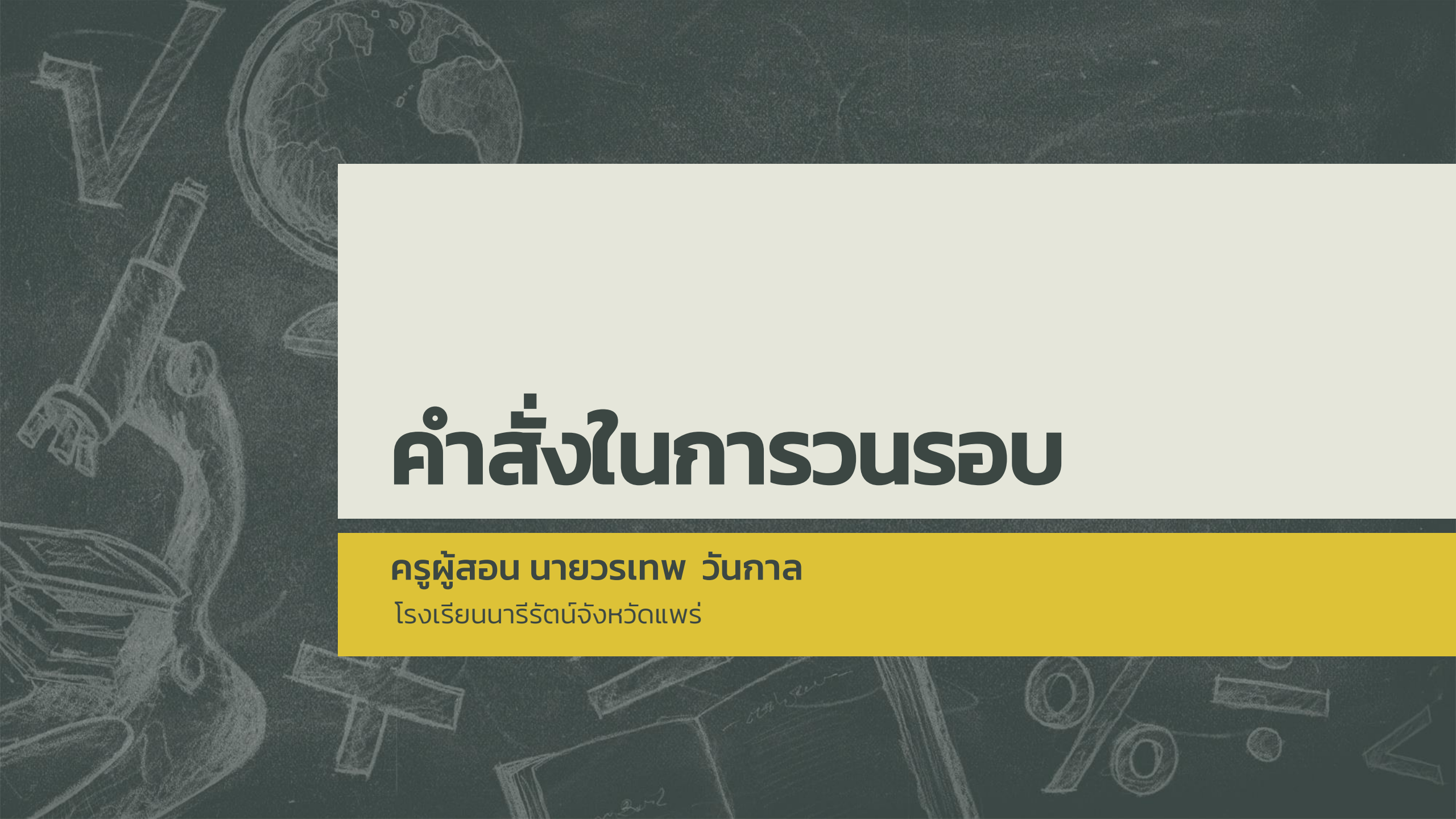




คำสั่งในการวนรอบ

ครูผู้สอน นายวรเทพ วันกาล
โรงเรียนนารีรัตน์จังหวัดแพร่

คำสั่งในการวนรอบ

ในการวนรอบการทำงานในโปรแกรมคอมพิวเตอร์ของแต่ละภาษาที่ถูกเขียนขึ้นมา นั้น จะต้องอาศัยฟังก์ชันในการวนรอบ ซึ่งแต่ละภาษานั้นก็จะมีคำสั่งหรือชุดคำสั่งที่คล้ายกันในภาษาซีนั้นจะใช้คำสั่งในการวนรอบอยู่ 3 คำสั่งคือ

1

คำสั่ง for

2

คำสั่ง while

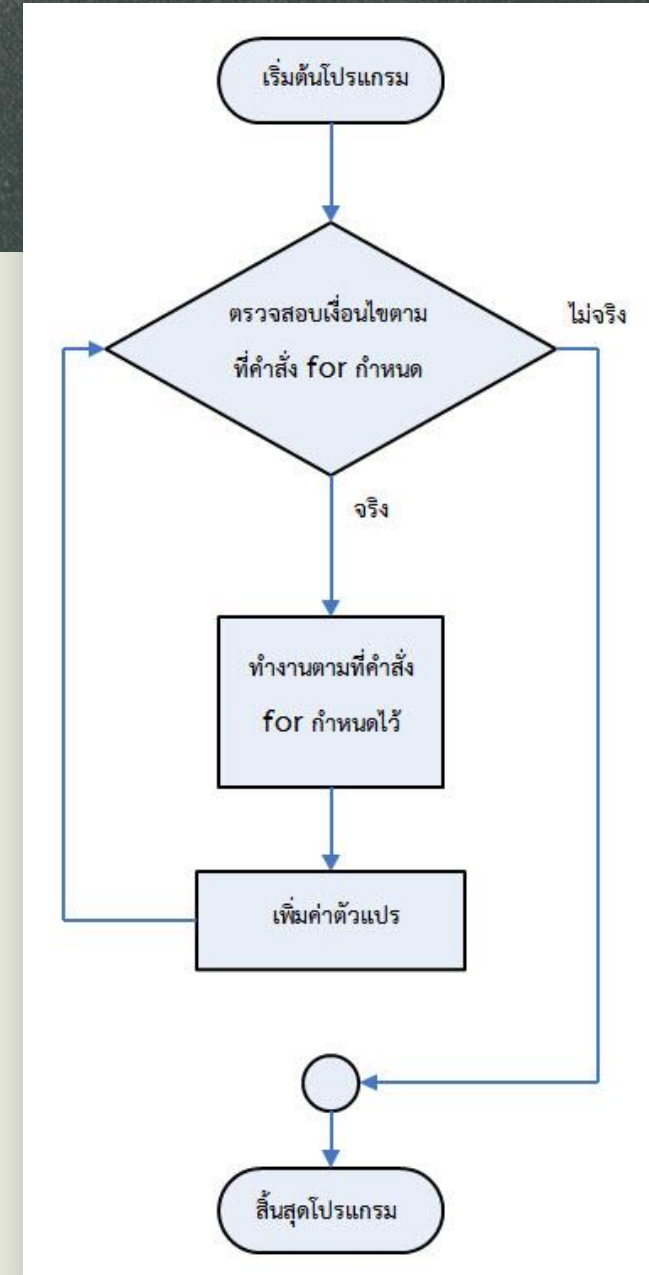
3

คำสั่ง do-while

คำสั่งในการวนรอบ

1. คำสั่ง for

คำสั่ง for เป็นคำสั่งที่ใช้ในการวนรอบ เมื่อต้องการวนรอบตามจำนวนรอบที่ต้องการ เช่น ต้องการวน 10 รอบ เป็นต้น มักจะใช้กับการเขียนโปรแกรมที่รู้จำนวนรอบที่แน่นอน คำสั่ง for มีรูปแบบการทำงานและรูปแบบคำสั่ง ดังภาพ



คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

รูปแบบโดยปกตินั้น เมื่อกำหนดรูปแบบคำสั่ง for แล้วคำสั่งที่จะสามารถทำงานได้จะทำงานเพียงหนึ่งคำสั่งเท่านั้น นั่นก็คือคำสั่งที่เขียนต่อจากคำสั่ง for

```
for (Initial; Condition; Step)  
Statement;
```

```
for (กำหนดค่าเริ่มต้นให้กับตัวแปร; เงื่อนไขที่ต้องการตรวจสอบ; ปรับค่าของตัวแปร) {  
    คำสั่งที่ 1;  
}  
คำสั่งที่ 2;
```

คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

หรือหากต้องการให้การวนรอบหนึ่งรอบสามารถทำงานได้หลายคำสั่งพร้อม ๆ กันในแต่ละรอบสามารถใช้เครื่องหมาย { } เข้ามากำหนดส่วนของการทำงานเกี่ยวกับคำสั่งหรือ statement หลาย ๆ ชุดได้

```
for (Initial; Condition; Step)
{
    Statement1;
    Statement2;
    Statement3;
    StatementN;
}
```

for
Initial
Condition
Step
Statement1-N

คือ เป็นคำสั่งมาตรฐานที่อยู่ในภาษา C เพื่อใช้ในการวนรอบ
คือ ตัวแปรที่ใช้กำหนดค่าเริ่มต้นในการวนรอบ
คือ เงื่อนไขของตัวแปรที่ใช้ในการกำหนดการวนรอบ
คือ ตัวแปรที่ใช้ในการเพิ่มค่าเพื่อการวนรอบในแต่ละรอบ
คือ คำสั่ง หรือชุดคำสั่งที่ใช้ทำงานในแต่ละรอบ

คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

ตัวอย่างโปรแกรมที่ 1



คุณครูต้องการให้นักเรียนแสดงผลลัพท์การคูณกันของแม่สูตรคูณ โดยให้นักเรียนกรอกเลขแม่สูตรคูณลงไป แล้วให้ทำการคูณกันจากเลข 1 จนถึงเลข 12 เช่น นักเรียนกรอกเลขแม่สูตรคูณแม่ 5 จะต้องแสดงผล $5 \times 1 = 5$ ไปจนถึง $5 \times 12 = 60$ เป็นต้น จึงให้นักเรียนเขียนโปรแกรมเพื่อแสดงผลข้อมูลดังนี้

“ให้นักเรียนเขียนโปรแกรมโดยใช้คำสั่ง for แสดงผลลัพท์การคูณกันของแม่สูตรคูณ โดยให้นักเรียนกรอกเลขแม่สูตรคูณลงไป แล้วให้ทำการคูณกันจากเลข 1 จนถึงเลข 12 เช่น นักเรียนกรอกเลขแม่สูตรคูณแม่ 5 จะต้องแสดงผล $5 \times 1 = 5$ ไปจนถึง $5 \times 12 = 60$ เป็นต้น บนจอภาพ”

คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

วิเคราะห์ปัญหา (Analysis)



สิ่งที่โจทย์กำหนด

กำหนดตัวแปร a ที่เก็บค่าแม่สูตรคูณ และ b ที่เก็บค่าจำนวนครั้งในการคูณ โดยใช้ Format Code ชนิด Integer กำหนดการวนรอบด้วยตัวแปร b เริ่มที่ 1 เงื่อนไขคือ b น้อยกว่า เท่ากับ 12 และกำหนดจุดสิ้นสุดของการวนรอบของตัวแปร b

สิ่งที่โจทย์ต้องการ

รับค่าของตัวแปร a ที่เก็บค่าแม่สูตรคูณ แสดงค่าตัวแปร a ตามด้วยเครื่องหมาย $\times$ ตามด้วยตัวแปร b ที่เก็บค่าจำนวนครั้งในการคูณ และผลลัพธ์ตัวแปร a คูณกับ b ในแต่ละรอบ แล้วขึ้นบรรทัดใหม่บนจอภาพ และแสดงข้อความ Thank you บนจอภาพ เมื่อออกจากการวนรอบแล้ว

คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

การแก้ปัญหา
(Problem Solving)

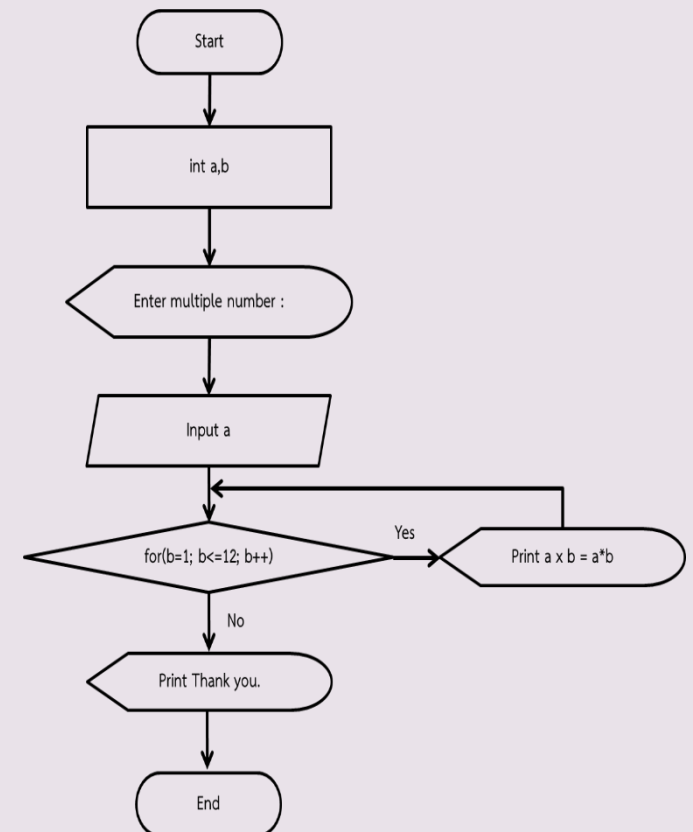


Pseudo Code

Algorithm for

1. Start
2. int a,b
3. Print Enter multiple number :
4. Input a
5. for(b=1; b<=12; b++)
 - 5.1 Print a x b = a*b
6. Print Thank you.
7. End

Flowchart



คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

เขียนโปรแกรม (Coding)

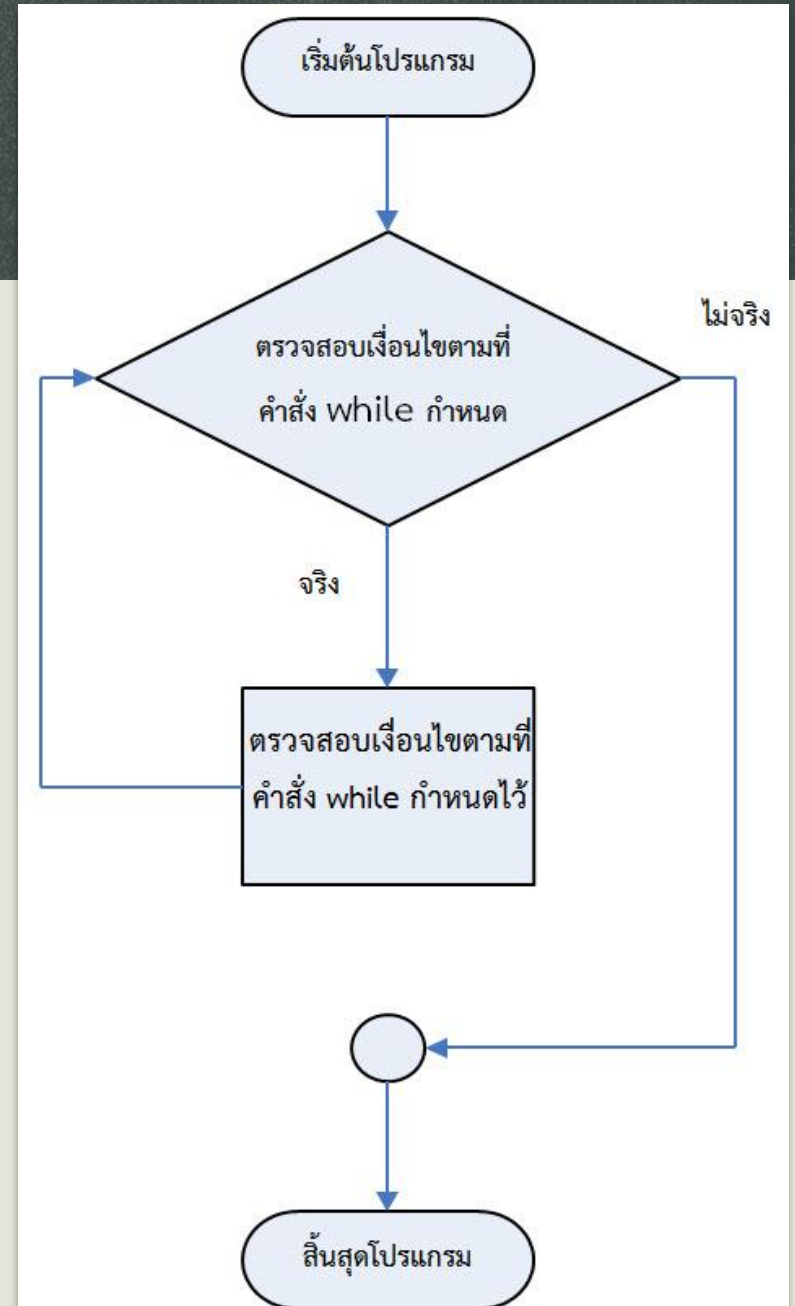


```
#include <stdio.h>
int a,b;
int main( )
{
    printf("Enter multiple number : ");
    scanf("%d",&a);
    for(b=1; b<=12; b++)
    {
        printf("%d x %d = %d \n",a,b,b*a);
    }
    printf("Thank you.");
}
```

คำสั่งในการวนรอบ

2. คำสั่ง while

คำสั่ง while เป็นคำสั่งที่ใช้ควบคุมการวนรอบของโปรแกรม โดยจะมีเงื่อนไข(Condition) เป็นตัวตรวจสอบว่าหากเงื่อนไขที่กำหนดไว้เป็นจริงก็จะทำงานตามคำสั่ง หรือชุดคำสั่งที่กำหนดไว้แล้วก็จะวนไปตรวจสอบเงื่อนไขที่กำหนดไว้อีกครั้ง หากเงื่อนไขยังคงเป็นจริงก็จะทำงานในรอบต่อไปอีก หากเป็นเท็จก็จะออกจากการวนรอบของ while เพื่อทำงานยังคำสั่งต่อไป ดังภาพ



คำสั่งในการวนรอบ

รูปแบบคำสั่ง while

รูปแบบโดยปกตินั้น เมื่อกำหนดรูปแบบคำสั่ง for แล้วคำสั่งที่จะสามารถทำงานได้จะทำงานเพียงหนึ่งคำสั่งเท่านั้น นั่นก็คือคำสั่งที่เขียนต่อจากคำสั่ง for

```
while (Condition)  
Statement;
```

```
while (เงื่อนไข) {  
    คำสั่งที่ 1;  
}  
คำสั่งที่ 2;
```

คำสั่งในการวนรอบ

รูปแบบคำสั่ง for

หรือหากต้องการให้การวนรอบแต่ละครั้งสามารถทำงานได้หลายชุดคำสั่ง ก็จะนำเอาเครื่องหมาย { } มาเขียนไว้เพื่อเป็นการกำหนดขอบเขตของการทำงานทั้งหมด

```
while (Condition)
{
    Statement1;
    Statement2;
    Statement3;
    StatementN;
}
```

while	คือ คำสั่งที่ใช้ในภาษาซี เป็นคำสั่งสงวน
Condition	คือ เงื่อนไขที่ต้องให้มีการตรวจสอบก่อนการทำงานในการวนรอบของ while
Statement	คือ คำสั่ง หรือชุดคำสั่งที่ใช้ทำงานในแต่ละรอบ

คำสั่งในการวนรอบ

คำสั่ง while

ตัวอย่างโปรแกรมที่ 1



คุณครูต้องการให้นักเรียนแสดงผลเฉพาะตัวเลขที่อยู่ระหว่าง 1 ถึง 20 บนจอภาพ จึงให้นักเรียนเขียนโปรแกรมเพื่อแสดงผลข้อมูล ดังนี้

“ให้นักเรียนเขียนโปรแกรมโดยแสดงเลขที่ทุกจำนวนที่อยู่ระหว่าง 1 ถึง 20 ออกมาบนจอภาพ โปรแกรมนี้จะแสดงให้เห็นถึงการตรวจสอบเงื่อนไขการวนรอบการทำงานของคำสั่ง while”

คำสั่งในการวนรอบ

คำสั่ง while

วิเคราะห์ปัญหา (Analysis)



สิ่งที่โจทย์กำหนด

กำหนดตัวแปร $a=1$ โดยใช้ Format Code ชนิด Integer กำหนดการวนรอบด้วยคำสั่ง while โดยมีเงื่อนไขว่า ตัวแปร a น้อยกว่า หรือเท่ากับ 20 และกำหนดค่าใหม่ ให้ตัวแปร a โดยที่บวกค่าเพิ่มขึ้นอีก 2

สิ่งที่โจทย์ต้องการ

แสดงตัวเลขคี่ที่อยู่ระหว่าง 1 ถึง 20 บนจอภาพ

คำสั่งในการวนรอบ

คำสั่ง while

การแก้ปัญหา
(Problem Solving)

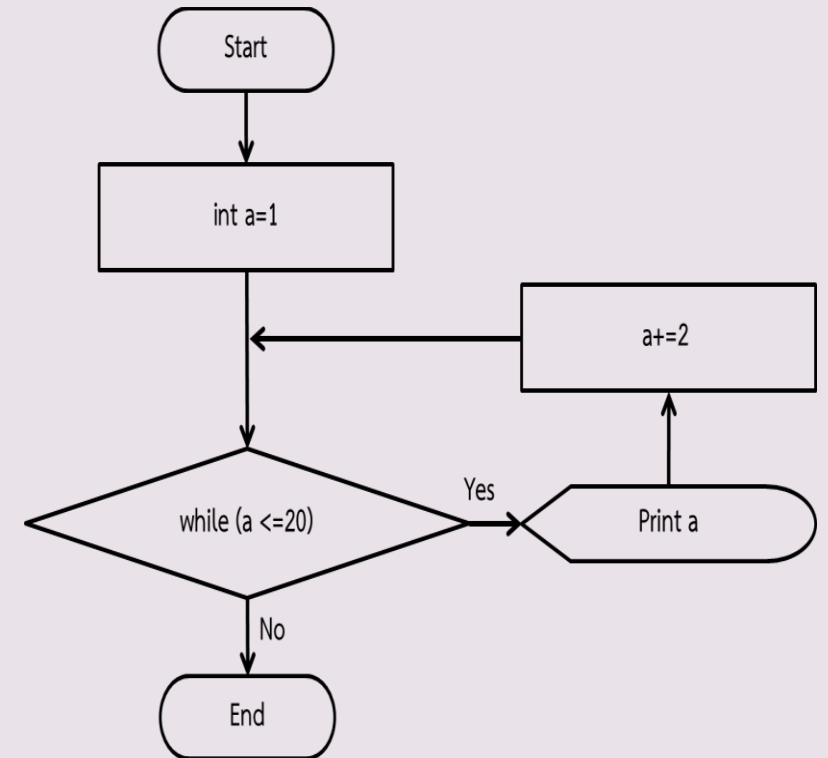


Pseudo Code

Algorithm while

1. Start
2. int a=1
3. while (a <=20)
 - 3.1 Print a
 - 3.2 a+=2
4. End

Flowchart



คำสั่งในการวนรอบ

คำสั่ง while

เขียนโปรแกรม (Coding)

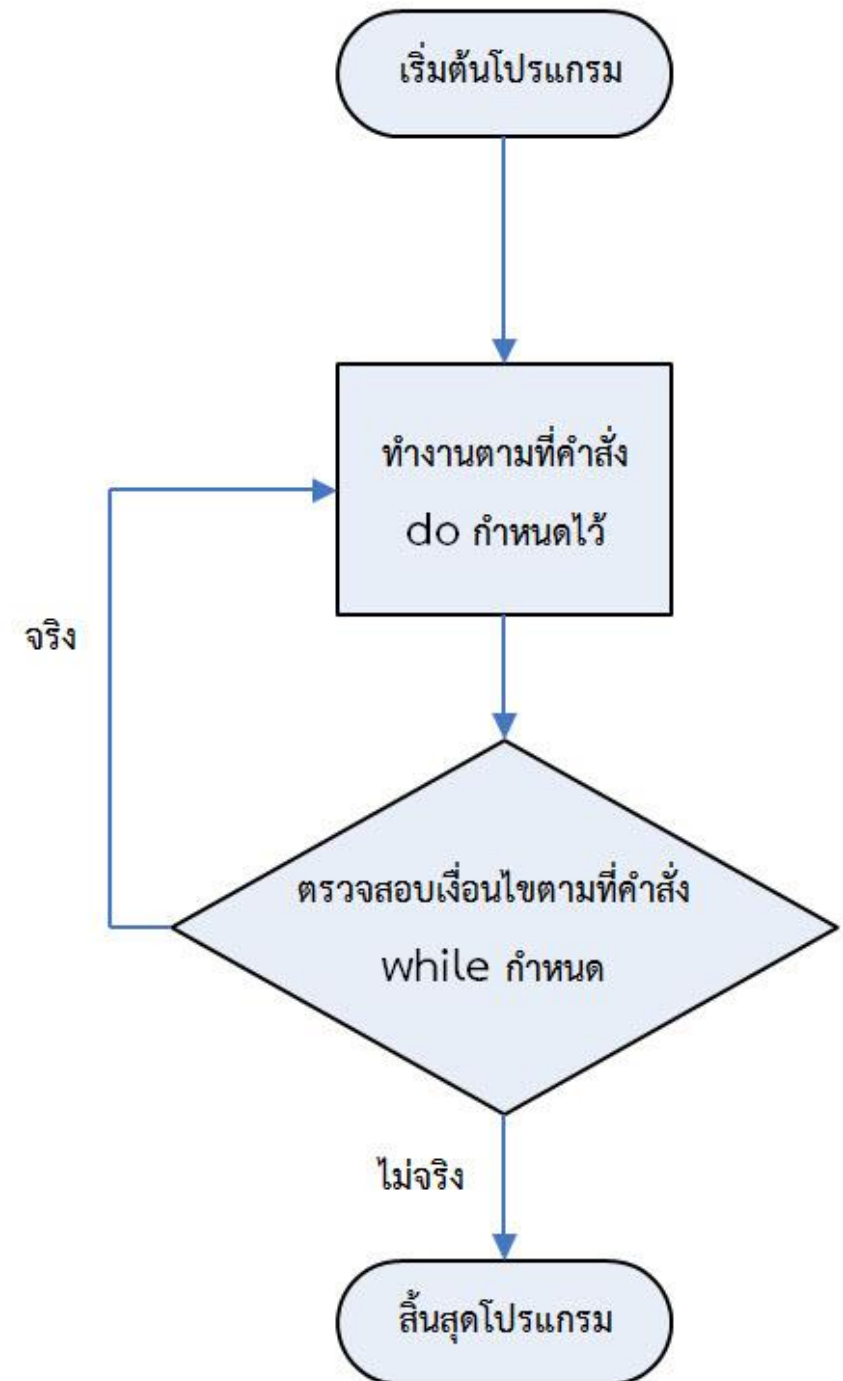


```
#include <stdio.h>
int a;
int main( )
{
    a = 1;
    while (a <=20)
    {
        printf("%d ",a);
        a+=2;
    }
}
```

คำสั่งในการวนรอบ

3. คำสั่ง do-while

คำสั่ง do-while นั้นมีการทำงานคล้ายกับคำสั่ง while ซึ่งจะมีการตรวจสอบเงื่อนไขการทำงาน หากเป็นจริงก็จะทำงานตามคำสั่ง หรือชุดคำสั่งที่ได้กำหนดไว้ แต่คำสั่ง do-while นั้นจะทำงานก่อน 1 รอบ แล้วจึงมีการตรวจสอบเงื่อนไข หากเป็นจริงก็จะทำงานต่อในรอบต่อไป หากเป็นเท็จก็จะออกจากการทำงานในรอบการทำงานนั้น แต่อย่างไรก็เกิดการทำงานแล้ว 1 รอบ ดังภาพ



คำสั่งในการวนรอบ

รูปแบบคำสั่ง do-while

รูปแบบโดยปกตินั้น เมื่อกำหนดรูปแบบคำสั่ง for แล้วคำสั่งที่จะสามารถทำงานได้จะทำงานเพียงหนึ่งคำสั่งเท่านั้น นั่นก็คือคำสั่งที่เขียนต่อจากคำสั่ง for

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

```
do {  
    คำสั่งที่ 1;  
} while (เงื่อนไข);  
คำสั่งที่ 2;
```

คำสั่งในการวนรอบ

รูปแบบคำสั่ง do-while

หรือหากต้องการให้แต่ละการวนรอบนั้น ทำงานมากกว่าหนึ่งคำสั่งก็สามารถทำได้ โดยการใช้อเครื่องหมาย { } รอบชุดคำสั่ง (statement)

```
do
{
    statement1;
    statement2;
    statement3;
    statementN;
}
while (condition);
```

do	คือ เริ่มต้นคำสั่ง do-while ซึ่งจะตามด้วย 1 ชุดคำสั่งที่ต้องการทำงาน แต่หากต้องการให้ทำงานมากกว่า 1 ชุดคำสั่งจะต้องใช้เครื่องหมาย { } เข้ามาช่วย
while	คือ คำสั่งที่จะต้องเขียนไว้ แล้วจะต้องตามด้วยเงื่อนไขที่ต้องการตรวจสอบ
condition	คือ เงื่อนไขที่ต้องการตรวจสอบ

คำสั่งในการวนรอบ

3. คำสั่ง do-while

ตัวอย่างโปรแกรมที่ 1



คุณครูต้องการให้นักเรียนแสดงค่าตัวเลขที่น้อยกว่าหรือเท่ากับ 10 แต่ถ้าตัวเลขมากกว่า 10 ให้แสดงข้อความ End program บนจอภาพ จึงให้นักเรียนเขียนโปรแกรมเพื่อแสดงผลข้อมูล ดังนี้

“ให้นักเรียนเขียนโปรแกรมโดยใช้คำสั่ง do-while เริ่มการวนรอบด้วยคำสั่ง do แสดงค่าของ ตัวแปร a ชนิด Integer แล้วขึ้นบรรทัดใหม่ และหลังจากคำสั่ง while คือ a จะต้องน้อยกว่าหรือเท่ากับ 10 ถ้าผลลัพธ์ออกมาเป็นจริงก็จะกลับไปทำงานตามที่คำสั่ง do กำหนดไว้ แต่หากเป็นเท็จก็จะออกจากการทำงานวนรอบนี้ และจะแสดงข้อความ End program บนจอภาพ”

คำสั่งในการวนรอบ

3. คำสั่ง do-while

วิเคราะห์ปัญหา (Analysis)



สิ่งที่โจทย์กำหนด

กำหนดตัวแปร $a=1$ โดยใช้ Format Code ชนิด Integer กำหนดค่าตัวแปร a เท่ากับ 11 เมื่อเริ่มการวนรอบด้วยคำสั่ง do และกำหนดเงื่อนไขหลังคำสั่ง while คือ a จะต้องน้อยกว่าหรือเท่ากับ 10 ถ้าผลลัพธ์ออกมาเป็นจริงก็จะกลับไปทำงานตามที่คำสั่ง do กำหนดไว้ แต่หากเป็นเท็จก็จะออกจากการทำงานวนรอบนี้ ไปทำงานยังคำสั่งต่อไปของโปรแกรม

สิ่งที่โจทย์ต้องการ

แสดงค่าของตัวแปร a และแสดงข้อความ End program บนจอภาพ

คำสั่งในการวนรอบ

3. คำสั่ง do-while

การแก้ปัญหา
(Problem Solving)

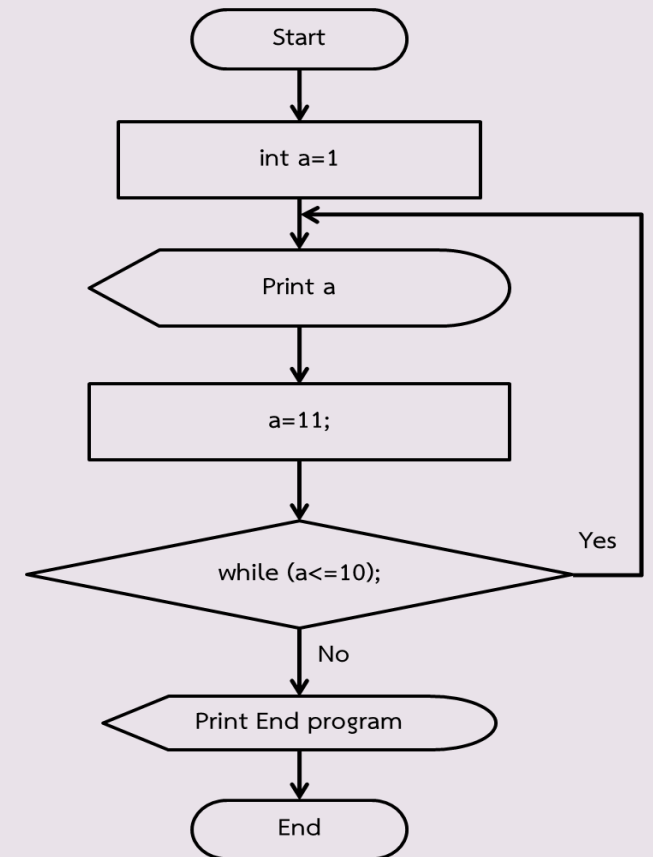


Pseudo Code

Algorithm do-while

1. Start
2. int a=1
3. do
 - 3.1 Print a
 - 3.2 a=11;
4. while (a<=10)
 - 4.1. Print End program
5. End

Flowchart



คำสั่งในการวนรอบ

3. คำสั่ง do-while

เขียนโปรแกรม (Coding)



```
#include <stdio.h>
main( )
{
    int a;
    a=1;
    do
    {
        a=a+1;
        printf("%d \n",a);
    }
    while (a<=10);
    printf("End program");
}
```