

Programming Fundamental

Pseudo Code

นำเข้าข้อมูล

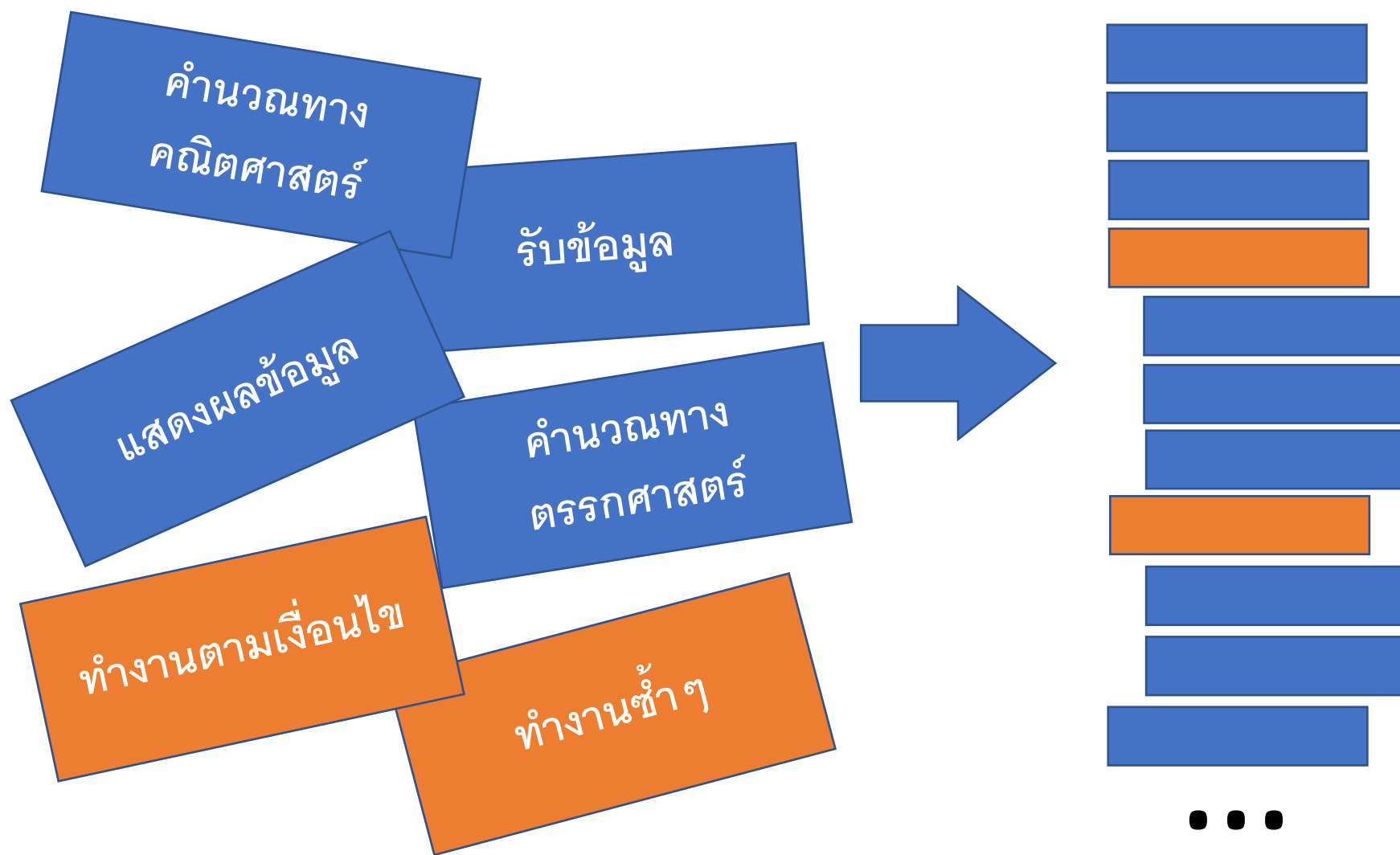
ส่งออกข้อมูล

คำนวณทาง
คณิตศาสตร์

คำนวณทาง
ตรรกศาสตร์

ทำงานตามเงื่อนไข

ทำงานซ้ำ



กลไกสำคัญของการเขียนโปรแกรม คือการใช้ “ตัวแปร” ในการ

- เก็บข้อมูล
- แสดงผลข้อมูล
- เปรียบเทียบทางตรรกศาสตร์
- คำนวณและเก็บผลลัพธ์ของการคำนวณ

แล้วใช้ “คำสั่งควบคุมลำดับการทำงาน” ได้แก่

- การทำงานซ้ำ
- การทำงานแบบมีเงื่อนไข

เป็นตัวควบคุมการทำงาน

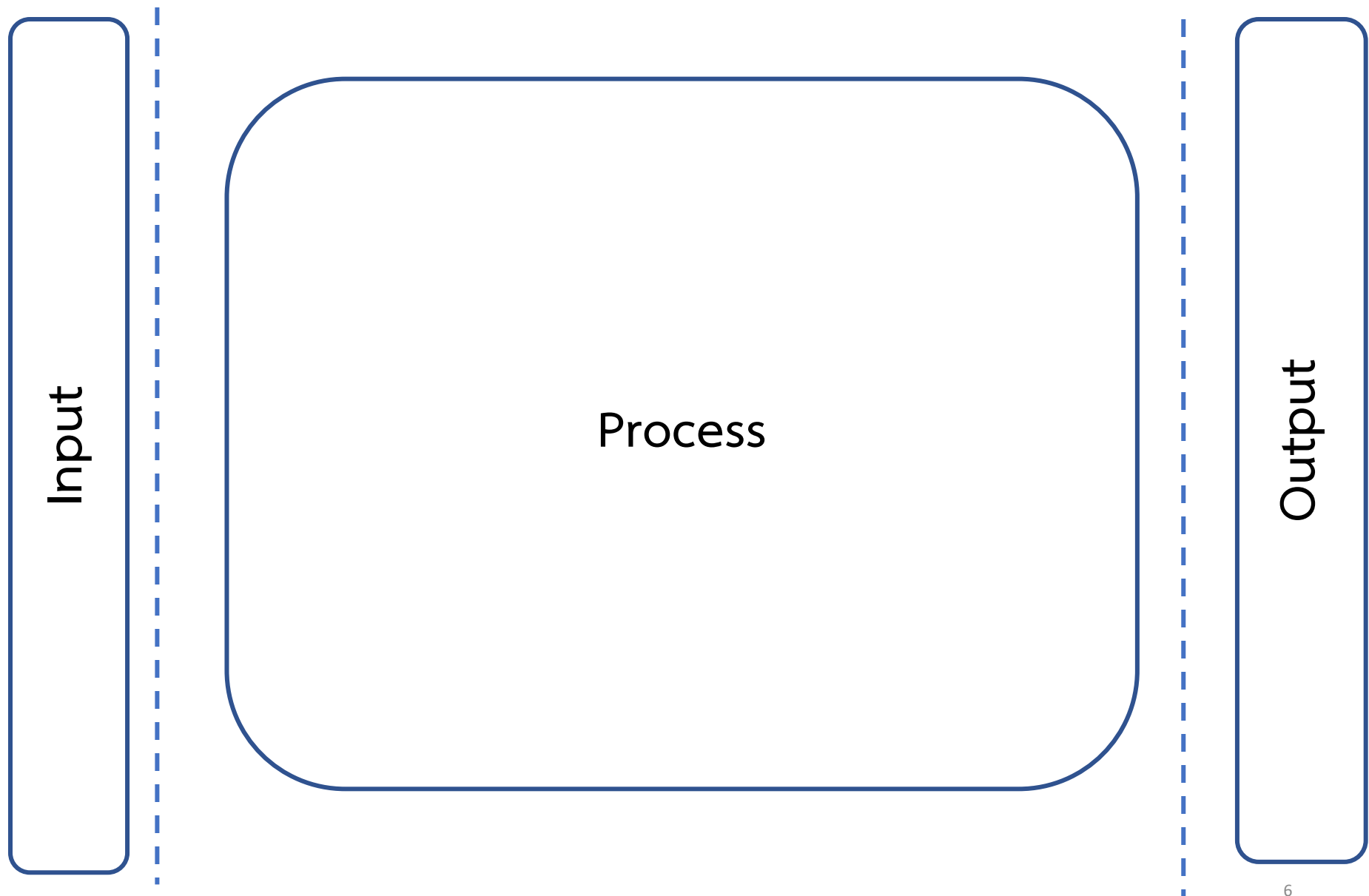
การทำงานพื้นฐานของคอมพิวเตอร์

- รับข้อมูลเก็บไว้ในตัวแปร X
- นำค่าจากตัวแปร X ไปแสดงผล
- คำนวณค่า/กำหนดค่า X ตามสูตร.....

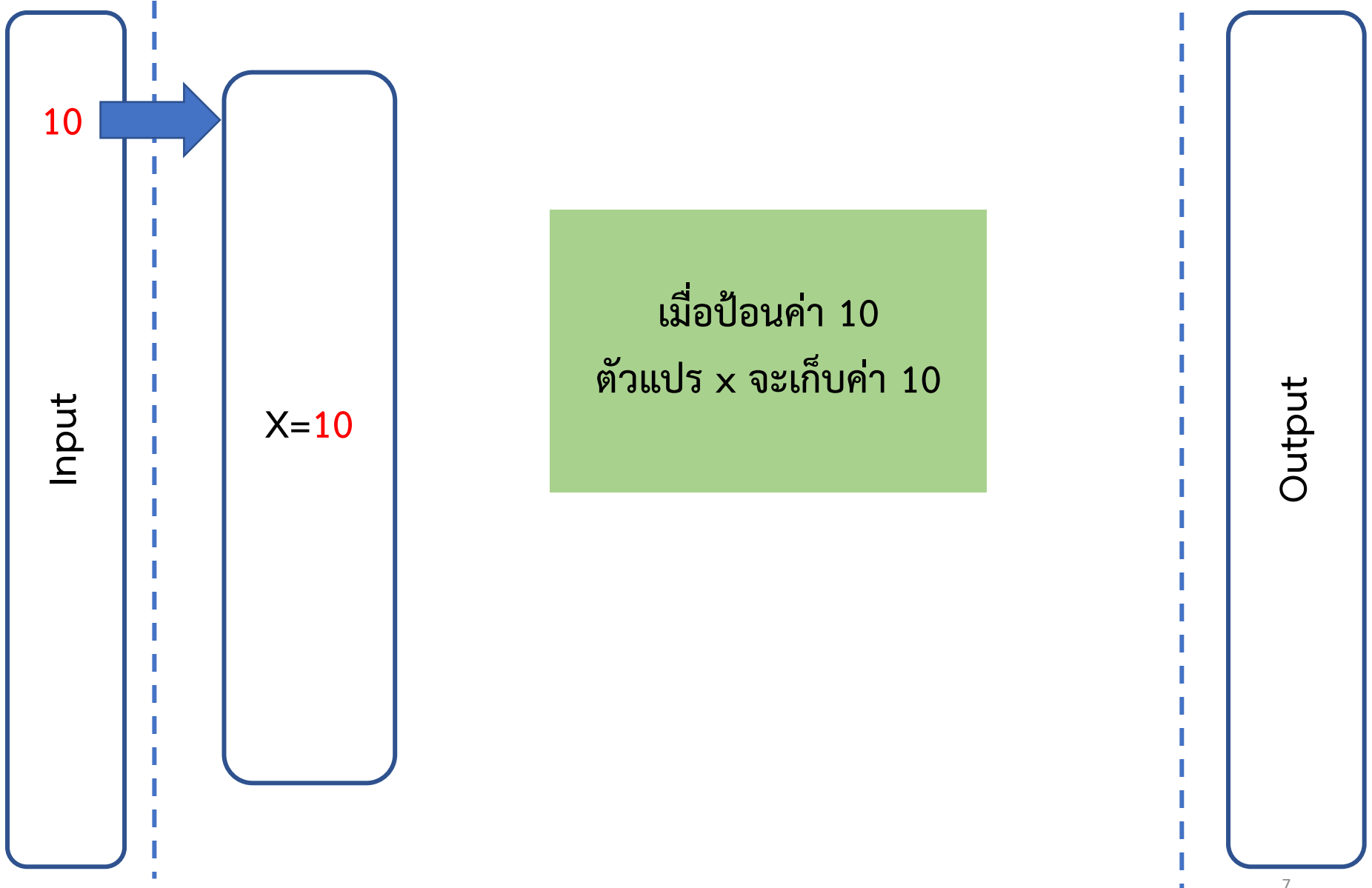
โค้ดเทียม
(pseudo code)

- เมื่อ $f(x)$ เป็นจริง ให้ทำงาน..... ถ้าเป็นเท็จ ให้ทำงาน
- เมื่อ $f(x)$ เป็นจริง ให้ทำงาน ซ้ำไปเรื่อยๆ

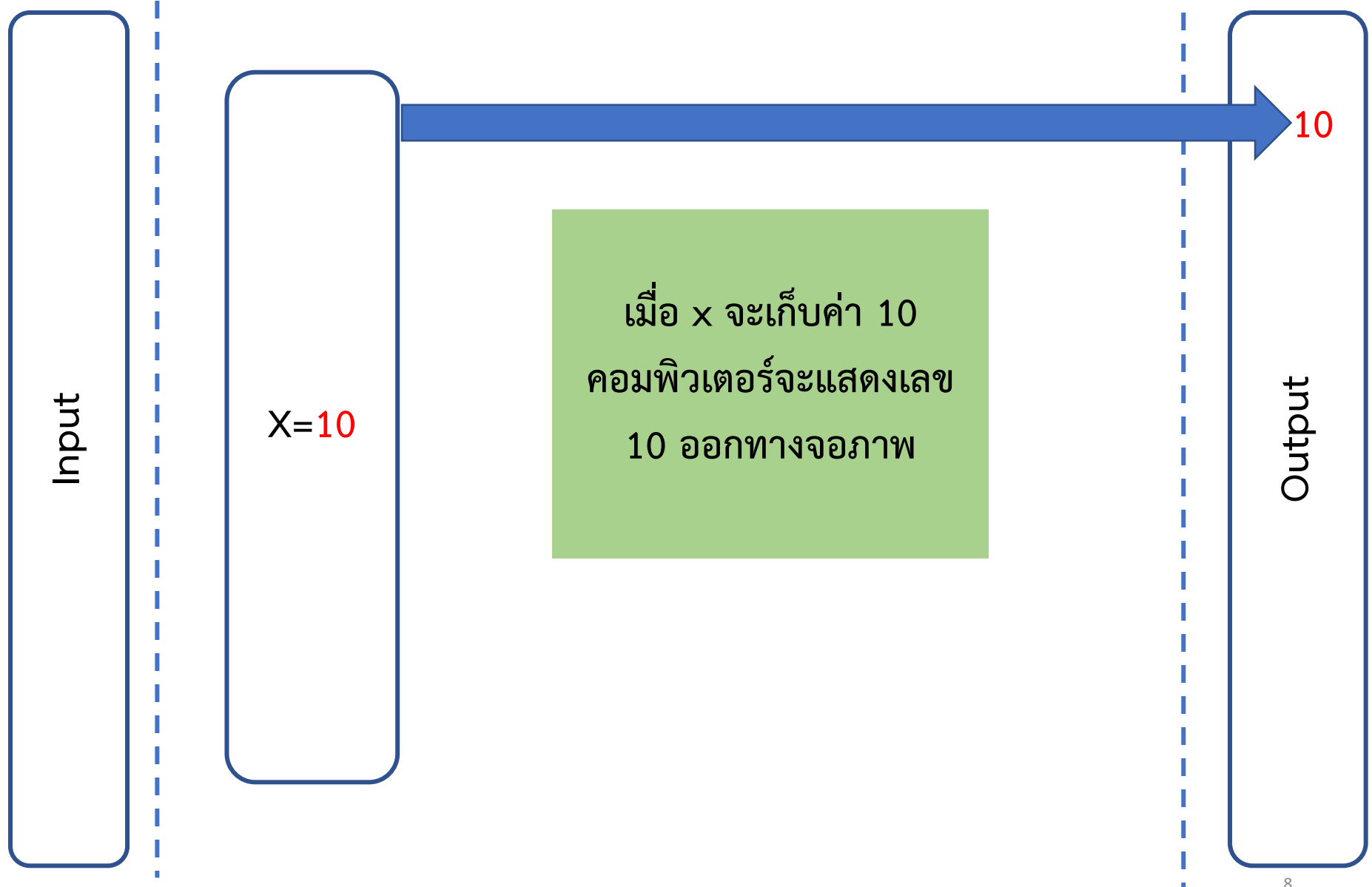
องค์ประกอบ IPO ของโปรแกรม



Pseudo Code : รับข้อมูล เก็บไว้ในตัวแปร x



Pseudo Code : แสดงผลข้อมูลในตัวแปร x



Pseudo Code : คำนวณค่า x ตามสูตร...

Input

X

10



20

X=20

หรือ

X=X+10

หรือ

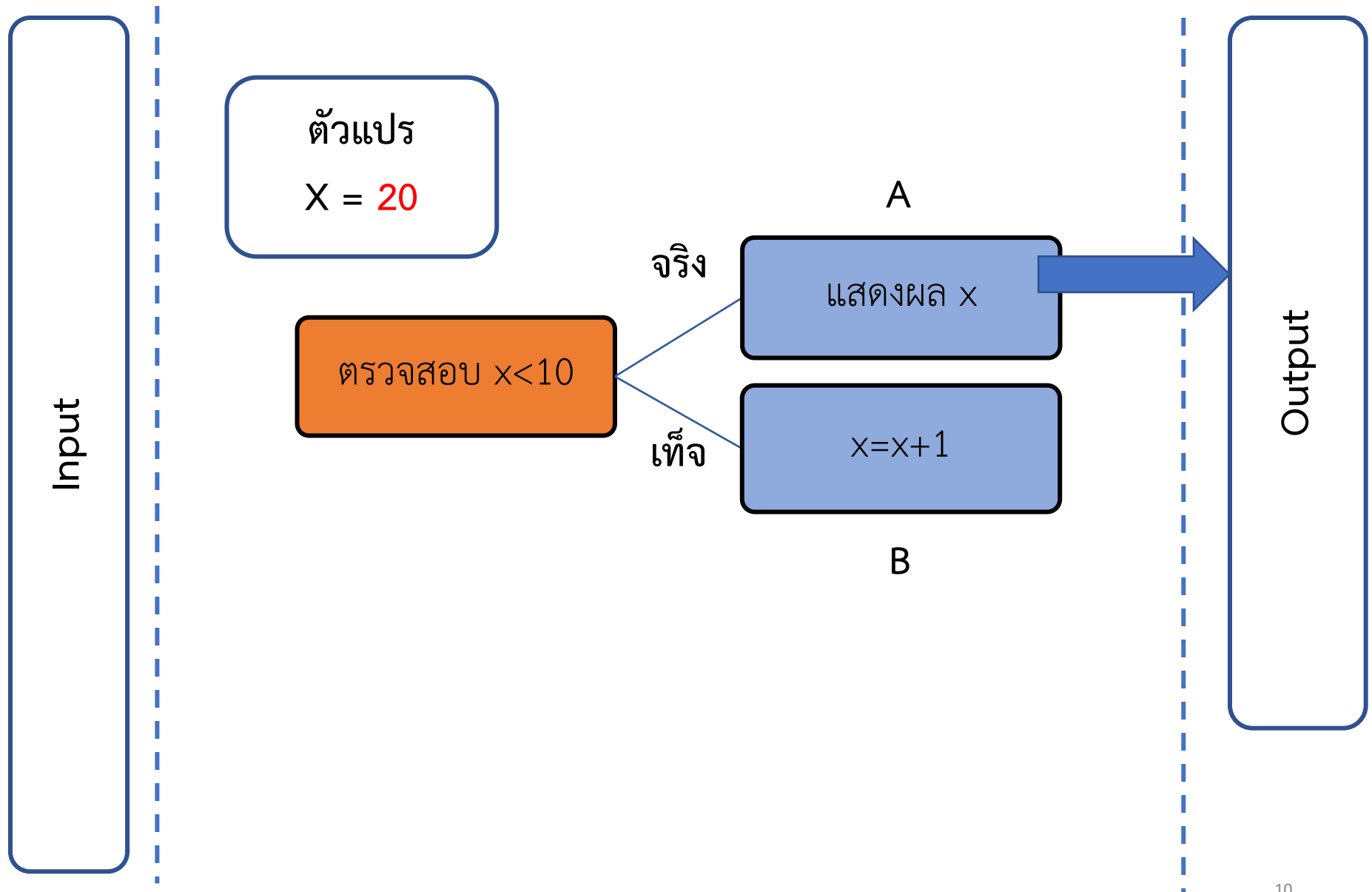
X=2*X

หรือ

X=X+X

Output

Pseudo Code : ถ้า $x < 10$ เป็นจริง ให้ทำ [A] ถ้าไม่ใช่ให้ทำ [B]

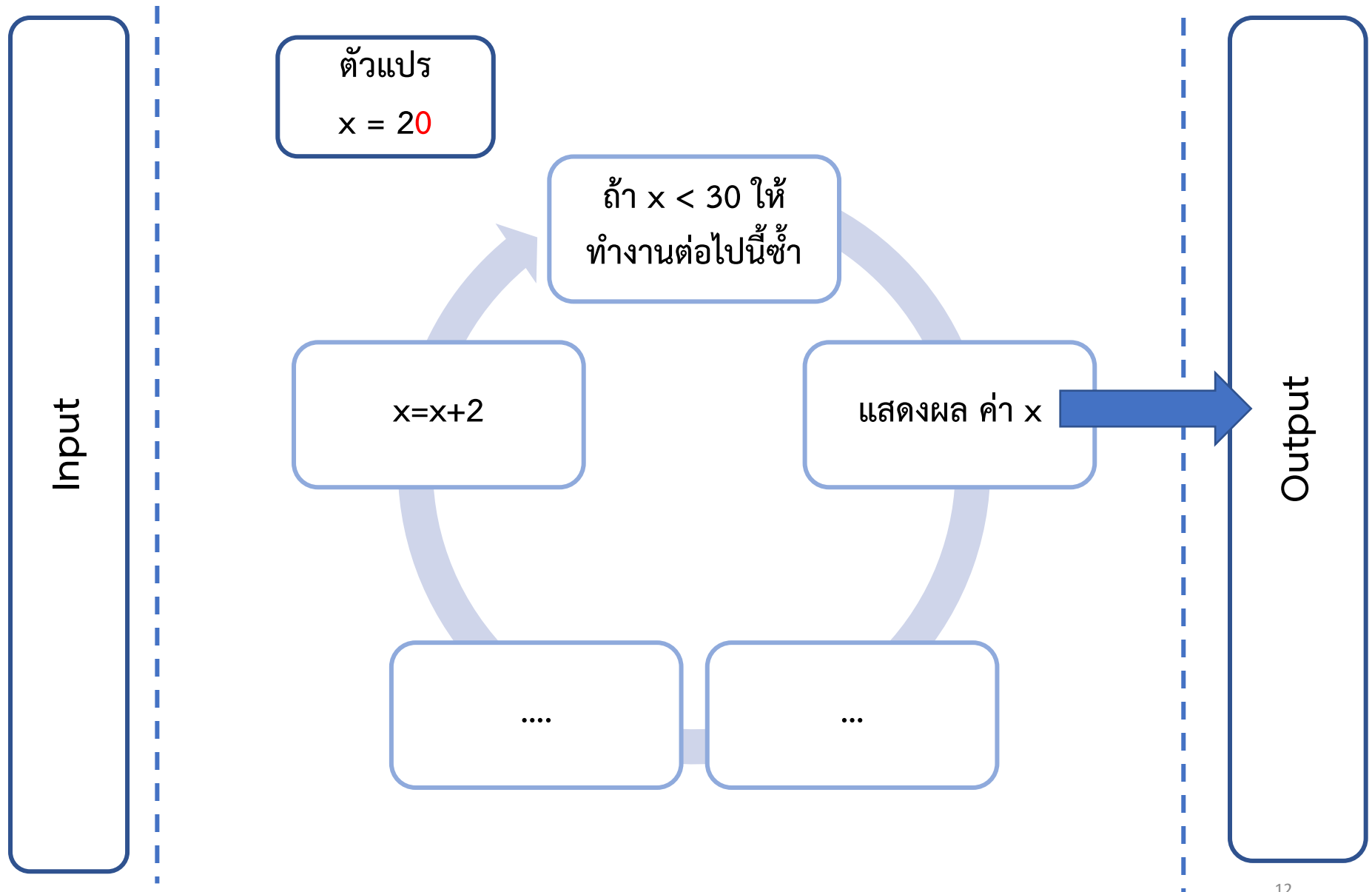


Activity : Pseudo Code นี้ให้ผลลัพธ์การทำงานคืออะไร

1. รับตัวเลข 2 ตัวเก็บไว้ในตัวแปร X,Y
2. ถ้า $x > y$
 1. เป็นจริงให้ แสดงผล x
 2. ถ้าเป็นเท็จ แสดงผล y

- a) เมื่อป้อนอินพุต 20,30 จะได้ผลลัพธ์คืออะไร ทำงานบรรทัดไหนบ้าง
- b) เมื่อป้อนอินพุต 5,28 จะได้ผลลัพธ์คืออะไร ทำงานบรรทัดไหนบ้าง
- c) เมื่อป้อนอินพุต 9,9 จะได้ผลลัพธ์คืออะไร ทำงานบรรทัดไหนบ้าง

Pseudo Code : เมื่อ $x < 30$ ให้ทำงานคำสั่ง ... ซ้ำ



Activity : Pseudo Code นี้ให้ผลลัพธ์การทำงานคืออะไร

1. กำหนดค่า $n = 1$
2. ทำงานต่อไปนี้เมื่อ $n < 5$
 1. แสดงผลค่า n
 2. $n = n + 1$

ผลลัพธ์การทำงานคืออะไร
มีการทำงานที่บรรทัดใดบ้าง ตามลำดับ

Activity : Pseudo Code นี้ให้ผลลัพธ์การทำงานคืออะไร

1. รับอินพุตเป็นตัวเลข 2 ตัว เก็บใน a,b
2. ให้ $x=a$
3. ทำงานซ้ำ ถ้า $x \leq b$
 1. ถ้า x เป็นเลขคี่
 1. ให้แสดงผลค่า x
 2. $X=x+1$

เมื่อป้อนอินพุตเป็น 1,10
ผลลัพธ์การทำงานคืออะไร
มีการทำงานที่บรรทัดใดบ้าง ตามลำดับ

Activity : Pseudo Code นี้ให้ผลลัพธ์การทำงานคืออะไร

1. รับตัวเลข 1 ตัวเก็บไว้ในตัวแปร m
2. ทำซ้ำเมื่อ $m > 0$
 1. แสดงผล *
 2. $m = m - 1$

เมื่อป้อนอินพุต 10 จะได้ผลลัพธ์คืออะไร

Activity : Pseudo Code นี้ให้ผลลัพธ์การทำงานคืออะไร

1. รับตัวเลขเก็บไว้ในตัวแปร n
2. $x=1$
3. ทำงานต่อไปนี้ซ้ำ เมื่อ $x \leq n$
 1. $y=1$
 2. ทำงานต่อไปนี้ซ้ำ เมื่อ $y \leq n$
 1. แสดงผล *
 2. $y=y+1$
 3. แสดงผล new line
 4. $x=x+1$

เมื่อป้อนอินพุตเป็น 4
ผลลัพธ์การทำงานคืออะไร

มีการทำงานตามลำดับที่บรรทัดใดบ้าง

เมื่อหยุดการทำงานของโปรแกรม
ที่ค่า $x=3$, $y=2$
ผลลัพธ์การทำงานคืออะไร

Activity : ให้นักศึกษาเขียน Pseudo Code ให้มี input/output ดังนี้

Input : 1

Output

*

Input : 2

Output

*

**

Input : 3

Output

*

**

Input : 4

Output

*

**

Input : 5

Output

*

**

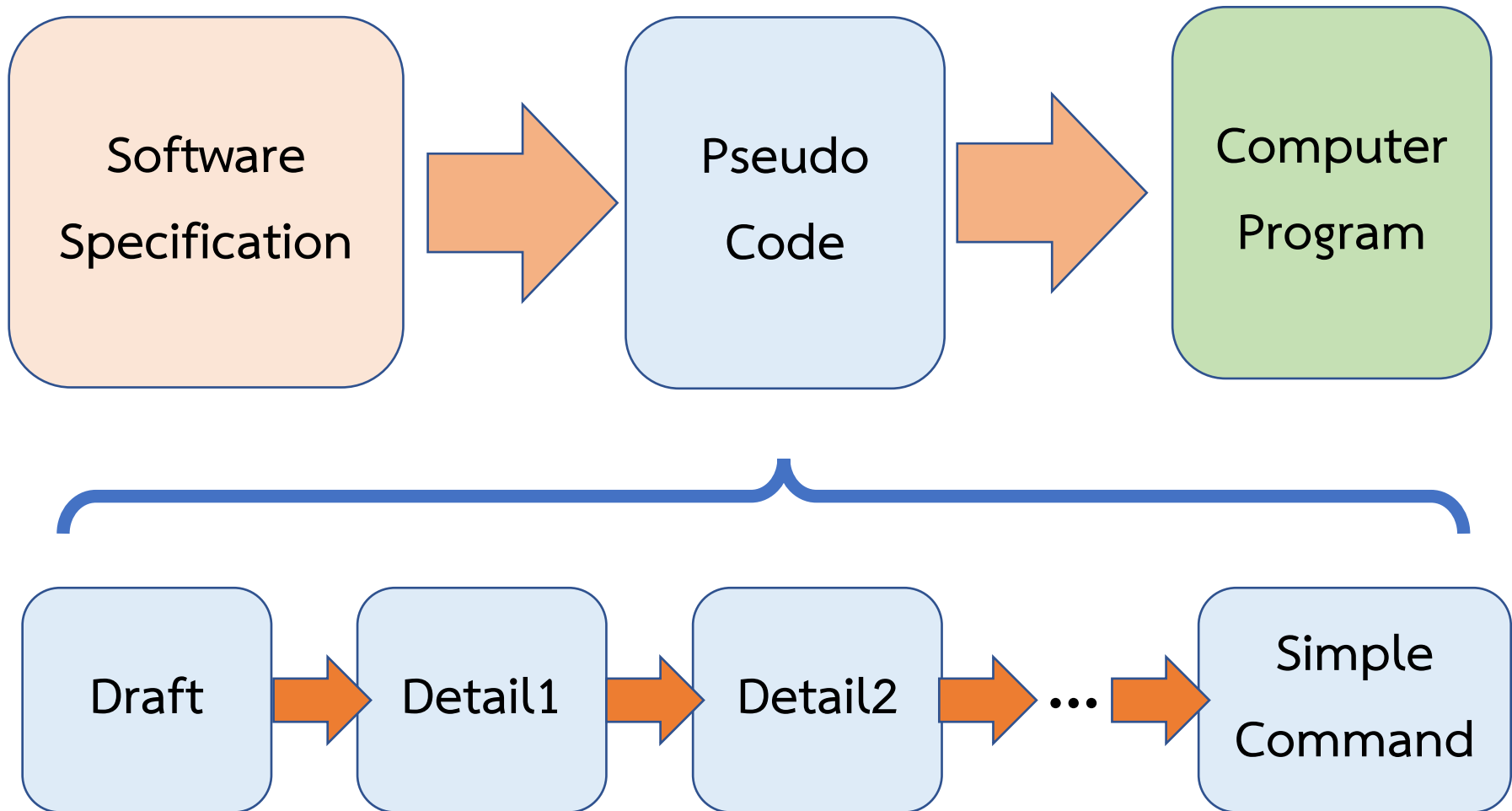
Input : 6

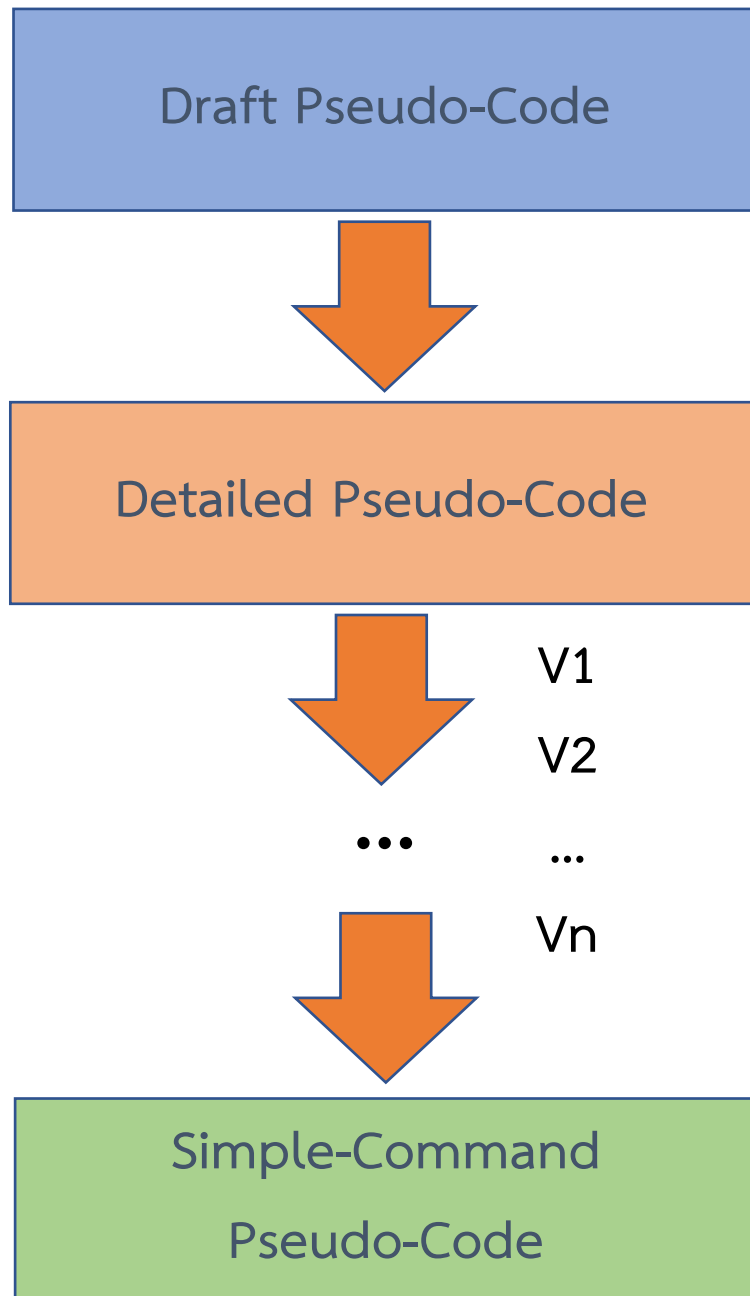
Output

*

**

Pseudo Code
in
Top-Down Approach





Pseudo Code ที่มีรายละเอียดการทำงาน
ใน**ภาพรวมครบ** อาจมีการทำงานที่ไม่ใช่
การทำงานพื้นฐานของคอมพิวเตอร์

เพิ่ม / เปลี่ยน วิธีทำงาน โดยใช้**นิยาม**
หรือการเทียบเคียงการทำงานอื่นๆ
ให้มีการทำงานพื้นฐานมากขึ้น

Pseudo Code ที่มีแต่การทำงานพื้นฐาน
ของ**คอมพิวเตอร์**

#1 : จากโจทย์จะเริ่มกำหนด Draft Pseudo-Code ที่อธิบายการทำงานในภาพรวมอย่างคร่าวๆ โดยยังไม่ต้องสนใจรายละเอียด

เขียนการทำงานเพื่อตรวจสอบว่าตัวเลขที่ป้อนเป็น
ตัวเลขจำนวนเฉพาะหรือไม่
ถ้าเป็นจำนวนเฉพาะ แสดงผล “Prime Number”
ถ้าไม่ใช่จำนวนเฉพาะ แสดงผล “Not Prime Number”

Draft Pseudo-Code

รับตัวเลขจำนวนเต็ม เก็บใน x
ถ้า x เป็น**จำนวนเฉพาะ** ให้แสดงผล “Prime Number”
ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”

#2 : Draft Pseudo-Code อาจมีมากกว่า 1

เลือกแนวทางที่ ???

- ทำงานได้ถูกต้องแน่นอน
- ทำงานเร็ว
- เขียนง่าย
- Code สั้น

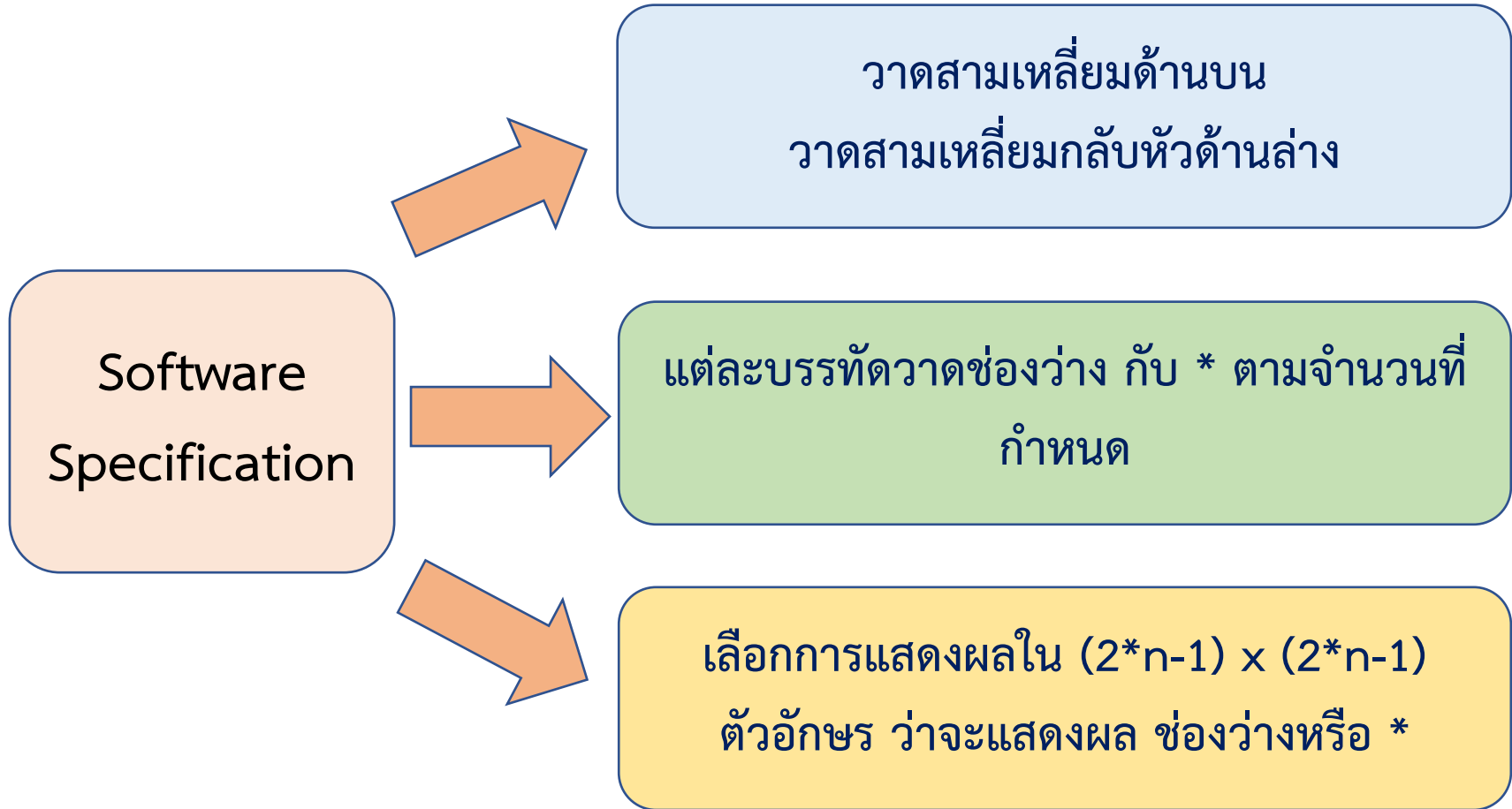
3

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

4

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


Draft Pseudo-code



เลือกอันไหนดี ??

...

#3 : เพิ่มรายละเอียดใน Draft Pseudo-Code ส่วนที่ไม่ใช่การทำงานพื้นฐาน ให้มี
การทำงานพื้นฐานมากขึ้นเป็น Detailed Pseudo-Code

อาจมีทางเลือกหลายทาง ให้เลือกทางที่เหมาะสมที่สุด

เพิ่มรายละเอียดโดยการ อธิบายตาม
นิยามที่เกี่ยวข้อง หรือการเทียบเคียง
การทำงานอื่น
เพื่อให้มีการทำงานพื้นฐานมากขึ้น

เลือกแนวทางที่ ???

- ทำงานได้ถูกต้องแน่นอน
- ทำงานเร็ว
- เขียนง่าย
- Code สั้น

Draft Pseudo-Code

รับตัวเลขจำนวนเต็ม เก็บใน x
ถ้า x เป็นจำนวนเฉพาะ ให้แสดงผล “Prime Number”
ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”

n เป็นจำนวนเฉพาะเมื่อไหร่ ??

Detailed !!!

A

ในตัวเลข $1..n$
มี 2 ตัวเท่านั้นที่ n หารได้ลงตัว

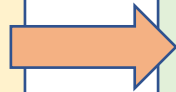
B

ในตัวเลข $2..n$
ตัวเลขที่ n หารได้ลงตัวเป็นตัวแรกคือ n

Detailed Pseudo-Code

A

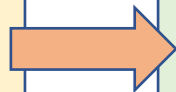
ในตัวเลข 1..n
มี 2 ตัวเท่านั้นที่หาร
ลงตัว



รับตัวเลขจำนวนเต็ม เก็บใน x
หาจำนวนตัวเลขใน 1..x ที่นำมาหาร x ลงตัว เก็บใน a
ถ้า a เป็น 2 ให้แสดงผล “Prime Number”
ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”

B

ในตัวเลข 2..n
ตัวเลขที่หารได้ลงตัวเป็น
ตัวแรกคือ n



รับตัวเลขจำนวนเต็ม เก็บใน x
หาตัวเลขใน 2..x ตัวแรกที่หาร x ลงตัว เก็บใน a
ถ้า a = x ให้แสดงผล “Prime Number”
ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”

#4 : ถ้า Detailed Pseudo Code ในข้อ #3 ยังมีส่วนที่ไม่เป็นการทำงาน
พื้นฐาน ให้ทำข้อ #3 ซ้ำไปเรื่อยๆ

เป็นการทำงานพื้นฐานทั้งหมดหรือยัง ??

Detailed Pseudo-Code

A

หาจำนวนตัวเลขใน $1..x$ ที่
นำมารวม x ลงตัว เก็บใน a

ให้ $a=0$

กำหนดตัวนับ $c = 1$

ทำซ้ำถ้า $c \leq x$:

ถ้า x หารด้วย c ลงตัว ให้เพิ่มค่า a
เพิ่มค่า c

B

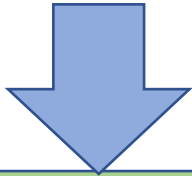
หาตัวเลขใน $2..x$ ตัวแรกที่
หาร x ลงตัว เก็บใน a

กำหนดตัวนับ $a = 2$

ทำซ้ำถ้า x หารด้วย a ไม่ลงตัว:
เพิ่มค่า a

หารแล้วเศษไม่เป็น 0

Detailed Pseudo-Code



Simple-Command
Pseudo-Code

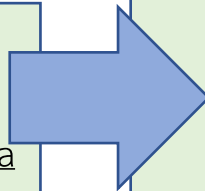
A

รับตัวเลขจำนวนเต็ม เก็บใน x

หาจำนวนตัวเลขใน 1..x ที่นำมาหาร x ลงตัว เก็บใน a

ถ้า a เป็น 2 ให้แสดงผล “Prime Number”

ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”



รับตัวเลขจำนวนเต็ม เก็บใน x

ให้ a=0

กำหนดตัวนับ c = 1

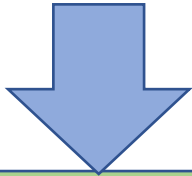
ทำซ้ำถ้า c ≤ x:

ถ้า x หารด้วย c ลงตัว ให้เพิ่มค่า a
เพิ่มค่า c

ถ้า a เป็น 2 ให้แสดงผล “Prime Number”

ถ้าไม่ใช่ให้แสดงผล “Not Prime Number”

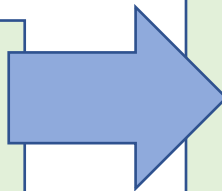
Detailed Pseudo-Code



Simple-Command
Pseudo-Code

B

รับตัวเลขจำนวนเต็ม เก็บใน x
หาตัวเลขใน $2..x$ ตัวแรกที่หาร x ลงตัว เก็บใน a
ถ้า $a = x$ ให้แสดงผล "Prime Number"
ถ้าไม่ใช่ให้แสดงผล "Not Prime Number"



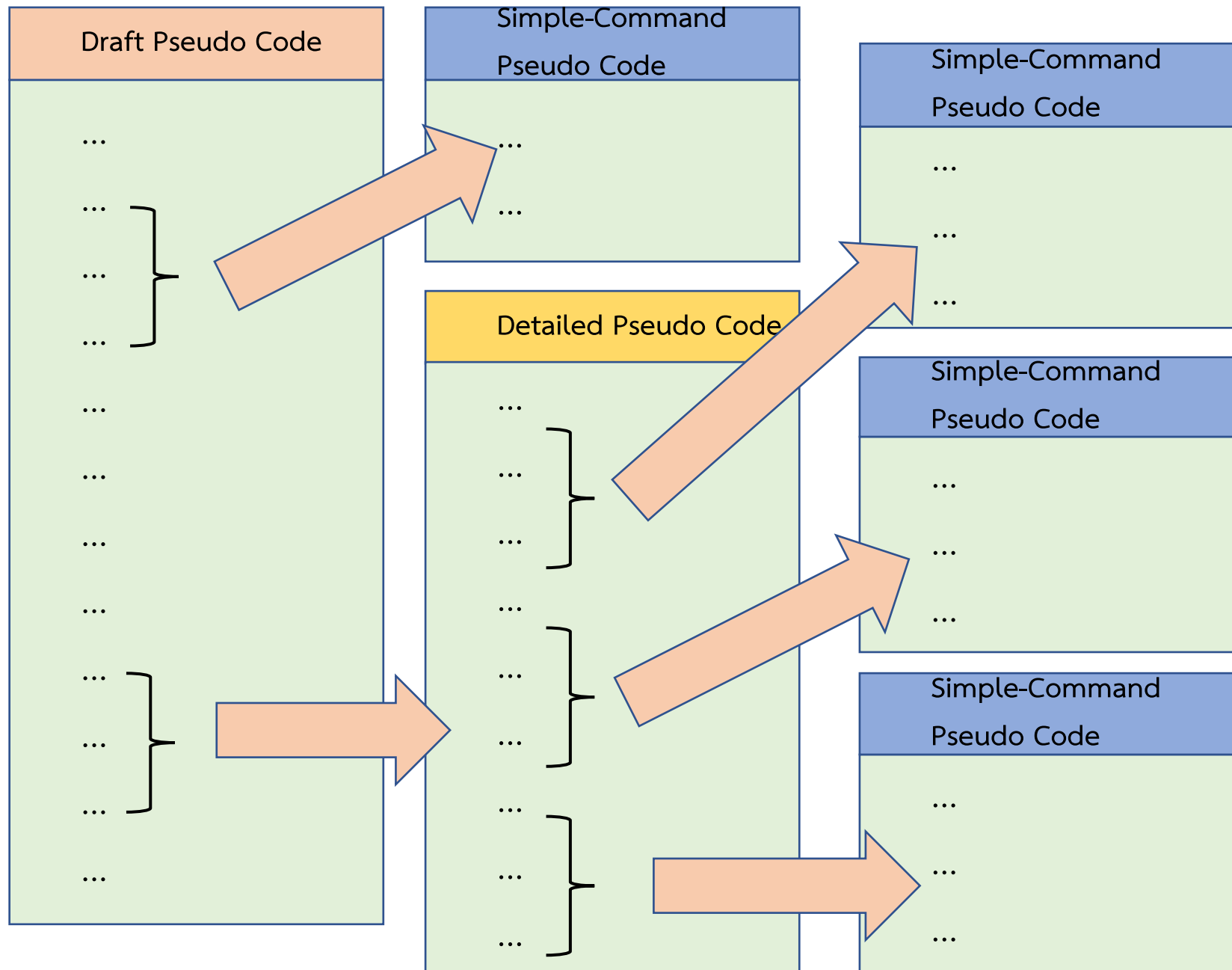
รับตัวเลขจำนวนเต็ม เก็บใน x

กำหนดตัวนับ $a = 2$

ทำซ้ำถ้า x หารด้วย a เศษมากกว่า 0:
เพิ่มค่า a

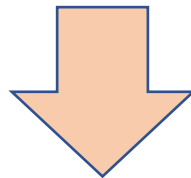
ถ้า $a = x$ ให้แสดงผล "Prime Number"

ถ้าไม่ใช่ให้แสดงผล "Not Prime Number"



#5 : Detailed Pseudo Code ที่มีแต่การทำงานพื้นฐาน
คือ Simple-Command Pseudo-Code ที่พร้อม
เปลี่ยนเป็น C Program

Simple Command
Pseudo-Code



C Program

ตัวอย่าง top-down approach ของการคิดเลขจำนวนเฉพาะ

Draft Pseudo Code

- รับ input เก็บในตัวแปร x
- ถ้า x เป็นจำนวนเฉพาะ
- ให้แสดงผล “prime number”
- ถ้าไม่ใช่
- ให้แสดงผล “not prime number”



จากชุดตัวเลข 2.. n หาตัวเลขตัวแรกที่นำมาหาร x ได้ลงตัว
ถ้า ตัวเลขนั้นคือ x (x เป็นจำนวนเฉพาะ)



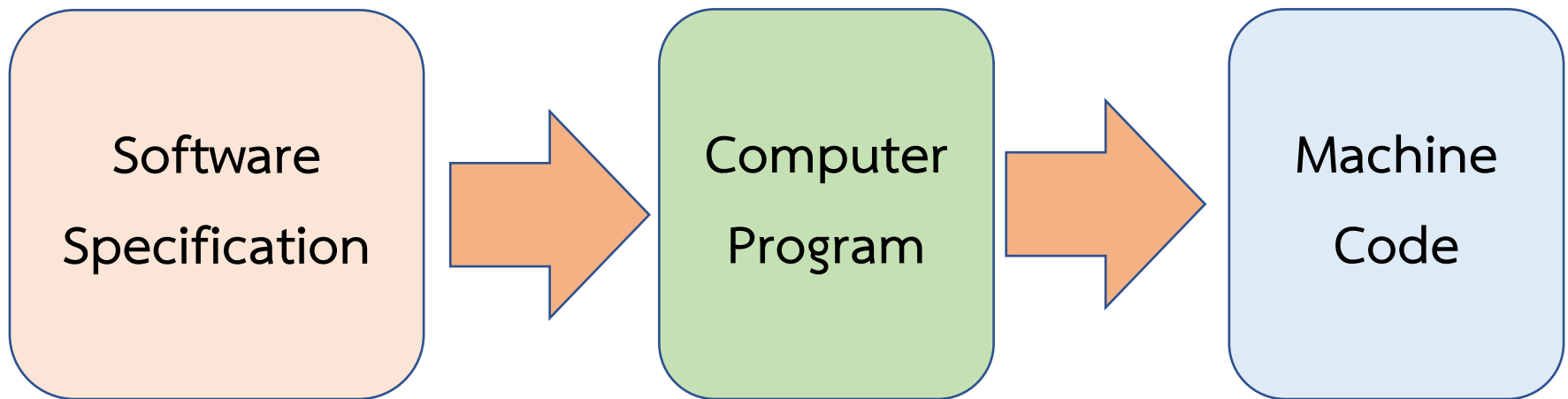
ให้ $a=2$
ทำซ้ำถ้า $a \leq x$ และ x หาร a ไม่ลงตัว
 $a=a+1$
ถ้า $x==a$

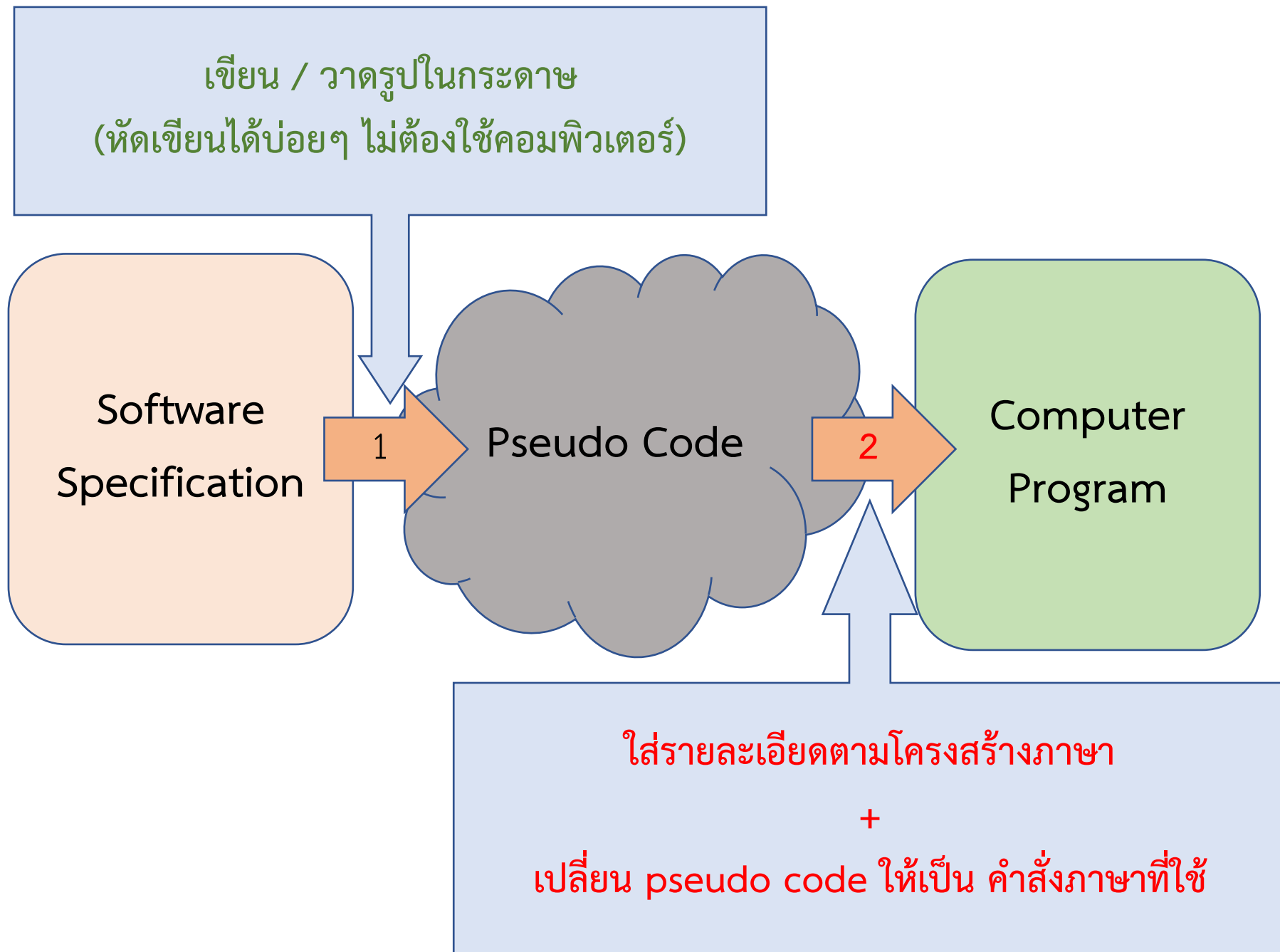
ตัวอย่าง top-down approach ของการคิดเลขจำนวนเฉพาะ

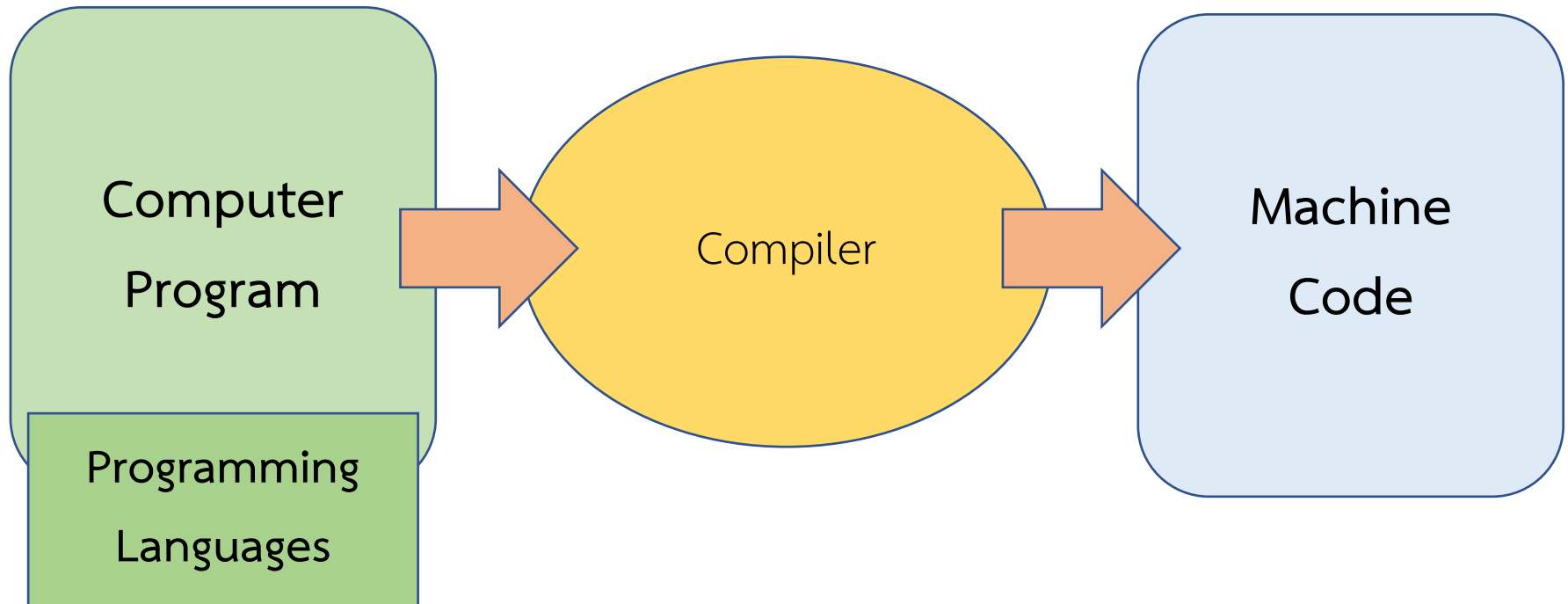
Simple Command Pseudo Code

- รับ input เก็บในตัวแปร x
- ให้ $a=2$
- ทำซ้ำถ้า $a \leq x$ และ xหาร a ไม่ลงตัว
- $a=a+1$
- ถ้า $x==a$
- ให้แสดงผล “prime number”
- ถ้าไม่ใช่
- ให้แสดงผล “not prime number”

From Pseudo Code to C Program









Project < programmer ต้องสร้างเอง >

Header File

Resource File

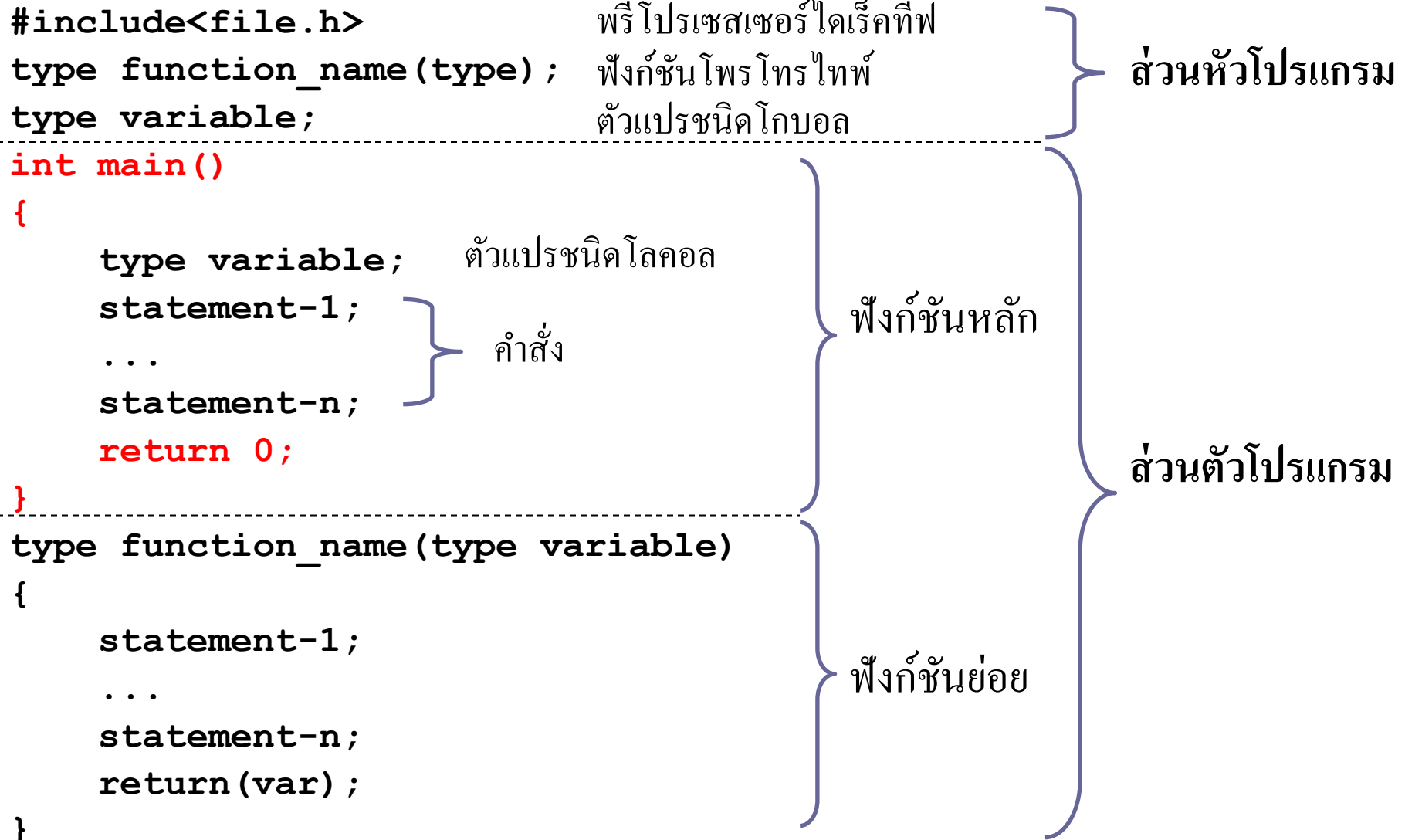
Source File < programmer ต้องสร้างเอง >

- file.c <- ภาษา C จากการแปลง Pseudo Code

#Activity : เครื่องมือ Visual Studio 2017
< สร้าง project และ source file >



โครงสร้างโปรแกรมภาษาซี



โปรแกรมที่ไม่มีการทำงาน แต่ตรงตามโครงสร้างภาษา C

```
int main()  
{  
    return 0;  
}
```

#Activity : เครื่องมือ Visual Studio 2017
< compile + run ไฟล์ภาษา C ที่ถูกต้องตามโครงสร้าง >



หลักการ 3 ข้อ

Pseudo Code -> C Program

#1 เปลี่ยน Pseudo Code ให้เป็นคำสั่งภาษา
C ที่ทำงานสอดคล้องกัน

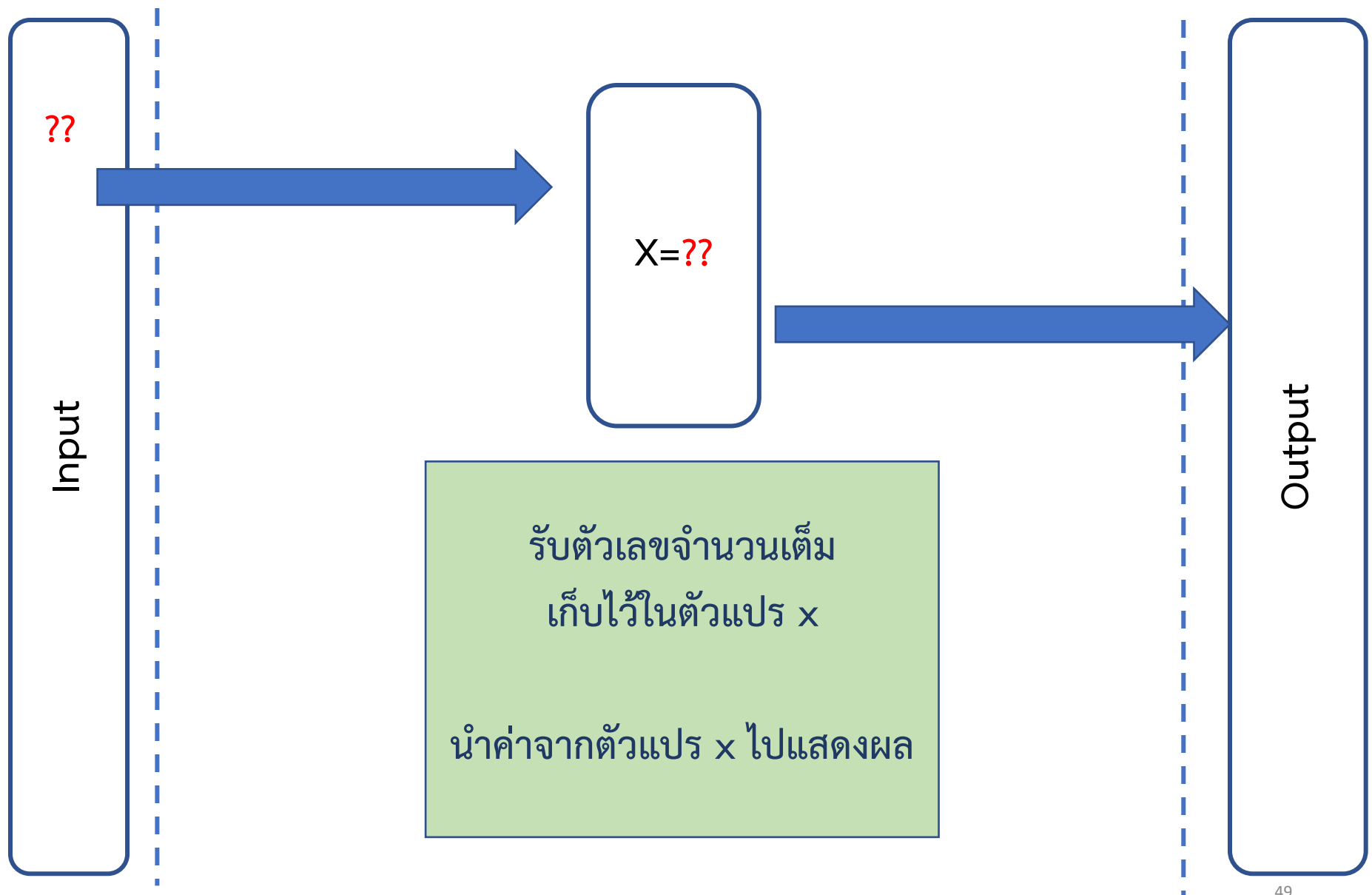
เรียงลำดับให้ถูกต้องใน { }

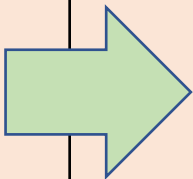
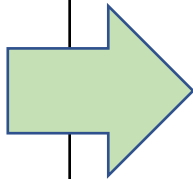
การทำงานพื้นฐานของคอมพิวเตอร์

- รับข้อมูลเก็บไว้ในตัวแปร X
- นำค่าจากตัวแปร X ไปแสดงผล
- คำนวณค่า/กำหนดค่า X ตามสูตร.....

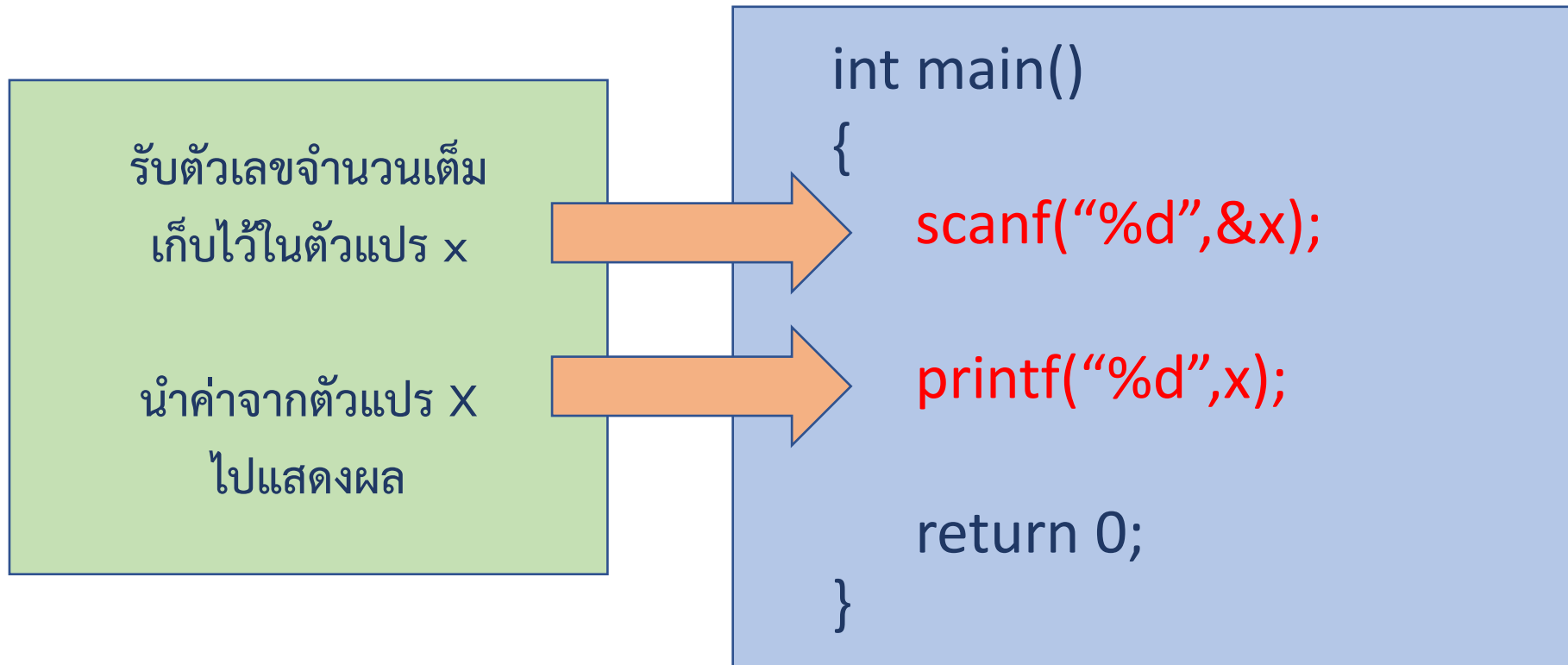
โค้ดเทียม
(pseudo code)

- เมื่อ $f(x)$ เป็นจริง ให้ทำงาน..... ถ้าเป็นเท็จ ให้ทำงาน
- เมื่อ $f(x)$ เป็นจริง ให้ทำงาน ซ้ำไปเรื่อยๆ



Pseudo Code	ตัวอย่างคำสั่งภาษา C ที่สอดคล้องกัน
รับข้อมูลเก็บไว้ใน ตัวแปร x 	x เป็นจำนวนเต็ม : <code>scanf("%d",&x);</code> x เป็นทศนิยม : <code>scanf("%f",&x);</code> x เป็นตัวอักษร : <code>scanf("%c",&x);</code>
นำค่าจากตัวแปร x ไปแสดงผล 	x เป็นจำนวนเต็ม : <code>printf("%d",x);</code> x เป็นทศนิยม : <code>printf("%f",x);</code> x เป็นตัวอักษร : <code>printf("%c",x);</code>

เลือกใช้คำสั่งไหน ??



#2 มีการใช้ตัวแปรเมื่อไหร่ ต้องประกาศตัวแปรก่อนใช้งาน

x เป็นจำนวนเต็ม -> `int x;`

x เป็นทศนิยม -> `float x;`

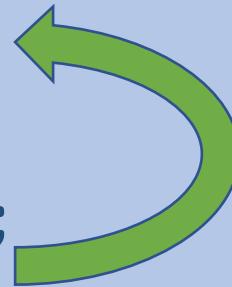
x เป็นตัวอักษร -> `char x;`

```
int main()  
{
```

```
    int x;
```

```
    scanf("%d",&x);  
    printf("%d",x);  
    return 0;
```

```
}
```



มีการใช้งานตัวแปร x
x เป็นจำนวนเต็ม

#3 แต่ละคำสั่งต้องเพิ่ม `#include` ที่สอดคล้อง
เพื่อให้ compiler รู้จักคำสั่งและ compile ได้

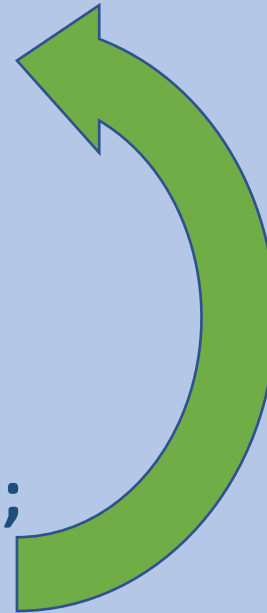
#include<???	คำสั่ง
#include<stdio.h>	scanf() , printf()
#include<math.h>	fabs() , pow() , sqrt() sin() , cos() , log()
#include<iostream>	system()

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

```
int main()  
{
```

```
    int x;  
    scanf("%d",&x);  
    printf("%d",x);  
    return 0;
```

```
}
```



มีการใช้งาน

scanf() , printf()

เพิ่ม

#include<stdio.h>

#Note: ในขั้นตอนการ compile จะมีการตรวจสอบคำสั่งที่ใช้ซึ่ง Visual Studio 2017 จะพิจารณาว่าคำสั่ง scanf ไม่ปลอดภัยจึงไม่อนุญาตให้โปรแกรมทำงาน

แก้ไขโดย

แก้คำสั่ง scanf เป็น `scanf_s`

หรือ

เพิ่ม `#define _CRT_SECURE_NO_WARNINGS 1` ที่บรรทัดแรกของโปรแกรม

```
#define _CRT_SECURE_NO_WARNINGS 1  
#include<stdio.h>
```

```
int main()
```

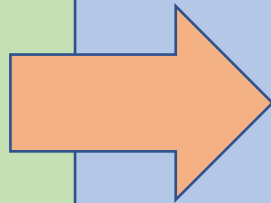
```
{
```

```
    int x;  
    scanf("%d",&x);  
    printf("%d",x);  
    return 0;
```

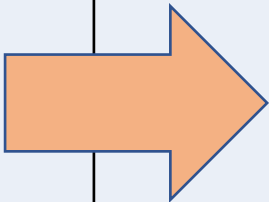
```
}
```

รับตัวเลขจำนวนเต็ม
เก็บไว้ในตัวแปร x

นำค่าจากตัวแปร x ไปแสดงผล



Pseudo Code	ตัวอย่างคำสั่งภาษา C
ให้โปรแกรมหยุดทำงาน จนกว่าจะมีการกดปุ่ม	<code>system("pause")</code>
เลื่อน cursor ไปบรรทัดถัดไป	<code>printf("\n")</code>
แสดงผล *	<code>printf("*")</code>

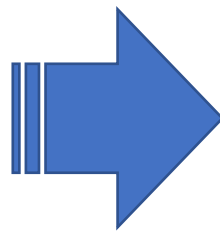
Pseudo Code	ตัวอย่างคำสั่งภาษา C
ถ้า (a>b) ให้ทำงาน xxxx; ถ้าไม่ใช่ให้ทำงาน yyyy;	 <pre> if (a>b) { xxxx; } else { yyyy; } </pre>

การเปรียบเทียบ	เครื่องหมายที่ใช้
มากกว่า	>
น้อยกว่า	<
เท่ากับ	==
มากกว่าหรือเท่ากับ	>=
น้อยกว่าหรือเท่ากับ	<=
ไม่เท่ากับ	!=

รับตัวเลขจำนวนเต็ม
เก็บไว้ในตัวแปร x

นำค่าจากตัวแปร X ไป
แสดงผล

เพิ่ม Pseudo Code
การทำงานของโปรแกรม
ตรวจสอบว่าป้อนค่า
น้อยกว่า 10 หรือไม่



รับตัวเลขจำนวนเต็ม
เก็บไว้ในตัวแปร x

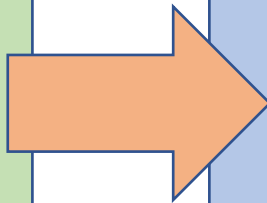
นำค่าจากตัวแปร X ไป
แสดงผล

ถ้า $x < 10$ ให้แสดงผลว่า
“less than 10”

รับตัวเลขจำนวนเต็ม
เก็บไว้ในตัวแปร x

นำค่าจากตัวแปร x ไป
แสดงผล

ถ้า $x < 10$ ให้แสดงผลว่า
“less than 10”



Compile & Run

```
#define _CRT_SECURE_NO_WARNINGS 1
#include<stdio.h>
#include<iostream>

int main()
{
    int x;
    scanf("%d", &x);
    printf("%d", x);

    if (x < 10)
    {
        printf("less than 10");
    }

    system("pause");
    return 0;
}
```

Pseudo Code	ตัวอย่างคำสั่งภาษา C
<p data-bbox="241 649 801 935">ถ้า (a>b) ให้ทำงาน xxxx; yyyy; ซ้ำไปเรื่อย ๆ</p>	<p data-bbox="1159 564 1468 1021">while (a>b) { xxxx; yyyy; }</p>

แสดงตัวเลขตั้งแต่ 1..x

รับตัวเลขจำนวนเต็ม
เก็บไว้ในตัวแปร x

กำหนดค่า i=1
ถ้า i<=x ให้ทำงาน...

แสดงผลค่า i
เพิ่มค่า i
ซ้ำไปเรื่อยๆ

```
#define _CRT_SECURE_NO_WARNINGS 1
#include<stdio.h>
#include<iostream>
int main()
{
    int x,i;
    scanf("%d", &x);
    i = 1;
    while (i<=x)
    {
        printf("%d",i);
        i = i + 1;
    }
    system("pause");
    return 0;
}
```


Note : ให้นักศึกษาศึกษาวิธีการเขียน Draft Pseudo-Code และเพิ่มรายละเอียดจนเป็น Simple-Command Pseudo-Code พร้อมทั้งเขียนโปรแกรมเพื่อทดสอบการทำงานในเอกสาร Case Study

3

```

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

4

```

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