต้นจากกำหนดค่าให้กับตัวแปร  $i$  เท่ากับ 1 จากนั้นนำค่า  $i$  ไปตรวจสอบกับเงื่อนไขว่า  $i$  น้อยกว่าหรือเท่ากับ 5 หรือไม่ ( $i \leq 5$ ); ถ้าเปรียบเทียบเงื่อนไขแล้วได้ผลลัพธ์เป็นจริง โปรแกรมจะพิมพ์ค่าใน  $i$  ออกมา จากนั้นจะทำการเพิ่มค่า  $i$  ขึ้นอีก 1 ( $i++$ ) ซึ่งตอนนี้  $i$  จะมีค่าเป็น 2 จากนั้นจะนำค่า  $i$  ไปเปรียบเทียบกับเงื่อนไขอีกครั้ง ซึ่งพบว่ายังคงเป็นจริงอยู่ จึงพิมพ์ค่า  $i$  ออกมา จากนั้นก็เพิ่มค่า  $i$  ขึ้นอีก 1 การทำงานจะวนรอบเช่นนี้ไปเรื่อยๆ จนกระทั่งค่าของ  $i$  มีค่าเท่ากับ 6 ซึ่งเมื่อนำไปเปรียบเทียบกับเงื่อนไขแล้ว จะพบว่าเป็นเท็จก็จะออกจากคำสั่ง for ซึ่งการทำงานของโปรแกรมนี้สามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 1 แสดงผังการทำงานของโปรแกรมที่ใช้คำสั่ง for



ผลการทำงานของโปรแกรมนี้คือ จะพิมพ์เลข 1 - 5 ดังนี้

1  
2  
3  
4  
5

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\example5_1.exe
1
2
3
4
5
```

นอกจากจะทำการเพิ่มค่าของตัวแปรแล้ว เรายังสามารถลดค่าของตัวแปรได้ด้วยดังส่วนของโปรแกรมต่อไปนี้

```
int i; //สร้างตัวแปร i เพื่อเก็บเลขจำนวนเต็ม
for(i = 5; i >= 0; i--) //กำหนดค่า i เท่ากับ 1 และกำหนดเงื่อนไขให้วนลูป 6 รอบ
 printf("%d\n", i); //แต่ละรอบให้แสดงค่าของ i
```

จากส่วนของโปรแกรมข้างต้น เมื่อโปรแกรมทำงานจะทำการวนรอบจำนวน 6 รอบ โดยแต่ละรอบจะลดค่าของตัวแปร i ลงครั้งละ 1 ดังนั้นโปรแกรมจึงพิมพ์เลข 5 จนถึงเลข 0

ผลการทำงานของโปรแกรมจะเป็นดังนี้

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\example5_2.exe
5
4
3
2
1
0
```



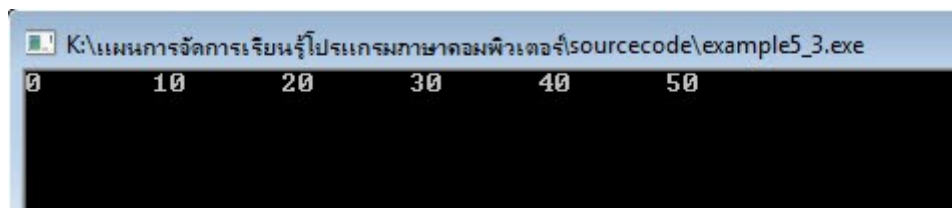


นอกจากนี้เรายังสามารถเพิ่มค่าหรือลดค่าให้กับตัวแปรที่ใช้ในการวนรอบ โดยการเพิ่มค่าหรือลดค่าครั้งละหลาย ๆ ค่าตามที่ต้องการได้ เช่น การเขียนคำสั่งต่อไปนี้จะพิมพ์เลข 0 ถึง 50 โดยมีการเพิ่มค่าในแต่ละรอบของการวนลูปครั้งละ 10 ค่า

```
int i;
for(i = 0; i <= 50; i = i + 10)
 printf("%d\t", i);
```

ผลการทำงานของโปรแกรมนี้จะพิมพ์ค่า 0 ถึง 50 โดยจะแสดงดังนี้

0      10      20      30      40      50

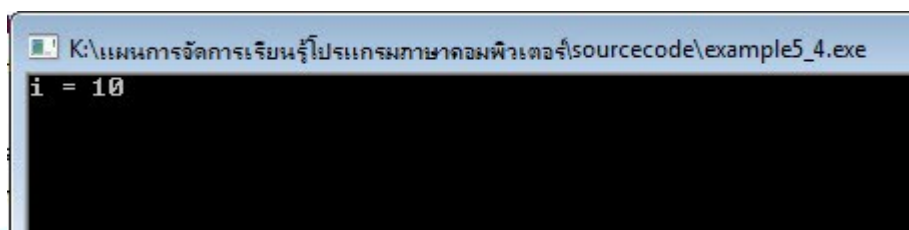


หากเรากำหนดเงื่อนไขเป็นเท็จ คอมพิวเตอร์จะไม่เข้าไปทำงานในลูป for เช่นคำสั่งต่อไปนี้

```
int i = 10, x;
for(x = 10; x != i ; x++) //กำหนดให้ x มีค่าเท่ากับ 10 แล้วนำไปเปรียบเทียบกับค่าใน i
{
 printf("x = %d\t", x); //คำสั่งภายในลูป
}
printf("i = %d", i); //คำสั่งนอกลูป
```

จากส่วนของโปรแกรมข้างต้น คอมพิวเตอร์จะไม่เข้าไปทำงานตามคำสั่งในลูป for เนื่องจากเงื่อนไขเป็นเท็จ ( $x \neq i$ ) แต่จะไปทำคำสั่งที่อยู่นอกลูป for คือทำคำสั่ง `printf("i = %d\t", i);` โปรแกรมจะพิมพ์ `i = 10`

ผลการทำงานของโปรแกรมจะเป็นดังนี้



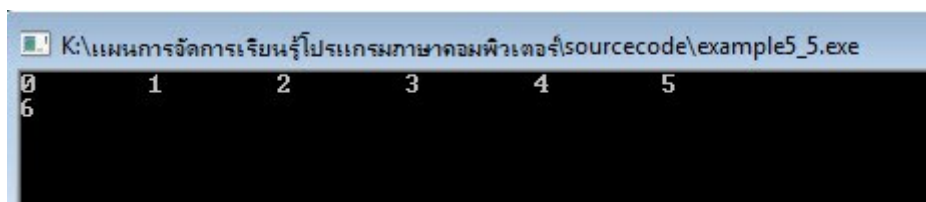


จากส่วนของโปรแกรมข้างล่างต่อไปนี้จะสังเกตว่าไม่มีส่วนของการเพิ่มค่าหรือลดค่าของตัวแปรที่ใช้ในการวนรอบในบรรทัดคำสั่ง for ดังนี้

```
int i;
for(i = 0; i <= 5;) //คำสั่ง for ที่ไม่มีส่วนที่สาม หรือส่วนที่ใช้ในการวนรอบ
{
 printf("%d\t",i); //พิมพ์ค่าของตัวแปร i
 i++; //เพิ่มค่าในตัวแปร i ขึ้นอีกหนึ่ง
}
printf("\n%d",i); //ขึ้นบรรทัดใหม่และพิมพ์ค่าในตัวแปร i
```

ผลการการทำงานของโปรแกรมจะเป็นดังนี้

```
0 1 2 3 4 5
6
```



การออกจากลูป for นอกจากการตรวจสอบเงื่อนไขว่าเป็นเท็จแล้ว เรายังสามารถใช้คำสั่ง break เพื่อสั่งให้ออกจากลูป for ได้ ดังตัวอย่างส่วนของโปรแกรมต่อไปนี้

```
int i , x = 2;
for(i = 1; i <= 10; i++) //กำหนดค่า i = 1 และกำหนดการวนลูปไว้ 10 รอบ
{
 printf("%d\t", i); //แสดงค่าในตัวแปร i
 if(i == 6) break; //ถ้าค่าในตัวแปร i เท่ากับ 6 ให้ออกจากลูป for
}
printf("\nx = %d", x); //ขึ้นบรรทัดใหม่และพิมพ์ค่าในตัวแปร x
```

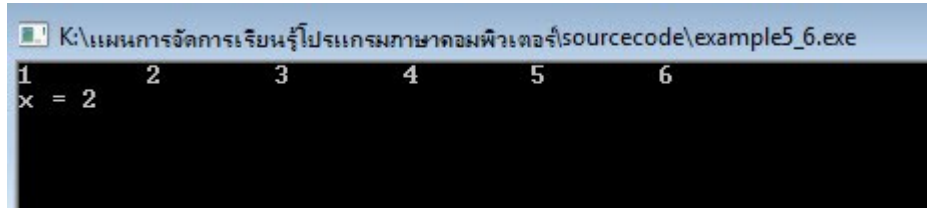




ผลการทำงานของโปรแกรมจะเป็นดังนี้

1      2      3      4      5      6

x = 2



คำสั่ง break; สามารถนำไปใช้งานร่วมกับคำสั่ง for, while, do - while และ switch ได้

### ตัวอย่างการใช้คำสั่ง for

โปรแกรมหาค่าเฉลี่ยของเลขที่ป้อนเข้ามา จำนวน 5 ค่า

```
#include<conio.h>
#include<stdio.h>
main()
{
 int i ; //สร้างตัวแปรชนิด int ชื่อ i ไว้สำหรับการวนนับลูป
 float number, average, sum = 0.0f ; /*สร้างตัวแปรชนิด float ชื่อ number ,
 average และ sum โดยกำหนดค่า
 sum เท่ากับ 0.0*/
 for (i=0; i<5; i++) /*ใช้คำสั่ง for วนลูปจำนวน 5 รอบ
 เพื่อรับค่า เข้ามา 5 จำนวน*/
 {
 printf("Enter number %d : ",i+1); //แสดงข้อความให้ป้อนตัวเลขเข้ามา
 scanf("%f",&number); /*รับค่าที่ป้อนเข้ามาเก็บไว้ในตัวแปร number
 sum += number; /*บวกค่าของตัวแปร sum กับค่าของ
 ตัวแปร number โดยผลลัพธ์เก็บไว้ในที่
 ตัวแปร sum*/
 }
 average = sum / 5; /*คำนวณค่าเฉลี่ยโดยหารค่าที่อยู่ในตัวแปร sum ด้วย 5
 ผลลัพธ์เก็บไว้ในตัวแปร average*/
 printf("Average is %.2f ", average); //แสดงค่าที่อยู่ในตัวแปร average
}
```





ผลการทำงานของโปรแกรมจะเป็นดังนี้

Enter number 1 : 6  
 Enter number 2 : 10  
 Enter number 3 : 8  
 Enter number 4 : 5  
 Enter number 5 : 7  
 Average is 7.20

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\example5_7.exe
Enter number 1: 6
Enter number 2: 10
Enter number 3: 8
Enter number 4: 5
Enter number 5: 7
Average is 7.20
```

## การใช้คำสั่ง while

while เป็นคำสั่งที่มีการทำงานซ้ำๆ เป็นลูปและมีลักษณะการทำงานของคำสั่งคล้ายกับคำสั่ง for แตกต่างกันตรงที่ใช้ while ไม่ต้องทราบจำนวนรอบของการทำงานที่แน่นอน แต่ต้องมีเงื่อนไขที่เป็นเท็จจึงจะออกจากคำสั่ง while ได้

### รูปแบบการใช้คำสั่ง while

while (condition) statement;

หรือ

while (condition)

{

statement\_1;

statement\_2;

statement\_3;





```
.....
statement_n;
}
```

โดยที่

condition คือ นิพจน์การกำหนดเงื่อนไขที่จะให้วนรอบการทำงานในลูป while  
statement คือ คำสั่งที่จะให้ทำงาน เมื่อเงื่อนไขเป็นจริง

### ข้อควรระวัง

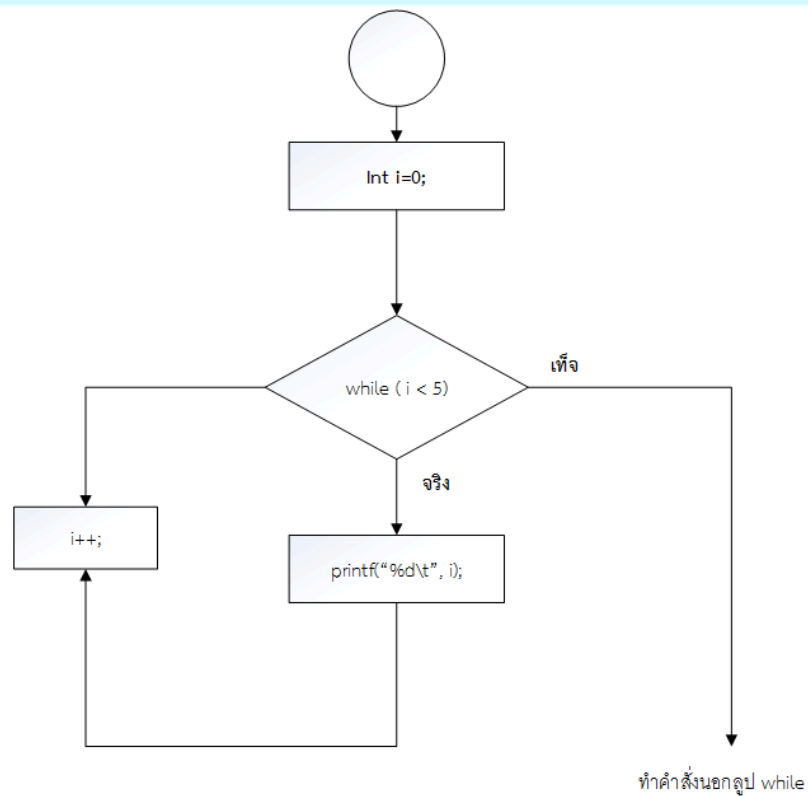
ภายในลูปจะต้องมีหนึ่งคำสั่ง ในการเพิ่ม ลด หรือ เปลี่ยนแปลงค่าของตัวแปรที่ใช้ตรวจสอบเงื่อนไข

ตัวอย่างเช่น ถ้าเขียนคำสั่ง while ดังต่อไปนี้

```
int i = 0; //สร้างตัวแปร i พร้อมกำหนดค่าเท่ากับ 0
while(i < 5) //ใช้คำสั่ง while กำหนดเงื่อนไข i น้อยกว่า 5
{
 printf("%d\t", i); //ถ้าเงื่อนไขเป็นจริงให้พิมพ์ค่า i
 i++; //เพิ่มค่า i ขึ้นอีก 1 แล้ววนลูปกลับไปตรวจสอบเงื่อนไข
}
```

เมื่อรันโปรแกรมจะทำการกำหนดค่า i เท่ากับ 0 แล้ว เข้าสู่คำสั่ง while เพื่อทำการตรวจสอบเงื่อนไข หากพบว่าเงื่อนไขเป็นจริงจะเข้ามาทำงานในลูปของ while คือ พิมพ์ค่าในตัวแปร i จากนั้นจะไปทำคำสั่ง i++ ซึ่งตอนนี้ค่า i จะมีค่าเป็น 1 แล้ววนลูปกลับขึ้นไปทำคำสั่ง while เพื่อตรวจสอบเงื่อนไข ซึ่งยังพบว่าเงื่อนไขเป็นจริงอยู่ก็จะเข้ามาทำคำสั่งในลูปของ while อีกครั้ง แล้วเพิ่มค่า i ขึ้นอีก 1 การทำงานจะวนลูปเช่นนี้ไปเรื่อย ๆ จนกระทั่งตรวจสอบเงื่อนไขแล้วพบว่าไม่เป็นเท็จ นั่นคือ i มีค่าเท่ากับ 5 ก็ จะออกจากลูป while ซึ่งการทำงานของโปรแกรมนี้สามารถเขียนเป็นผังงานได้ดังนี้



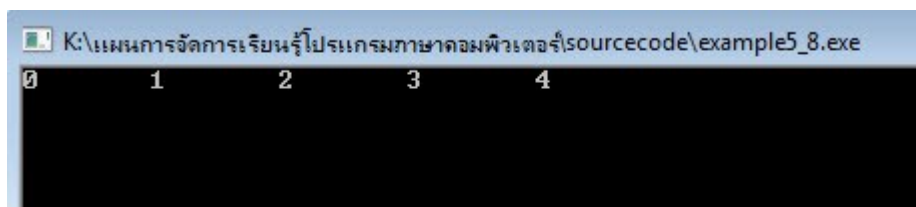


รูปที่ 2 แสดงผังการทำงานของโปรแกรมที่ใช้คำสั่ง while

จากรูปที่ 2 จะพบว่าคำสั่ง while นั้นจะทำการตรวจสอบเงื่อนไขก่อน ถ้าพบว่าเงื่อนไขเป็นจริงก็จะเข้าไปทำงานตามคำสั่งในลูปของ while แต่ถ้าเงื่อนไขเป็นเท็จก็จะออกจากคำสั่ง while ทันที

ผลการทำงานของโปรแกรมจะเป็นดังนี้

0      1      2      3      4





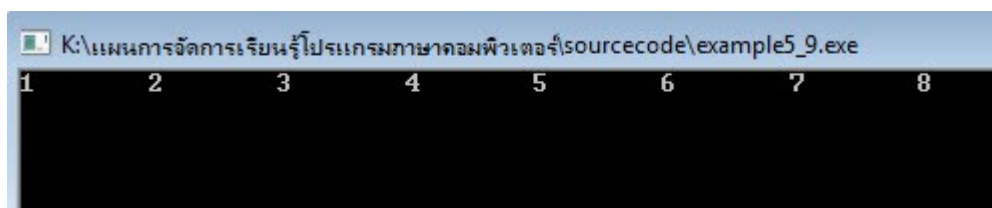


ตัวอย่างการใช้คำสั่ง while

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i = 1 ; //สร้างตัวแปรชื่อ i พร้อมกำหนดค่าเท่ากับ 1
 while (i <= 8) /*ใช้คำสั่ง while กำหนดเงื่อนไขวนรอบ
 การทำงาน トラバเท่าที่ค่าของ i ยังคงน้อยกว่าหรือ
 เท่ากับ 8*/
 {
 printf("%d\t" , i); //แสดงค่าที่อยู่ในตัวแปร i
 i ++ ; //เพิ่มค่าที่อยู่ในตัวแปร i ขึ้นไปอีกหนึ่ง
 }
 getch();
}
```

ผลการทำงานของโปรแกรมจะเป็นดังนี้

1      2      3      4      5      6      7      8





## การใช้คำสั่ง do - while

คำสั่ง do - while เป็นคำสั่งที่มีการทำงานซ้ำ ๆ วนรอบคล้ายกับคำสั่ง while มาก แตกต่างกันตรงที่คำสั่ง do - while จะทดสอบเงื่อนไขที่คำสั่ง while หลังจากที่ได้ทำงานตามคำสั่ง do ภายในลูปไปแล้ว 1 รอบ จากนั้นจึงค่อยย้อนกลับมาทดสอบเงื่อนไขอีกครั้งหนึ่ง ถ้าเงื่อนไขเป็นจริงก็จะทำงานตามคำสั่งภายในลูป แต่ถ้าเงื่อนไขเป็นเท็จจะออกจากคำสั่ง do - while ทันที

### รูปแบบการใช้ฟังก์ชัน do while

```
do
{
 statement_1;
 statement_2;
 statement_3;

 statement_N;
}
while (condition);
```

โดยที่

condition คือ นิพจน์การกำหนดเงื่อนไขที่จะให้วนการทำงานในลูป do - while  
statement คือ คำสั่งที่จะให้ทำงาน เมื่อเงื่อนไขเป็นจริง

ตัวอย่างเช่น ถ้าเขียนคำสั่ง do - while ดังต่อไปนี้

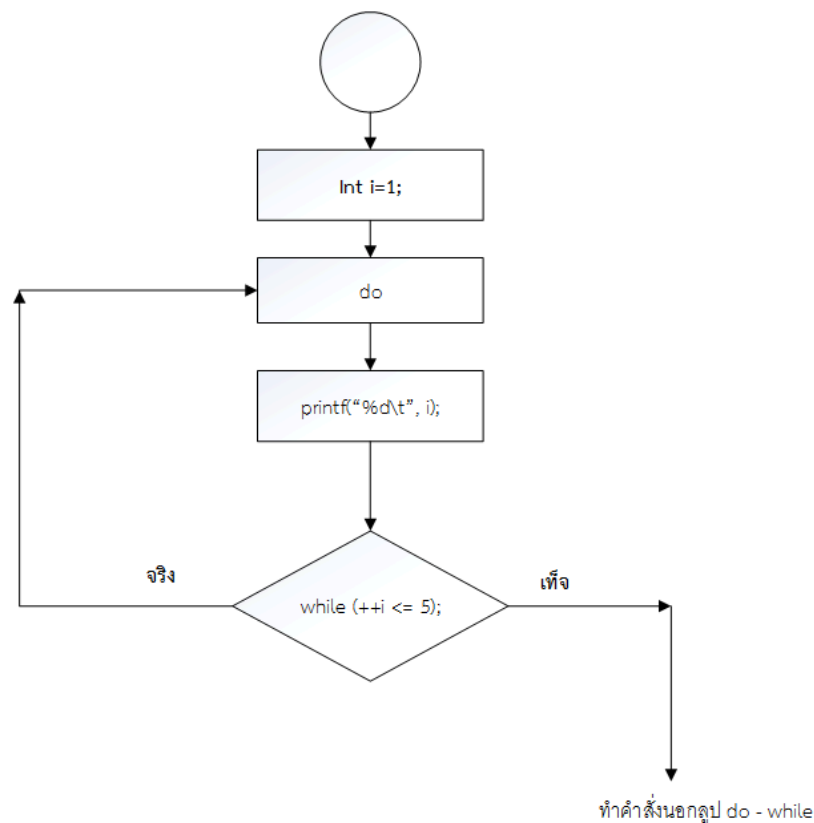
```
int i = 1; //สร้างตัวแปรชื่อ i พร้อมกำหนดค่าเท่ากับ 1
do //ให้ทำ
{
 printf("%d\t", i); //พิมพ์ค่าในตัวแปร i
}
while (++i <= 5); //เพิ่มค่า i ขึ้นอีก 1 แล้วตรวจสอบเงื่อนไข
```



ผลการทำงานของโปรแกรมจะเป็นดังนี้

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\example5_10.exe
1
2
3
4
5
```

เมื่อรันโปรแกรม จะทำการกำหนดค่า  $i$  เท่ากับ 1 แล้วเข้าสู่คำสั่ง `do` เพื่อทำการพิมพ์ค่าของ  $i$  จากนั้นเพิ่มค่า  $i$  ขึ้นอีก 1 ซึ่งตอนนี้  $i$  มีค่าเป็น 2 แล้วทำการตรวจสอบเงื่อนไขว่า  $i$  มีค่าน้อยกว่าหรือเท่ากับ 5 หรือไม่ ซึ่งพบว่าเงื่อนไขเป็นจริง จึงวนลูปกลับขึ้นไปทำคำสั่ง `do` ซึ่งจะพิมพ์ค่า  $i$  (ค่า 2) จากนั้นเพิ่มค่า  $i$  ขึ้นเป็น 3 แล้วตรวจสอบเงื่อนไข การทำงานจะวนลูปเช่นนี้ไปเรื่อย ๆ จน  $i$  มีค่าเท่ากับ 6 ซึ่งเมื่อตรวจสอบเงื่อนไขแล้วพบว่าไม่เป็นเท็จ จึงออกจากคำสั่ง `do - while` และการทำงานของโปรแกรมนี้สามารถเขียนเป็นผังงานได้ดังนี้



รูปที่ 3 แสดงผังการทำงานของโปรแกรมที่ใช้คำสั่ง `do - while`

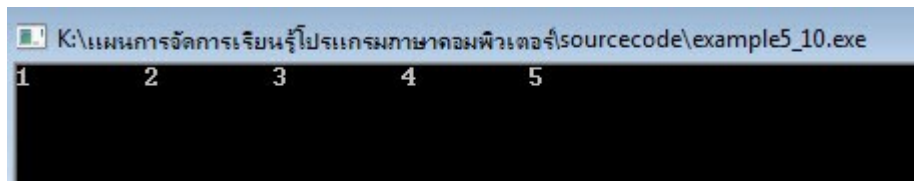




จากรูปที่ 3 จะพบว่าคำสั่ง do - while นั้นจะทำงานตามชุดคำสั่งภายในลูปก่อน 1 รอบ จากนั้นจึงไปตรวจสอบเงื่อนไข หากพบว่าเงื่อนไขเป็นจริงจะกลับเข้าไปทำงานตามชุดคำสั่งภายในลูป แต่ถ้าหากพบว่าเงื่อนไขเป็นเท็จก็จะออกจากลูปทันที

ผลการทำงานของโปรแกรมจะเป็นดังนี้

1      2      3      4      5



ตัวอย่างการใช้คำสั่ง do - while

โปรแกรมนับเลขจาก 1 - 10

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
main()
```

```
{
```

```
 int num = 1 ; /*สร้างตัวแปรแบบจำนวนเต็มชื่อ num
```

```
 โดยกำหนดให้มีค่าเท่ากับ 1*/
```

```
 do /*ใช้คำสั่ง do วนรอบการทำงานอย่างน้อย 1 รอบ
```

```
 {
```

```
 printf ("%d\t",num); //แสดงค่าที่อยู่ในตัวแปร num
```

```
 num++; //เพิ่มค่าในตัวแปร num ขึ้นอีกหนึ่ง
```

```
 }
```

```
 while (num<=10); /*กำหนดเงื่อนไข ถ้าเงื่อนไขเป็นจริงจะวนกลับไป
```

```
 ทำงานที่คำสั่ง do อีกรอบจนกว่าเงื่อนไขจะ
```

```
 เป็นเท็จ จึงออกจากลูป*/
```

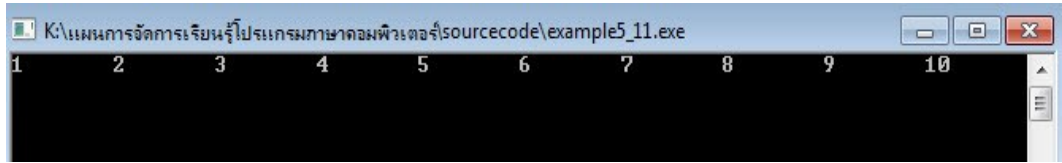
```
}
```





ผลการทำงานของโปรแกรมจะเป็นดังนี้

1      2      3      4      5      6      7      8      9      10



เราได้เรียนรู้วิธีการใช้คำสั่ง break มาบ้างแล้ว ยังมีคำสั่งอีกคำสั่งหนึ่งที่ทำงานตรงกันข้ามกับคำสั่ง break นั่นคือคำสั่ง continue ซึ่งสามารถใช้ร่วมกับคำสั่ง for, while หรือ do - while ได้เช่นเดียวกัน โดยเมื่อโปรแกรมทำงานมาถึงคำสั่ง continue จะทำคำสั่งภายในลูปต่อไปโดยไม่ทำคำสั่งที่ตามหลัง continue ดังตัวอย่างโปรแกรมต่อไปนี้

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i, x = 2;
 for (i = 1; i <= 10; i++) //กำหนดค่า i = 1 และกำหนดการวนลูปไว้ 10 รอบ
 {
 if(i == 5) //ถ้า i เท่ากับ 5 ให้ทำคำสั่ง continue
 continue; //วนกลับขึ้นไปทำคำสั่ง i++ ที่บรรทัด for
 printf ("%d\t",i); //แสดงค่าในตัวแปร i
 }
 printf("\nx = %d", x); /*เป็นคำสั่งนอกลูป for โดยสั่งให้ขึ้นบรรทัดใหม่และ
 พิมพ์ค่าในตัวแปร x */
 getch();
}
```





ผลการทำงานของโปรแกรมจะเป็นดังนี้

```
1 2 3 4 6 7 8 9 10
x = 2
```

จากผลการรันโปรแกรม เราจะพบว่าโปรแกรมจะไม่พิมพ์เลข 5 ออกมา แต่จะข้ามไปพิมพ์เลข 6 เลย ทั้งนี้เนื่องจากเมื่อ  $i$  เท่ากับ 5 จะทำให้เงื่อนไขในคำสั่ง `if (i == 5)` เป็นจริง จึงไปทำงานที่คำสั่ง `continue` นั่นคือย้อนกลับขึ้นไปทำคำสั่ง `for` โดยไม่ลง去做คำสั่ง `printf ("%d\t",i);` นอกจากนี้ในภาษายังมีฟังก์ชันที่ใช้สำหรับออกจากการทำงานของลูปนั่นคือฟังก์ชัน `exit()` ซึ่งการใช้ฟังก์ชัน `exit()` ต้องทำการเรียกใช้ไฟล์ `stdlib.h` เข้ามาไว้ในส่วนหัวของโปรแกรมก่อน ดังตัวอย่างส่วนของโปรแกรมต่อไปนี้

```
if (y == 0) //กำหนดเงื่อนไขถ้า y เท่ากับ 0
exit(0); //ให้ออกจากโปรแกรม
```





## สรุปท้ายบท



### คำสั่งแบบวนรอบการทำงาน ได้แก่

คำสั่ง for เป็นคำสั่งที่สั่งให้โปรแกรมมีการทำงานซ้ำๆ วนรอบจนกว่าเงื่อนไขที่กำหนดไว้เป็นเท็จ จึงออกจากคำสั่ง for ไปทำคำสั่งถัดไป และควรใช้คำสั่ง for ในกรณีที่ทราบจำนวนรอบของการทำงาน

คำสั่ง while เป็นคำสั่งที่สั่งให้โปรแกรมมีการทำงานซ้ำ ๆ เป็นลูป และมีลักษณะการทำงานคล้ายกับคำสั่ง for แตกต่างกันตรงที่ใช้ while ไม่ต้องทราบจำนวนรอบของการทำงานที่แน่นอน แต่ต้องมีเงื่อนไขเป็นเท็จจึงจะออกจากคำสั่ง while ได้

คำสั่ง do - while เป็นคำสั่งที่สั่งให้โปรแกรมมีการทำงานซ้ำ ๆ เป็นลูป และมีลักษณะการทำงานคล้ายกับคำสั่ง while แตกต่างกันตรงที่คำสั่ง do - while จะทดสอบเงื่อนไขที่คำสั่ง while หลังจากที่ได้ทำงานตามคำสั่ง do ภายในลูปไปแล้ว 1 รอบ จากนั้นจึงค่อยย้อนกลับมาทดสอบเงื่อนไขอีกครั้งหนึ่ง ถ้าเงื่อนไขเป็นจริงก็จะทำงานตามคำสั่งภายในลูป แต่ถ้าเงื่อนไขเป็นเท็จจะออกจากคำสั่ง do - while





คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง for (2 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....







2. พิมพ์โปรแกรมแสดงตัวอักษร A ถึง Z โดยใช้คำสั่ง for บันทึกชื่อไฟล์เป็น work5\_1.cpp

```
#include<stdio.h>
#include<conio.h>
main()
{
char i;
for(i = 'A'; i <= 'Z';i++)
{
printf("%c ",i);
}
}
```

ผลการทำงานของโปรแกรมจะเป็นดังนี้ (1 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. พิมพ์โปรแกรมแสดงการใช้คำสั่ง for ซ้อน for หรือเรียกว่า nested for โดยตัวแปร i จะทำงานควบคุมลูปนอก (Outer Loop) ส่วนตัวแปร j จะทำงานควบคุมลูปใน (Inner Loop) บันทึกชื่อไฟล์เป็น work5\_2.cpp พร้อมเขียนคำอธิบายของคำสั่งแต่ละบรรทัด (6 คะแนน)

```
#include<stdio.h>
#include<conio.h>
main()
{
int i, j;
for(i = 1; i <= 3;i++)
```

.....

.....





```
{
 printf("Outer loop %d\n",i);
 for (j = 1; j <= i; j++)
 printf("Inner loop %d\n", j);
 printf("\n");
}
getch();
}
```

4. ผลการทำงานของโปรแกรม (1 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....





## เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.1



คะแนนเต็ม 10 คะแนน การให้คะแนนพิจารณาจากเกณฑ์การตอบคำถามแต่ละข้อ

ข้อละ 2 คะแนน

คะแนน 2 เขียนรูปแบบคำสั่งได้ถูกต้องและอธิบายรายละเอียด

คะแนน 1 เขียนรูปแบบคำสั่งได้ถูกต้องแต่ไม่อธิบายรายละเอียด

คะแนน 0 ตอบคำถามไม่ถูกต้อง/ไม่ตอบคำถาม

ข้อละ 1 คะแนน

คะแนน 1 ตอบคำถามได้ถูกต้อง

คะแนน 0 ตอบคำถามไม่ถูกต้อง

## เกณฑ์การตัดสินคุณภาพ



| ช่วงคะแนน | ระดับคุณภาพ |
|-----------|-------------|
| 9 - 10    | ดีมาก       |
| 6 - 8     | ดี          |
| 4 - 5     | พอใช้       |
| 0 - 3     | ควรปรับปรุง |

เกณฑ์การผ่าน ตั้งแต่ระดับคุณภาพดีขึ้นไป





คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง while (2 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. พิมพ์โปรแกรมบวกเลขโดยเริ่มจาก 0 จนถึงค่าสุดท้ายที่ป้อนเข้าสู่โปรแกรม โดยใช้คำสั่ง while บันทึกชื่อไฟล์เป็น work5\_3.cpp พร้อมเขียนคำอธิบายของคำสั่งแต่ละบรรทัด (7 คะแนน)

```
#include<stdio.h>
#include<conio.h>
int begin = 0 , end , sum ;
main()
{
 printf ("Enter end number : ");
 scanf("%d",&end);
 while (begin <= end)
```





```
{
 sum = sum+begin;
 begin ++;
}
printf("Sum = %d" , sum);
}
```

3. ผลการทำงานของโปรแกรม (1 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....





## เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.2



### ข้อ 1 2 คะแนน โดยกำหนดเกณฑ์ดังนี้

คะแนน 2 เขียนรูปแบบคำสั่งได้ถูกต้องและอธิบายรายละเอียด

คะแนน 1 เขียนรูปแบบคำสั่งได้ถูกต้องแต่ไม่อธิบายรายละเอียด

คะแนน 0 ตอบคำถามไม่ถูกต้อง/ไม่ตอบคำถาม

### ข้อ 2 รวม 7 คะแนน โดยแยกคำถามแต่ละส่วน ดังนี้

คะแนน 1 ถ้าเขียนคำอธิบายถูกต้อง

คะแนน 0 เขียนคำอธิบายไม่ถูกต้องหรือไม่เขียนคำอธิบาย

### ข้อ 3 1 คะแนน โดยกำหนดเกณฑ์ดังนี้

คะแนน 1 ผลการทำงานของโปรแกรมถูกต้องตามเงื่อนไขที่กำหนด

คะแนน 0 ผลการทำงานของโปรแกรมไม่ถูกต้องตามเงื่อนไขที่กำหนดหรือ  
โปรแกรมทำงานไม่ได้

## เกณฑ์การตัดสินคุณภาพ



| ช่วงคะแนน | ระดับคุณภาพ |
|-----------|-------------|
| 9 - 10    | ดีมาก       |
| 6 - 8     | ดี          |
| 4 - 5     | พอใช้       |
| 0 - 3     | ควรปรับปรุง |

เกณฑ์การผ่าน ตั้งแต่ระดับคุณภาพดีขึ้นไป





คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง do – while (2 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....





2. พิมพ์โปรแกรมสูตรคูณ โดยใช้คำสั่ง do - while บันทึกชื่อไฟล์เป็น work5\_4.cpp (7 คะแนน)

```
#include<stdio.h>
#include<conio.h>
main()
{
 int num,count = 1;.....
 printf("Program Sutkun\n"); //แสดงข้อความชื่อโปรแกรม
 printf("=====\n");
 printf("Enter sutkun mother : ");
 scanf("%d",&num);
 do
 {
 printf("%3d X %d = %d\n",num,count,count*num);

 count++;
 }
 while(count<=12);

 getch();
}
```

3. ผลการทำงานของโปรแกรม (1 คะแนน)

.....

.....

.....

.....

.....

.....







## เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.3



### ข้อ 1 2 คะแนน โดยกำหนดเกณฑ์ดังนี้

คะแนน 2 เขียนรูปแบบคำสั่งได้ถูกต้องและอธิบายรายละเอียด

คะแนน 1 เขียนรูปแบบคำสั่งได้ถูกต้องแต่ไม่อธิบายรายละเอียด

คะแนน 0 ตอบคำถามไม่ถูกต้อง/ไม่ตอบคำถาม

### ข้อ 2 รวม 7 คะแนน โดยแยกคำถามแต่ละส่วน ดังนี้

คะแนน 1 ถ้าเขียนคำอธิบายถูกต้อง

คะแนน 0 เขียนคำอธิบายไม่ถูกต้องหรือไม่เขียนคำอธิบาย

### ข้อ 3 1 คะแนน โดยกำหนดเกณฑ์ดังนี้

คะแนน 1 ผลการทำงานของโปรแกรมถูกต้องตามเงื่อนไขที่กำหนด

คะแนน 0 ผลการทำงานของโปรแกรมไม่ถูกต้องตามเงื่อนไขที่กำหนดหรือ  
โปรแกรมทำงานไม่ได้

## เกณฑ์การตัดสินคุณภาพ



| ช่วงคะแนน | ระดับคุณภาพ |
|-----------|-------------|
| 9 - 10    | ดีมาก       |
| 6 - 8     | ดี          |
| 4 - 5     | พอใช้       |
| 0 - 3     | ควรปรับปรุง |

เกณฑ์การผ่าน ตั้งแต่ระดับคุณภาพดีขึ้นไป





คำชี้แจง ให้นักเรียนฝึกปฏิบัติเขียนโปรแกรม (10 คะแนน)

1. จงเขียนโปรแกรมเพื่อหาค่าเฉลี่ยของเลขจำนวนเต็มตามจำนวนที่ผู้ใช้โปรแกรมป้อนเข้ามา โดยใช้ คำสั่ง for และให้ตั้งชื่อไฟล์ว่า test5\_1.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_1.exe
Enter numbers to the average: 5
Enter number 1: 25
Enter number 2: 35
Enter number 3: 28
Enter number 4: 49
Enter number 5: 85
Average is 44.40
```

2. จงเขียนโปรแกรมรับเลขจำนวนเต็มจากผู้ใช้งานจนกว่าผู้ใช้จะป้อนเลข -9999 ให้หาว่ามีเลขจำนวนเต็มลบ (ไม่รวม -9999) อยู่กี่จำนวน และมีจำนวนเต็มบวก (รวมทั้งเลขศูนย์) อยู่กี่จำนวน โดยใช้ คำสั่ง while และให้ตั้งชื่อไฟล์ว่า test5\_2.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_2.exe
Enter number (<-9999 for end): 985
Enter number (<-9999 for end): 457
Enter number (<-9999 for end): 0
Enter number (<-9999 for end): -432
Enter number (<-9999 for end): -56
Enter number (<-9999 for end): -9999
Number less than zero = 2, more than or equal to zero = 3
```





3. จงเขียนโปรแกรมรับเลขจำนวนเต็มจากผู้ใช้ จะหยุดรับข้อมูลเมื่อผลรวมของเลขที่รับเข้ามามีค่ามากกว่า 200 และให้หาว่ามีการรับเลขจำนวนเต็มเข้ามามากี่จำนวน และผลรวมของจำนวนทั้งหมด โดยใช้คำสั่ง do- while และให้ตั้งชื่อไฟล์ว่า test5\_3.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_3.exe
Enter number: 25
Enter number: 34
Enter number: 68
Enter number: 76
Enter number 4 time
Sum of number 203
```

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_3.exe
Enter number: 12
Enter number: 15
Enter number: 14
Enter number: 45
Enter number: 15
Enter number: 18
Enter number: 19
Enter number: 20
Enter number: 50
Enter number 9 time
Sum of number 208
```

4. จงเขียนโปรแกรมเพื่อหาค่าเฉลี่ยของเลขจำนวนเต็ม 5 จำนวนตามที่ใช้โปรแกรมป้อนเข้ามา ให้หาตัวเลขที่มีค่ามากที่สุด และน้อยที่สุด โดยใช้คำสั่ง for และให้ตั้งชื่อไฟล์ว่า test5\_4.cpp

ตัวอย่างผลการทำงานโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\posttest5_1.exe
Enter number 1 : 250
Enter number 2 : -120.35
Enter number 3 : 85
Enter number 4 : 120.50
Enter number 5 : -56.80
Average number = 55.67
Max number = 250.00
Min number = -120.35
```





5. จงเขียนโปรแกรมคำนวณเงินฝากและดอกเบี้ย โดยโปรแกรมจะให้ผู้ใส่ป้อนค่าเงินต้น (Capital) อัตราดอกเบี้ย (Rate) และจำนวนปีที่ฝากเงินเข้ามา (Year) แล้วคำนวณโดยใช้สูตรดังต่อไปนี้  
 ดอกเบี้ย (Interest) =  $\text{capital} * \text{rate} / 100$  เงินต้น (capital) += interest และให้ตั้งชื่อไฟล์ว่า  
 pretest5\_5.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_4.exe
Capital : 10000
Interest rate : 5
Year : 3
Year Interest Sum
1 500.00 10500.00
2 525.00 11025.00
3 551.25 11576.25
4 578.81 12155.06
```





## เกณฑ์การให้คะแนนแบบฝึกทักษะที่ 5.4



พิจารณาการให้คะแนน จากการเขียนโปรแกรมในแต่ละข้อ ข้อละ 2 คะแนน

คะแนน 2 รันโปรแกรมผ่านและถูกต้อง

คะแนน 1 รันโปรแกรมได้แต่ยังมีข้อผิดพลาด

คะแนน 0 ไม่สามารถรันโปรแกรมได้

## เกณฑ์การตัดสินคุณภาพ



| ช่วงคะแนน | ระดับคุณภาพ |
|-----------|-------------|
| 9 - 10    | ดีมาก       |
| 6 - 8     | ดี          |
| 4 - 5     | พอใช้       |
| 0 - 3     | ควรปรับปรุง |

เกณฑ์การผ่าน ตั้งแต่ระดับคุณภาพดีขึ้นไป





- คำชี้แจง 1. แบบปรนัยเลือกตอบ 4 ตัวเลือก จำนวน 10 ข้อ  
2. ให้นักเรียนทำเครื่องหมาย x หน้าคำตอบที่ถูกต้องที่สุดเพียงข้อเดียว

1. จากโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
#include<stdio.h>
main()
{
 int x = 1, y = 1;
 while (x < 5)
 while (y < 5)
 {
 printf("Thailand\n");
 x++;
 y++;
 }
}
```

- ก. พิมพ์คำว่า Thailand ออกมา 6 ครั้ง  
ค. พิมพ์คำว่า Thailand ออกมา 3 ครั้ง

- ข. พิมพ์คำว่า Thailand ออกมา 5 ครั้ง  
ง. พิมพ์คำว่า Thailand ออกมา 4 ครั้ง





2. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int x = 0;
do
{
 printf("\tx = %d ",++x);
}
while (x<5);
```

ก.  $x = 0 \ x = 1 \ x = 2 \ x = 3 \ x = 4$

ข. เป็นการวนลูปแบบไม่รู้จบ

ค.  $x = 1 \ x = 2 \ x = 3 \ x = 4$

ง.  $x = 1 \ x = 2 \ x = 3 \ x = 4 \ x = 5$

3. จากส่วนของโปรแกรมต่อไปนี้ จะพิมพ์คำว่า Hi ออกมากี่ครั้ง

```
for (i = 1; i <15; i++) printf("Hi\n");
```

ก. 14 ครั้ง

ข. 17 ครั้ง

ค. 15 ครั้ง

ง. 12 ครั้ง

4. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int a = 0;
while (++a <= 12)
{
 if (a == 5)
 break;
 printf("%d",a);
}
```

ก. 012345

ข. 123456

ค. 1234

ง. 01234

5. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int i = 0;
do
{
 switch (i++)
 {
 case 0: putchar('A');
 case 1: putchar('B');
 case 2: putchar('C');
 }
} while (i < 3);
```





ก. BCBCBC

ข. BCABCA

ค. ABCABC

ง. ABCBCC

6. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือ y มีค่าเท่ากับเท่าใด

```
int x = 1, y = 1;
while (x < 5)
{
 x++; y += x;
}
printf("%d",y);
```

ก. 14

ข. 15

ค. 12

ง. 13

7. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int x, y;
for (x = 1, y = 12; x < 12, y > 5; x++, y--)
 printf("%d",x);
```

ก. 1234567

ข. 12345

ค. 1357911

ง. 1246810

8. จากส่วนของโปรแกรมต่อไปนี้ จะพิมพ์คำว่า Hi ออกมากี่ครั้ง

```
for (i = 0; i >=15; i+2) printf("Hi\n");
```

ก. 17 ครั้ง

ข. 0 ครั้ง

ค. 13 ครั้ง

ง. 15 ครั้ง

9. จากส่วนของโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
int a = 1;
while (a < 11)
{
 a++;
 printf("%d ",++a);
}
```

ก. 3 5 7 9 11

ข. 1 3 5 7 9 11

ค. 2 4 6 8 10

ง. 0 2 4 6 8 10







10. จากโปรแกรมต่อไปนี้ ผลลัพธ์จากการทำงานของโปรแกรมคือข้อใด

```
#include<stdio.h>
main()
{
 int x = 0; int y = 5;
 while (y > 1)
 {
 if ((x % y) == 0)
 {
 printf(" ");
 y--;
 x = 0;
 }
 printf("*");
 x++;
 }
}
```

ก. \*\* \*\*\* \*\*\*\*\*

ค. \* \*\* \*\*\* \*\*\*\*\*

ข. \*\*\*\*\* \*\* \*

ง. \*\*\*\*\* \*\*\*\*\* \*\*



ขอให้ทุกคนโชคดี  
ในการสอบครั้งนี้  
นะจ๊ะ





## เกณฑ์การให้คะแนนแบบทดสอบหลังเรียน



การให้คะแนน พิจารณาจากการตอบคำถามแต่ละข้อ ข้อละ 1 คะแนน

|                  |             |
|------------------|-------------|
| ตอบถูก           | ให้ 1 คะแนน |
| ตอบผิด           | ให้ 0 คะแนน |
| ไม่ตอบ           | ให้ 0 คะแนน |
| ตอบมากกว่า 1 ข้อ | ให้ 0 คะแนน |

## เกณฑ์การตัดสินคุณภาพ



| ช่วงคะแนน | ระดับคุณภาพ |
|-----------|-------------|
| 9 - 10    | ดีมาก       |
| 6 - 8     | ดี          |
| 4 - 5     | พอใช้       |
| 0 - 3     | ควรปรับปรุง |

เกณฑ์การผ่าน ตั้งแต่ระดับคุณภาพดีขึ้นไป





- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. ข | 2. ง | 3. ค | 4. ค | 5. ค  |
| 6. ก | 7. ง | 8. ง | 9. ง | 10. ข |



- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. ง | 2. ง | 3. ค | 4. ค | 5. ง  |
| 6. ข | 7. ก | 8. ข | 9. ก | 10. ข |





คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง for (2 คะแนน)

รูปแบบการใช้ฟังก์ชัน for

```
for (initial; condition; change)
 statement;
```

หรือ

```
for (initial; condition; change)
{
 statement_1;
 statement_2;
 statement_3;

 statement_n;
}
```

โดยที่

initial คือ นิพจน์ที่ใช้กำหนดค่าเริ่มต้นให้กับตัวแปรที่จะใช้วนลูป เช่น `y = 0`  
หมายถึง กำหนดให้ตัวแปร `y` มีค่าเริ่มต้นเป็น 0

condition คือ นิพจน์ที่ใช้ทดสอบเงื่อนไข ซึ่งจะมีค่าจริงหรือเท็จอย่างใดอย่าง  
หนึ่งเท่านั้น

change คือ นิพจน์ที่ใช้เพิ่มหรือลดค่าตัวแปรที่จะใช้วนลูป





2. พิมพ์โปรแกรมแสดงตัวอักษร A ถึง Z โดยใช้คำสั่ง for บันทึกชื่อไฟล์เป็น work5\_1.cpp

```
#include<stdio.h>
#include<conio.h>
main()
{
 char i;
 for(i = 'A'; i <= 'Z';i++)
 {
 printf("%c ",i);
 }
}
```

ผลการทำงานของโปรแกรมจะเป็นดังนี้ (1 คะแนน)



3. พิมพ์โปรแกรมแสดงการใช้คำสั่ง for ซ้อน for หรือเรียกว่า nested for โดยตัวแปร i จะทำงานควบคุม ลูปนอก (Outer Loop) ส่วนตัวแปร j จะทำงานควบคุมลูปใน (Inner Loop) บันทึกชื่อไฟล์เป็น work5\_2.cpp พร้อมเขียนคำอธิบายของคำสั่งแต่ละบรรทัด (6 คะแนน)

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i, j; //สร้างตัวแปร i และ j
 for(i = 1; i <= 3;i++) //กำหนดให้ i ควบคุมการวนลูปนอกจำนวน 3 รอบ
 {
 printf("Outer loop %d\n",i); //แสดงลำดับของลูปนอกในแต่ละรอบของลูปนอก
 for (j = 1; j <= i; j++) //แสดงลำดับของลูปในในแต่ละรอบ
```





```
printf("Inner loop %d\n", j); //แสดงลำดับของลูปในในแต่ละรอบ
printf("\n"); //ในแต่ละรอบของลูปนอกให้ขึ้นบรรทัดใหม่
}
getch();
}
```

4. ผลการทำงานของโปรแกรม (1 คะแนน)

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\work5_2.exe
Outer loop 1
Inner loop 1

Outer loop 2
Inner loop 1
Inner loop 2

Outer loop 3
Inner loop 1
Inner loop 2
Inner loop 3
```





คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง while (2 คะแนน)

รูปแบบการใช้คำสั่ง while

```
while (condition) statement;
```

หรือ

```
while (condition)
```

```
{
```

```
 statement_1;
```

```
 statement_2;
```

```
 statement_3;
```

```

```

```
 Statement_n;
```

```
}
```

โดยที่

condition คือ นิพจน์การกำหนดเงื่อนไขที่จะให้วนการทำงานในลูป while

statement คือ คำสั่งที่จะให้ทำงาน เมื่อเงื่อนไขเป็นจริง

2. พิมพ์โปรแกรมบวกเลขโดยเริ่มจาก 0 จนถึงค่าสุดท้ายที่ป้อนเข้าสู่โปรแกรม โดยใช้คำสั่ง while บันทึกชื่อไฟล์เป็น work5\_3.cpp พร้อมเขียนคำอธิบายของคำสั่งแต่ละบรรทัด (7 คะแนน)

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
int begin = 0 , end , sum ; //สร้างตัวแปรเพื่อเก็บตัวเลขเริ่มต้น ตัวเลขตัวสุดท้าย และผลรวม
```

```
main()
```

```
{
```

```
 printf ("Enter end number : "); //แสดงข้อความให้ป้อนตัวเลขตัวสุดท้าย
```





```
scanf("%d",&end); //รับค่าตัวเลขตัวสุดท้ายมาเก็บไว้ที่ตัวแปร .end
while (begin <= end) /*ใช้คำสั่ง...while...กำหนดเงื่อนไขวนการทำงานตามเท่าที่
 ค่าในตัวแปร .begin ยังคงน้อยกว่าหรือเท่ากับค่าในตัวแปร .end*/
{
 sum = sum+begin; //ในแต่ละรอบให้คำนวณหาค่าในตัวแปร .sum
 begin ++; //เพิ่มค่าที่อยู่ในตัวแปร .begin ขึ้นไปรอบละหนึ่ง
}
printf("Sum = %d" , sum); //แสดงค่าที่อยู่ในตัวแปร .sum
}
```

### 3. ผลการทำงานของโปรแกรม (1 คะแนน)

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\work5_3.exe
Enter end number : 10
Sum = 55
```







คำชี้แจง ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงอธิบายรูปแบบการใช้คำสั่ง do - while (2 คะแนน)

รูปแบบการใช้ฟังก์ชัน do..while

do

{

statement\_1;

statement\_2;

statement\_3;

.....

statement\_N;

}

while (condition);

โดยที่

condition คือ นิพจน์การกำหนดเงื่อนไขที่จะให้วนการทำงานในลูป do..while

statement คือ คำสั่งที่จะให้ทำงาน เมื่อเงื่อนไขเป็นจริง





2. พิมพ์โปรแกรมสูตรคูณ โดยใช้คำสั่ง do - while บันทึกชื่อไฟล์เป็น work5\_4.cpp (7 คะแนน)

```
#include<stdio.h>
#include<conio.h>
main()
{
 int num,count = 1; /*สร้างตัวแปรชนิดจำนวนเต็มชื่อ num และ count
 โดยกำหนดให้มีค่าเท่ากับ 1 */

 printf("Program Sutkun\n"); //แสดงข้อความชื่อโปรแกรม
 printf("=====\n");
 printf("Enter sutkun mother : "); //แสดงข้อความให้ป้อนแม่สูตรคูณ
 scanf("%d",&num); //รับตัวเลขแม่สูตรคูณเข้ามาเก็บไว้ที่ตัวแปร num
 do //ใช้คำสั่ง do วนรอบการทำงานอย่างน้อย 1 รอบ
 {
 printf("%3d X %d = %d\n",num,count,count*num); /*แสดงค่าที่
 อยู่ในตัวแปร*/

 count++; //เพิ่มค่าในตัวแปร count ขึ้นอีกหนึ่ง
 }
 while(count<=12); /*กำหนดเงื่อนไข โดยตรวจสอบค่าในตัวแปร count
 ว่าน้อยกว่าหรือเท่ากับ 12 หรือไม่ ถ้าเงื่อนไขเป็นจริง
 จะวนกลับขึ้นไปทำงานที่บรรทัดคำสั่ง do อีกรอบ
 จนกว่าเงื่อนไขจะเป็นเท็จจึงออกจากลูป*/

 getch();
}
```

3. ผลการทำงานของโปรแกรม (1 คะแนน)

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\work5_4.exe
Program Sutkun
=====
Enter sutkun mother : 9
9 X 1 = 9
9 X 2 = 18
9 X 3 = 27
9 X 4 = 36
9 X 5 = 45
9 X 6 = 54
9 X 7 = 63
9 X 8 = 72
9 X 9 = 81
9 X 10 = 90
9 X 11 = 99
9 X 12 = 108
```





คำชี้แจง ให้นักเรียนฝึกปฏิบัติเขียนโปรแกรม (10 คะแนน)

1. จงเขียนโปรแกรมเพื่อหาค่าเฉลี่ยของเลขจำนวนเต็มตามจำนวนที่ผู้ใช้โปรแกรมป้อนเข้ามา โดยใช้ คำสั่ง For และให้ตั้งชื่อไฟล์ว่า test5\_1.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_1.exe
Enter numbers to the average: 5
Enter number 1: 25
Enter number 2: 35
Enter number 3: 28
Enter number 4: 49
Enter number 5: 85
Average is 44.40
```

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i, num, n;
 float average, sum = 0.0f;
 printf("Enter numbers to the average: ");
 scanf("%d",&n);
 for (i = 0; i < n; i++)
 {
 printf("Enter number %d: ",i + 1);
 scanf("%d",&num);
```





```

 sum += num;
 }
 average = sum / 5;
 printf("Average is %.2f",average);
 getch();

```

2. จงเขียนโปรแกรมรับเลขจำนวนเต็มจากผู้ใช้ จนกว่าผู้ใช้จะป้อนเลข -9999 ให้หาว่ามีเลขจำนวนเต็มลบ (ไม่รวม -9999) อยู่กี่จำนวน และมีจำนวนเต็มบวก (รวมทั้งเลขศูนย์) อยู่กี่จำนวน โดยใช้คำสั่ง while และให้ตั้งชื่อไฟล์ว่า test5\_2.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```

K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_2.exe
Enter number (-9999 for end): 985
Enter number (-9999 for end): 457
Enter number (-9999 for end): 0
Enter number (-9999 for end): -432
Enter number (-9999 for end): -56
Enter number (-9999 for end): -9999
Number less than zero = 2, more than or equal to zero = 3

```

```

#include<stdio.h>
#include<conio.h>
main()
{
 int plus = 0, minus = 0, num;
 printf("Enter number (-9999 for end): ");
 scanf("%d",&num);
 while (num != -9999)
 {
 if (num < 0)
 minus++;
 else

```





```

 plus++;
 printf("Enter number (-9999 for end): ");
 scanf("%d",&num);
 }
 printf("Number less than zero = %d, more than or equal to zero = %d",
 minus, plus);
}

```

3. จงเขียนโปรแกรมรับเลขจำนวนเต็มจากผู้ใช้ จะหยุดรับข้อมูลเมื่อผลรวมของเลขที่รับเข้ามามีค่ามากกว่า200 และให้หาว่ามีการรับเลขจำนวนเต็มเข้ามามากี่จำนวน และผลรวมของจำนวนทั้งหมด โดยใช้คำสั่ง do- while และให้ตั้งชื่อไฟล์ว่า test5\_3.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```

K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_3.exe
Enter number: 25
Enter number: 34
Enter number: 68
Enter number: 76
Enter number 4 time
Sum of number 203

```

```

K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_3.exe
Enter number: 12
Enter number: 15
Enter number: 14
Enter number: 45
Enter number: 15
Enter number: 18
Enter number: 19
Enter number: 20
Enter number: 50
Enter number 9 time
Sum of number 208

```

```

#include<stdio.h>
#include<conio.h>
main()
{
 int sum = 0, count = 0, num;
 do
 {

```





```

printf("Enter number: ");
scanf("%d",&num);
sum += num;
count++;
}
while (sum <= 200);
printf("Enter number %d time\n",count);
printf("Sum of number %d",sum);
getch();
}

```

4. จงเขียนโปรแกรมเพื่อหาค่าเฉลี่ยของเลขจำนวนเต็ม 5 จำนวนตามที่ใช้โปรแกรมป้อนเข้ามา  
ให้หาตัวเลขที่มีค่ามากที่สุด และน้อยที่สุด โดยใช้คำสั่ง for และให้ตั้งชื่อไฟล์ว่า test5\_4.cpp

```

#include<stdio.h>
#include<conio.h>
main()
{
 int i;
 float in, max = 0, min = 0, num = 0;
 for (i = 0; i < 5; i++)
 {
 printf("Enter number %d : ",i + 1);
 scanf("%f",&in);
 if (max < in)
 max = in;
 if (in < min)
 min = in;
 num += in;
 }
 printf("Average number = %.2f\n",num/5);
 printf("Max number = %.2f\n",max);
 printf("Min number = %.2f\n",min);
 getch();
}

```





```
}
```

ตัวอย่างผลการทำงานโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\posttest5_1.exe
Enter number 1 : 250
Enter number 2 : -120.35
Enter number 3 : 85
Enter number 4 : 120.50
Enter number 5 : -56.80
Average number = 55.67
Max number = 250.00
Min number = -120.35
```

5. จงเขียนโปรแกรมคำนวณเงินฝากและดอกเบี้ย โดยโปรแกรมจะให้ผู้ใช้ป้อนค่าเงินต้น (Capital) อัตราดอกเบี้ย (Rate) และจำนวนปีที่ฝากเงินเข้ามา (Year) แล้วคำนวณโดยใช้สูตรดังต่อไปนี้  
 ดอกเบี้ย (Interest) = capital \* rate / 100 เงินต้น (capital) += interest และให้ตั้งชื่อไฟล์ว่า  
 pretest5\_5.cpp

ตัวอย่างผลการทำงานของโปรแกรม

```
K:\แผนการจัดการเรียนรู้โปรแกรมภาษาคอมพิวเตอร์\sourcecode\pretest5_4.exe
Capital : 10000
Interest rate : 5
Year : 3
Year Interest Sum
1 500.00 10500.00
2 525.00 11025.00
3 551.25 11576.25
4 578.81 12155.06
```

```
#include<stdio.h>
#include<conio.h>
main()
{
 float capital, rate, interest;
 int i, year;
 printf("Capital : ");
 scanf("%f",&capital);
```





```
printf("Interest rate : ");
scanf("%f",&rate);
printf("Year : ");
scanf("%d",&year);
printf("Year\t\tInterest\t\tSum\n");
for(i = 0; i <= year; i++)
{
 interest = capital * rate / 100;
 capital += interest;
 printf("%2d %20.2f %22.2f\n",i + 1,interest,capital);
}
getch();
}
```





# แบบบันทึกคะแนน



## แบบฝึกทักษะเล่มที่ 5 คำสั่งแบบวนรอบการทำงาน

ชื่อ-สกุล.....ชั้น.....เลขที่.....

| แบบทดสอบ          | คะแนนเต็ม | คะแนนที่ได้ |
|-------------------|-----------|-------------|
| แบบทดสอบก่อนเรียน | 10        |             |
| แบบทดสอบหลังเรียน | 10        |             |
| คะแนนความก้าวหน้า |           |             |

| แบบฝึกทักษะ        | คะแนนเต็ม | คะแนนที่ได้ |
|--------------------|-----------|-------------|
| แบบฝึกทักษะที่ 5.1 | 10        |             |
| แบบฝึกทักษะที่ 5.2 | 10        |             |
| แบบฝึกทักษะที่ 5.3 | 10        |             |
| แบบฝึกทักษะที่ 5.4 | 10        |             |
| รวมคะแนน           | 40        |             |

ร้อยละของคะแนนรวม = (คะแนนรวมที่ได้  $\times$  100) / 40 = (..... $\times$  100)/40 = .....

เกณฑ์การให้คะแนนการทำแบบฝึกทักษะ

- (4) ระดับดีมาก หมายถึง ได้คะแนนคิดเป็นร้อยละ 80 ขึ้นไป
- (3) ระดับดี หมายถึง ได้คะแนนคิดเป็นร้อยละ 60 – 79
- (2) ระดับพอใช้ หมายถึง ได้คะแนนคิดเป็นร้อยละ 40 – 59
- (1) ระดับต้องปรับปรุง หมายถึง ได้คะแนนคิดเป็นร้อยละต่ำกว่า 40

ระดับคุณภาพของผลงาน ☐ ดีมาก ☐ ดี ☐ พอใช้ ☐ ต้องปรับปรุง

เกณฑ์การผ่าน นักเรียนบันทึกผลได้คะแนนอย่างน้อยระดับคุณภาพ 3 (ดี) ขึ้นไป

สรุปผล ☐ ผ่าน ☐ ไม่ผ่าน

ลงชื่อ.....ผู้บันทึกผล วันที่...../...../.....





## บรรณานุกรม

- กระทรวงศึกษาธิการ, สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี. (2553), **คู่มือครูรายวิชาเพิ่มเติม เทคโนโลยีสารสนเทศและการสื่อสาร ภาษาซี กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 4 -6**. กรุงเทพฯ : สกสศ.ลาดพร้าว.
- \_\_\_\_\_. (2553), **หนังสือเรียนรายวิชาเพิ่มเติม เทคโนโลยีสารสนเทศและการสื่อสาร ภาษาซี กลุ่มสาระการเรียนรู้การงานอาชีพและเทคโนโลยี ชั้นมัธยมศึกษาปีที่ 4 -6**. กรุงเทพฯ : สกสศ.ลาดพร้าว.
- ธีรวัฒน์ ประกอบผล. (2553), **คู่มือการเขียนโปรแกรมภาษา C**. กรุงเทพฯ : ชิมพลิฟาย.
- บุญสืบ โพธิ์ศรี และ ชยธร ฉัตรสุวรรณ. (2546), **การเขียนโปรแกรมคอมพิวเตอร์ 1**. นนทบุรี : เจริญรุ่งเรืองการพิมพ์.
- ประภาพร ช่างไม้. (2549), **ฝึกเขียนโปรแกรมจากรวมโจทย์ข้อสอบภาษา C**. นนทบุรี : ไอดีซีฯ.
- \_\_\_\_\_. (2551), **คู่มือเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น**. นนทบุรี : ไอดีซีฯ.
- อรพิน ประวัตติบริสุทธ์. (2552), **คู่มือเรียนภาษาซีฉบับปรับปรุงใหม่**. พิมพ์ครั้งที่ 10. กรุงเทพฯ : โปรวิชั่น.
- โอภาส เอี่ยมสิริวงศ์. (2552), **การเขียนโปรแกรมด้วยภาษา C (C Programming with C)**. กรุงเทพฯ : ซีเอ็ดบุ๊คยูเคชั่น.



