

บทที่ 4 ผลงาน และการเขียนโปรแกรม กำหนดเงื่อนไข

ศท211 วิทยาการคอมพิวเตอร์ 1

โดย อาจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

วัตถุประสงค์

- เขียนผังงานโปรแกรมจากโจทย์ปัญหาได้
- เขียนโปรแกรมจากผังงานที่สร้างขึ้นได้
- เขียนผังงานและโปรแกรมที่มีการกำหนดเงื่อนไขได้

การพัฒนาโปรแกรมคอมพิวเตอร์

- ก่อนการพัฒนาโปรแกรม:
 - ทำความเข้าใจกับปัญหา
 - คิดวางแผนวิธีการแก้ไขปัญหานั้น
- ในขณะที่กำลังพัฒนาโปรแกรม:
 - พิจารณาถึงโครงสร้าง หรือชุดคำสั่งที่มีของภาษาคอมพิวเตอร์ที่ใช้อยู่
 - ทำตามหลักการการพัฒนาโปรแกรมที่ดีตลอดเวลา

คุณลักษณะของโปรแกรมที่ดี

- ทำงานได้ตามที่ต้องการ (Meet requirements)
- ถูกต้องแม่นยำ (Accurate)
- อ่านเข้าใจได้ (Readable)
- ปรับปรุงแก้ไขได้ง่าย (Maintainable)
- ง่ายต่อการใช้ (User friendly)
- มีวิธีการเขียนที่มีระบบแบบแผน

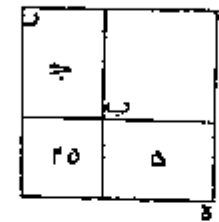
Algorithms



Muhammad ibn Musa al-Khwarizmi
محمد بن موسى الخوارزمي
(c. 780 – c. 850)

ภพ211 วิทยาการคอมพิวเตอร์1

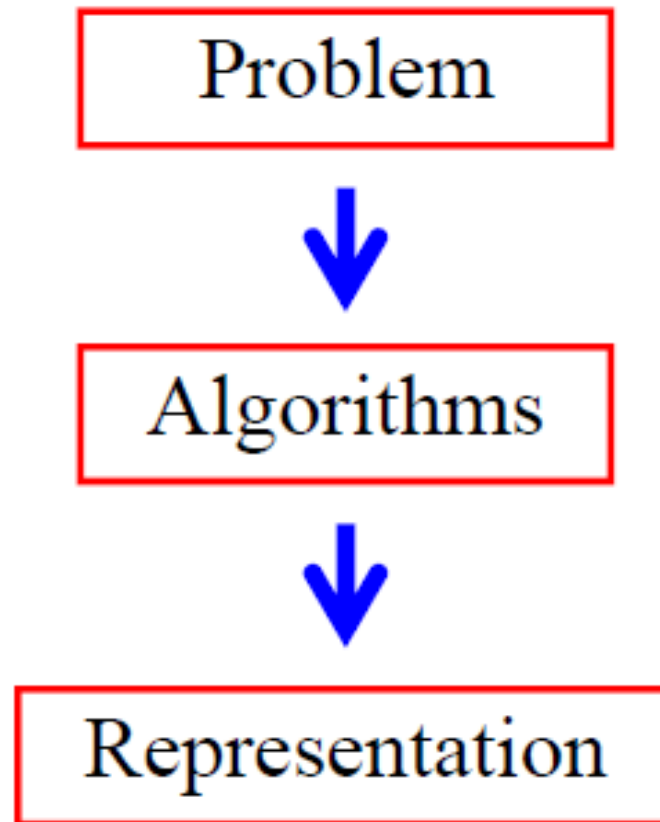
علي تسعة وثلاثين لثم السطح الاعظم الذي هو سطح ره فبلغ
ذلك كله اربعة وستين فاختدنا جذرها وهو ثمانية وهو احد
اضلاع السطح الاعظم فاذا نقصنا منه مثل ما زدنا عليه وهو
خمسة بقي ثلثة وهو ضلع سطح اب الذي هو المال وهو جذره
والمائل تسعة وهذه صورته



واما مال واحد وعشرون درهما يعدل عشرة اجذاره فانا
نجعل المال سطحاً مربعاً مجهول الاضلاع وهو سطح ان ثم نضم
اليه سطحاً متوازي الاضلاع عرضه مثل احد اضلاع سطح ان وهو
ضلع هـ والسطح بـ فصار طول السطحين جميعاً ضلع جـ هـ
وقد علمنا ان طوله عشرة من العدد لان كل سطح مربع
مساوي الاضلاع والزوايا فان احد اضلاعه مضروباً في واحد جذر
ذلك السطح وفي اثنين جذره فلما قال مال واحد وعشرون
يعدل عشرة اجذاره علمنا ان طول ضلع جـ هـ عشرة اعداد لان
ضلع جـ هـ جذر المال فقسّمنا ضلع جـ هـ بنصفين علي نقطة

Algorithms

- Problem
 - สามารถที่จะแก้ไขปัญหานั้นได้โดยการทำงานตามชุดคำสั่งอย่างมีลำดับ
- Algorithms
 - ขั้นตอนที่จะระบุถึงวิธีการแก้ไขปัญหานั้นหรือทำให้งานสำเร็จ
 - มีลำดับการทำงานที่แน่นอน
 - มีความชัดเจน
 - สามารถทำงานได้
 - มีจุดสิ้นสุดการทำงานที่ชัดเจน
- Representation
 - วิธีการแปล Algorithm เป็นภาษาคอมพิวเตอร์

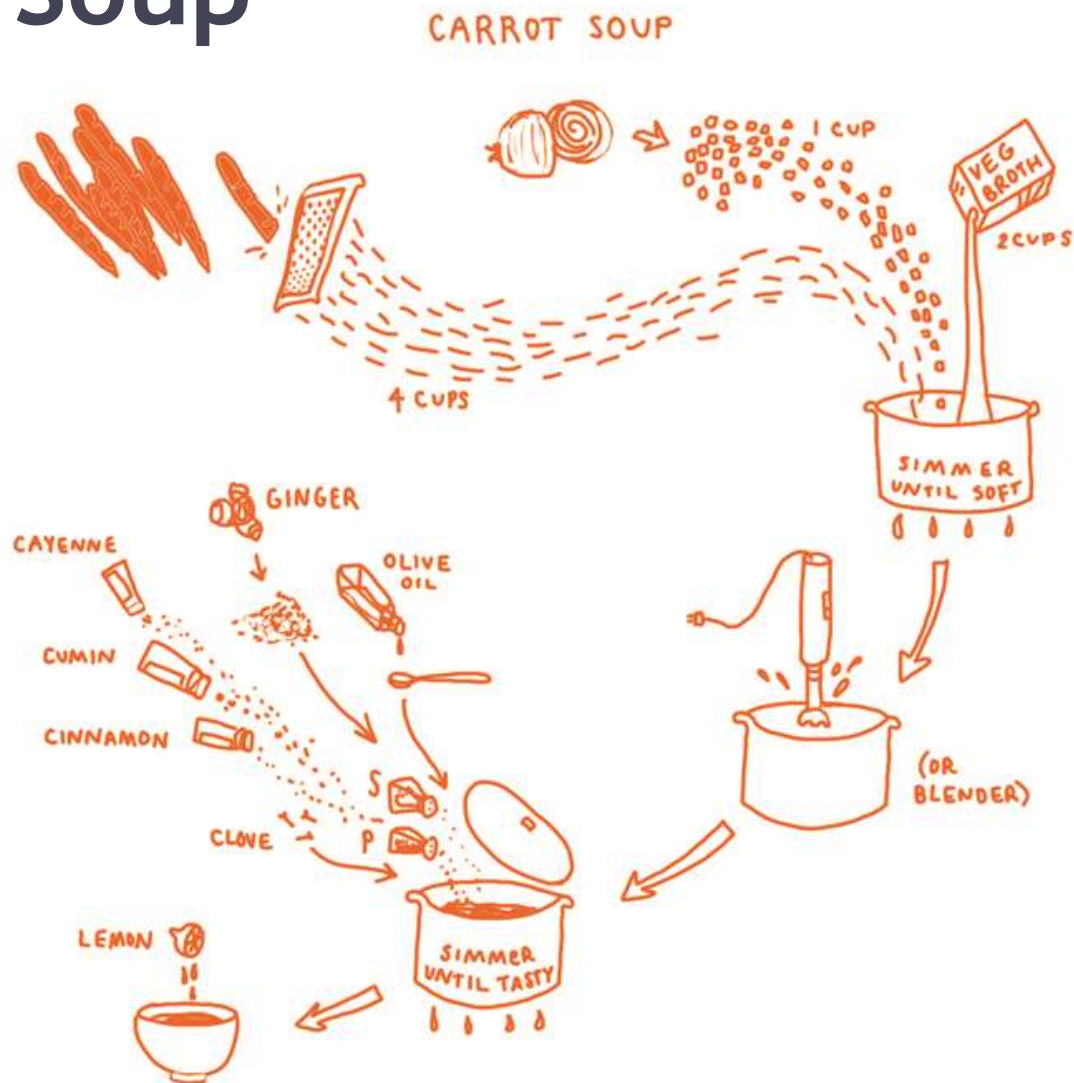


ปัญหา: การใช้เครื่องคอมพิวเตอร์

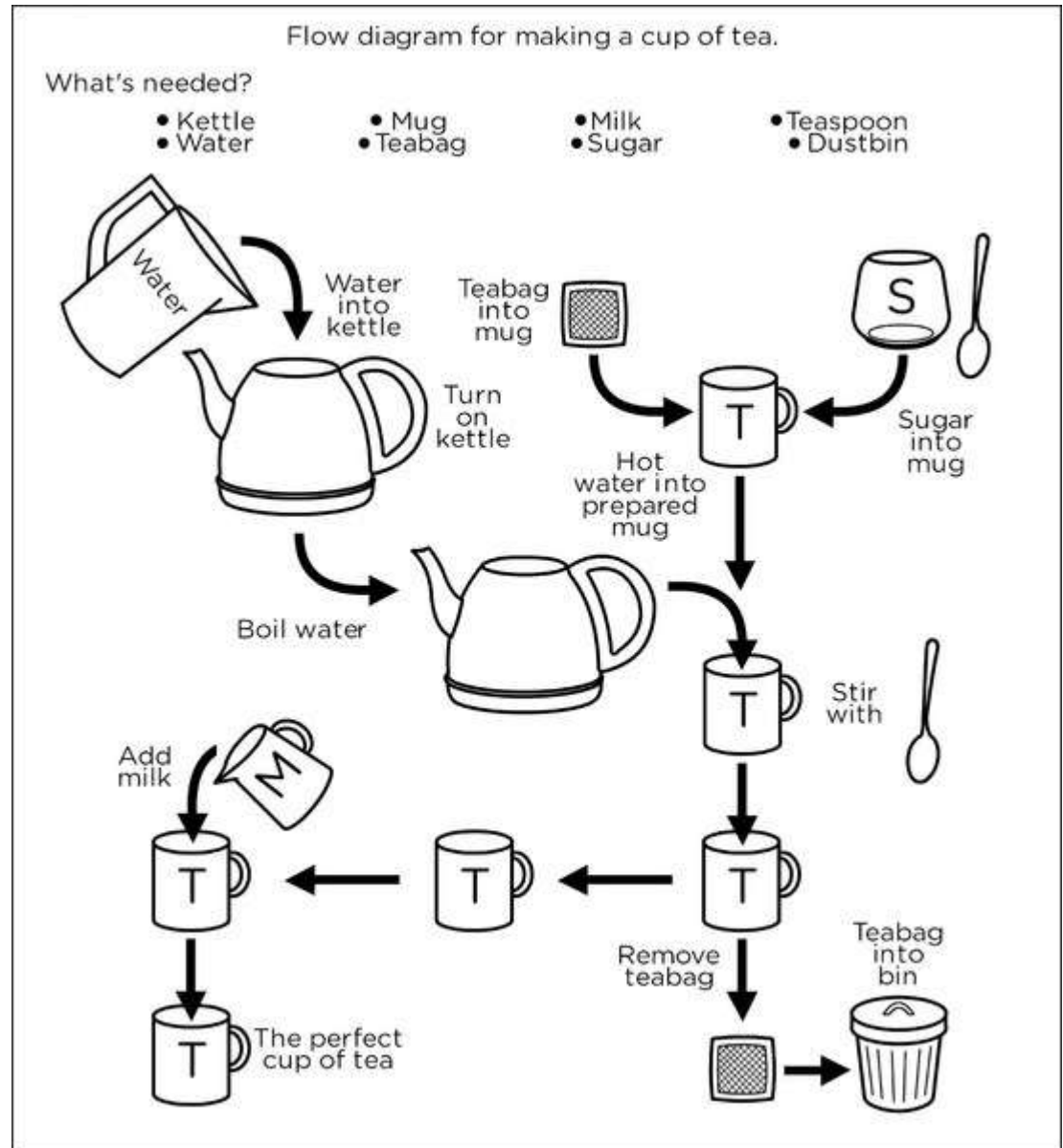
- เปิดเครื่องคอมพิวเตอร์
 - ↓
 - ใส่รหัสผ่าน
 - ↓
 - เปิดโปรแกรม
 - ↓
 - พิมพ์งาน
 - ↓
 - บันทึกข้อมูล
 - ↓
 - ปิดเครื่องคอมพิวเตอร์
- 

- เปิดเครื่องคอมพิวเตอร์
 - ↓
 - ใส่รหัสผ่าน
 - ↓
 - พิมพ์งาน
 - ↓
 - เปิดโปรแกรม
 - ↓
 - บันทึกข้อมูล
 - ↓
 - ปิดเครื่องคอมพิวเตอร์
- 

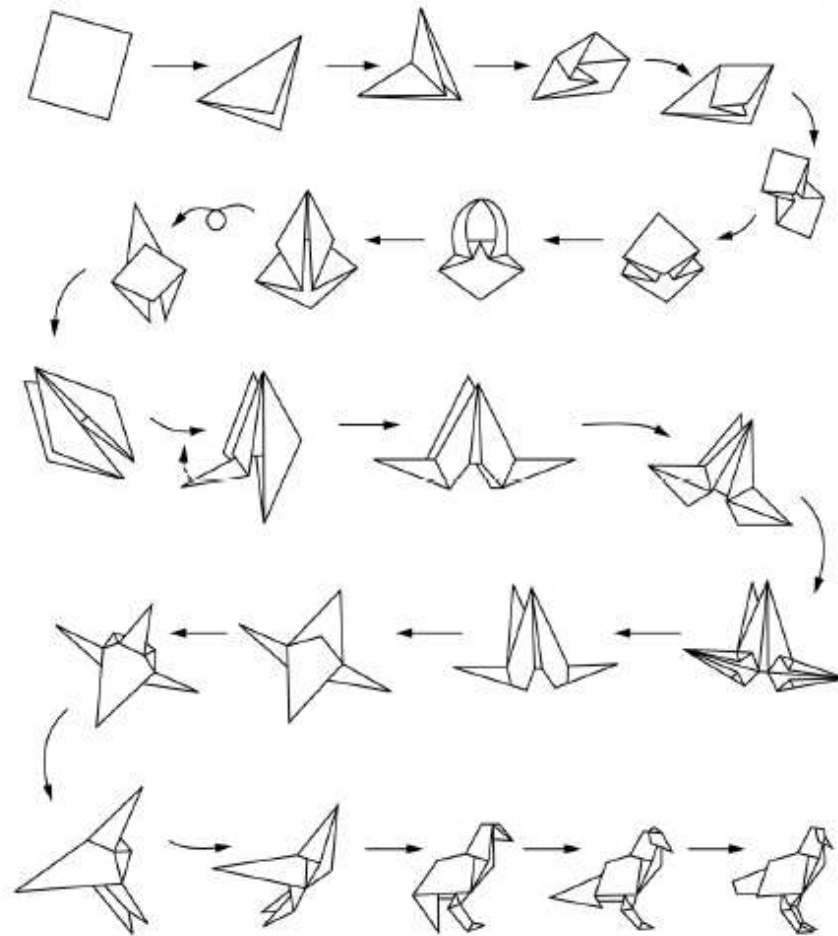
Carrot Soup



A cup of tea



Origami



4.1 ผังงาน (Flowchart)

แผนภาพซึ่งแสดงลำดับขั้นตอนของการทำงาน โดยแต่ละขั้นตอนจะแสดงโดยใช้สัญลักษณ์ ซึ่งมีความหมายบ่งบอกว่าขั้นตอนนั้นๆ มีลักษณะการทำงานแบบใด และในแต่ละขั้นตอนจะเชื่อมโยงกันด้วยลูกศรเพื่อที่จะแสดงลำดับการทำงาน

ประโยชน์ของผังงาน

- ช่วยให้สามารถทำความเข้าใจลำดับขั้นตอนการทำงานของโปรแกรมหรือระบบใดๆ ได้อย่างรวดเร็ว
- ช่วยแสดงลำดับขั้นตอนการทำงาน ทำให้สามารถเขียนโปรแกรมได้อย่างเป็นระบบไม่สับสน
- ช่วยในการแก้ไขข้อผิดพลาดของโปรแกรมได้รวดเร็ว และช่วยให้ผู้พัฒนาโปรแกรมต่อสามารถทำงานได้ง่ายสะดวกขึ้น
- ง่ายแก่บุคคลภายนอกในการติดตามขั้นตอนของการปฏิบัติงาน

ประเภทของผังงาน

- ผังงานระบบ (System Flowchart)

เป็นผังงานซึ่งแสดงขอบเขต และลำดับขั้นตอนการทำงานของระบบหนึ่งๆ รวมทั้งแสดงรูปแบบของข้อมูลเข้า และข้อมูลออกว่าถูกรับเข้าหรือแสดงผลโดยผ่านสื่อประเภทใด เนื่องจากผังงานระบบเป็นแผนภาพที่แสดงถึงระบบโดยรวม ดังนั้นกระบวนการหรือโปรแกรมหนึ่งๆ อาจถูกแสดงเป็นเพียงขั้นตอนหนึ่งในผังงานระบบเท่านั้น

- ผังงานโปรแกรม (Program Flowchart)

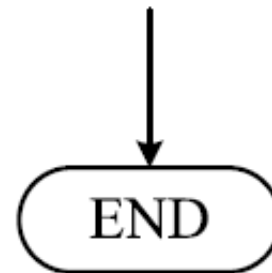
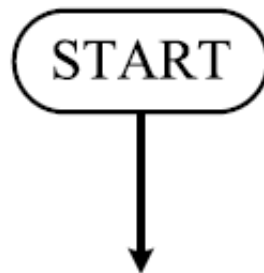
เป็นผังงานซึ่งแสดงลำดับขั้นตอนการทำงานของโปรแกรมหนึ่งๆ

สัญลักษณ์ที่ใช้ในการเขียนผังงาน

คพ211 วิทยาการคอมพิวเตอร์ 1



แทนการเริ่มต้น หรือสิ้นสุดของผังงาน



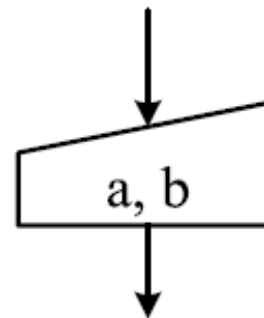
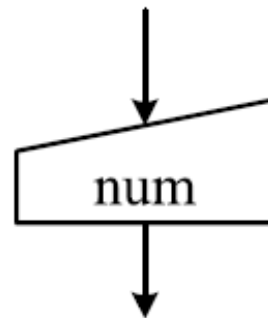
```
main(){  
}
```

```
return 0;
```

สัญลักษณ์ที่ใช้ในการเขียนผังงาน

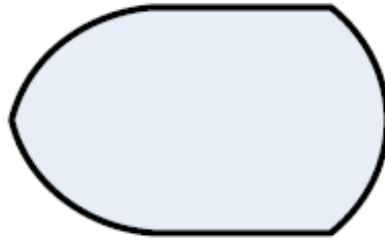


แทนการรับข้อมูลหรือป้อนข้อมูลเข้าทางแป้นพิมพ์

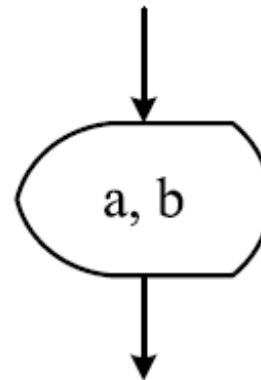
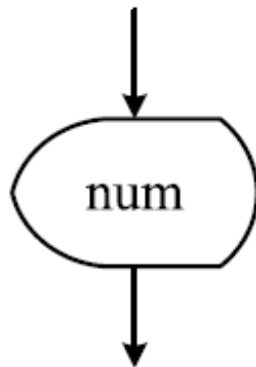


```
scanf("%d %d",&a,&b);
```


สัญลักษณ์ที่ใช้ในการเขียนผังงาน

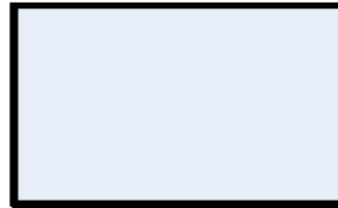


แทนการแสดงผลลัพธ์บนหน้าจอภาพ

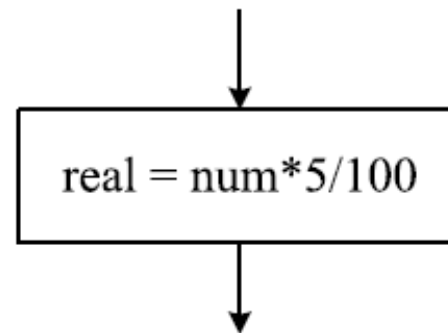
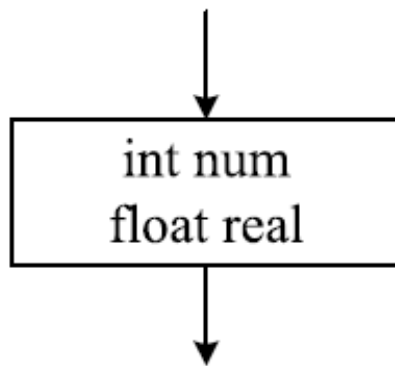


```
printf(“%d %d”,a,b);
```

สัญลักษณ์ที่ใช้ในการเขียนผังงาน

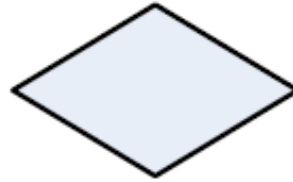


แทนการกำหนดค่าหรือการคำนวณค่า

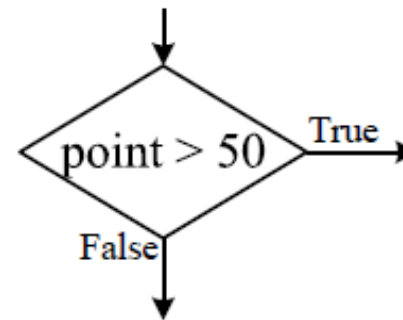
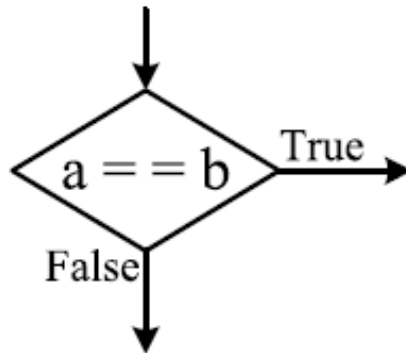


```
real=num*5/100;
```

สัญลักษณ์ที่ใช้ในการเขียนผังงาน



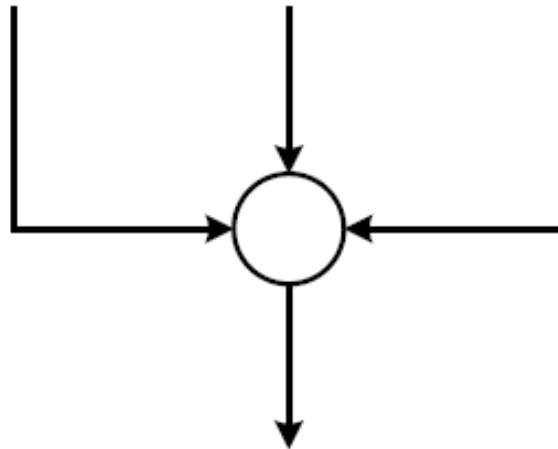
แทนการเปรียบเทียบ



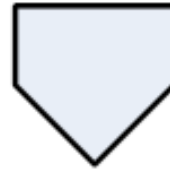
สัญลักษณ์ที่ใช้ในการเขียนผังงาน



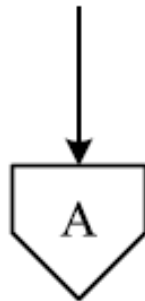
แทนจุดต่อเนื่องที่อยู่หน้าเดียวกัน



สัญลักษณ์ที่ใช้ในการเขียนผังงาน



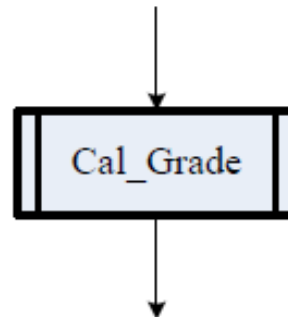
แทนจุดต่อเนืองที่อยู่คนละหน้ากัน




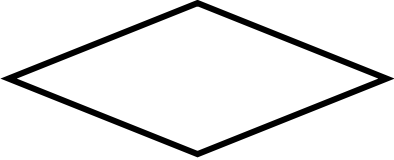


สัญลักษณ์ที่ใช้ในการเขียนผังงาน



แทนการเรียกใช้ฟังก์ชันที่สร้างขึ้น

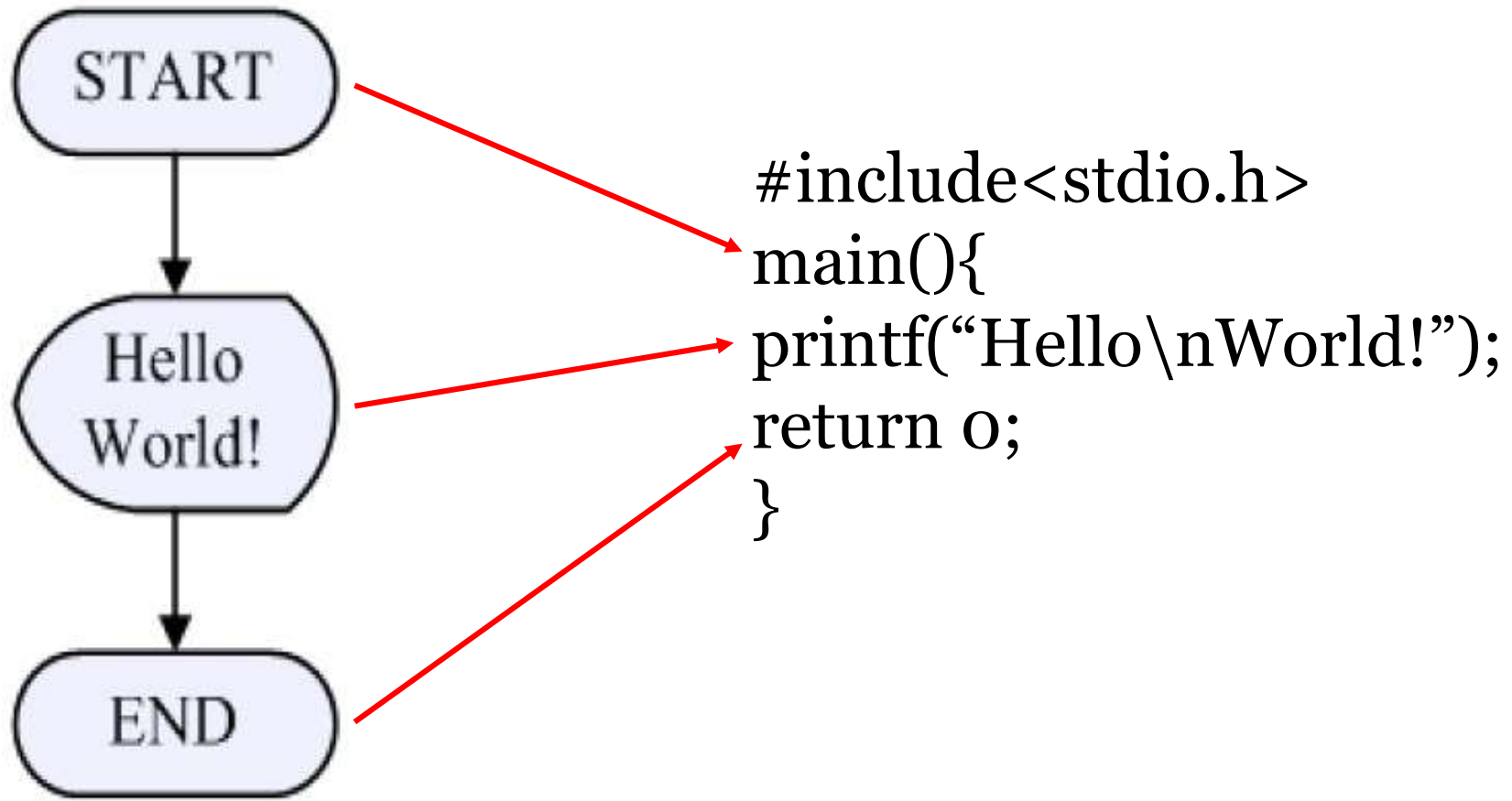


สัญลักษณ์	ความหมาย
	จุดเริ่มต้นหรือจุดจบของโปรแกรม (Terminal)
	การรับและแสดงผลของข้อมูล (Input/Output)
	การประมวลผล (Process)
	การตัดสินใจ (Decision /Selection)

หลักเกณฑ์ในการเขียนผังงาน

- สัญลักษณ์ที่ใช้อาจมีขนาดต่างกันได้ แต่จะต้องมีรูปร่างเป็นสัดส่วนตามมาตรฐาน
- ทิศทางของลูกศรในผังงาน ควรจะมีทิศทางจากบนลงล่าง หรือจากซ้ายไปขวา
- ผังงานควรมีความเรียบร้อย สะอาด พยายามหลีกเลี่ยงการเขียนลูกศรที่ทำให้เกิดจุดตัด เพราะจะทำให้อ่านและทำความเข้าใจผังงานได้ยาก
- ถ้าในผังงานมีการเขียนข้อความอธิบายใดๆ ควรทำให้สั้นกะทัดรัดและได้ใจความ อาจเขียนเป็นคำสั่งที่อยู่ในสัญลักษณ์ หรือใช้คำพูดแทน

ตัวอย่างผังงานโปรแกรม



แนวทางการสร้างผังงานสำหรับการเขียนโปรแกรม

- การวิเคราะห์ข้อมูลเอาต์พุต หรือผลลัพธ์ (Output Analysis)
 - วิเคราะห์ความต้องการของผู้ใช้ หรือผลลัพธ์จากโจทย์ปัญหา
- การวิเคราะห์ข้อมูลอินพุต (Input Analysis)
 - วิเคราะห์ข้อมูลที่ใช้ป้อน หรือข้อมูลที่โจทย์ให้มา
- การวิเคราะห์กระบวนการทำงาน (Process Analysis)
 - วิเคราะห์ขั้นตอนการทำงานที่ให้มาได้มา ซึ่งผลลัพธ์
- การกำหนดตัวแปร (Variable Define)
 - กำหนดตัวแปรที่ใช้งานการเขียนโปรแกรมเพื่อความถูกต้อง

แนวทางสร้างโปรแกรมคำนวณหาพื้นที่วงกลม

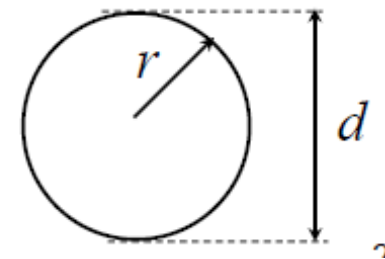
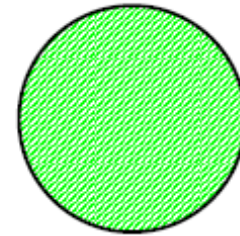
โปรแกรม 4.1 จงเขียนผังงาน และ โปรแกรมคำนวณหาพื้นที่วงกลม

- Output Analysis

ผลลัพธ์ที่ต้องการ คือ พื้นที่ของวงกลม

- Input Analysis

การคำนวณหาพื้นที่วงกลม จำเป็นต้องทราบขนาดของรัศมี (หรือเส้นผ่านศูนย์กลาง)



แนวทางสร้างโปรแกรมคำนวณหาพื้นที่วงกลม

- Process Analysis

1. รอรับค่ารัศมี (หรือเส้นผ่านศูนย์กลาง) จากผู้ใช้งาน
2. คำนวณหาพื้นที่วงกลมจากสูตร $A = \pi r^2$
3. แสดงผลค่าพื้นที่วงกลมออกทางหน้าจอ

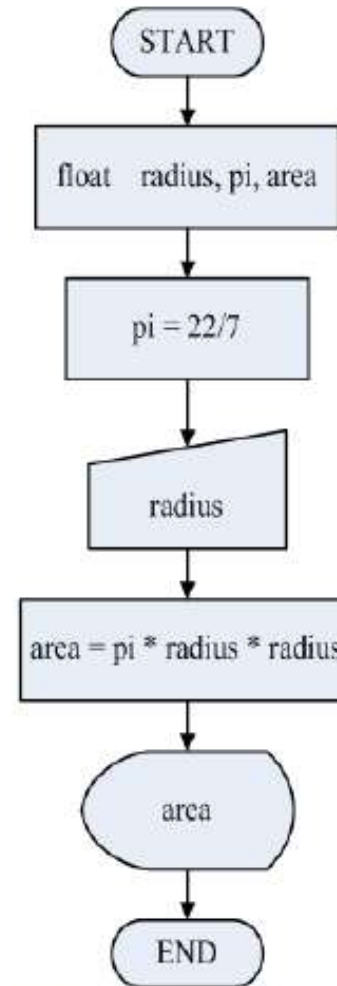
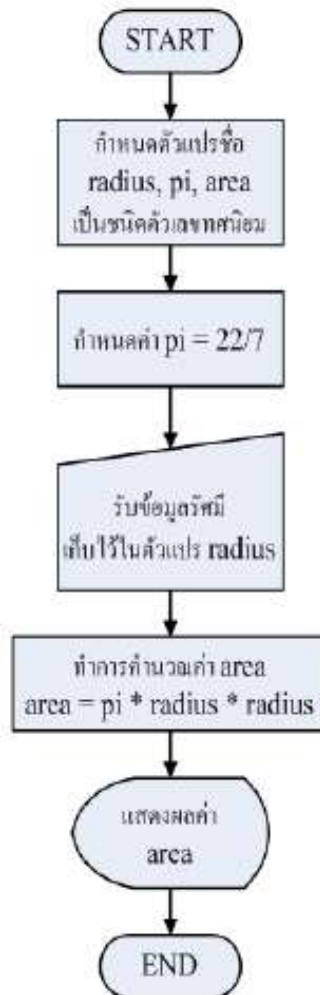
- Variable Define

radius : เป็นตัวแปรชนิดจำนวนทศนิยมสำหรับรับค่ารัศมี

pi : เป็นตัวแปรชนิดจำนวนทศนิยมสำหรับเก็บค่า

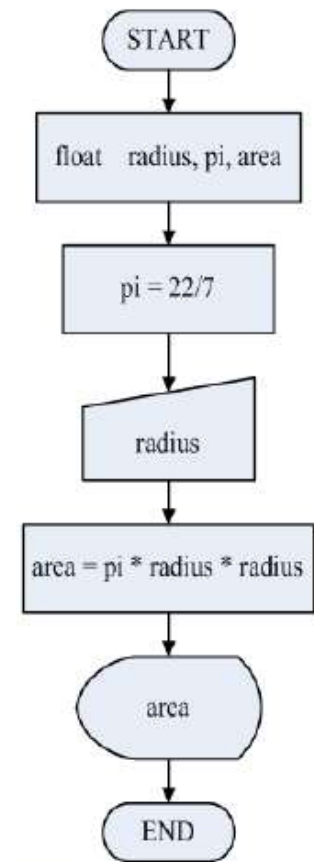
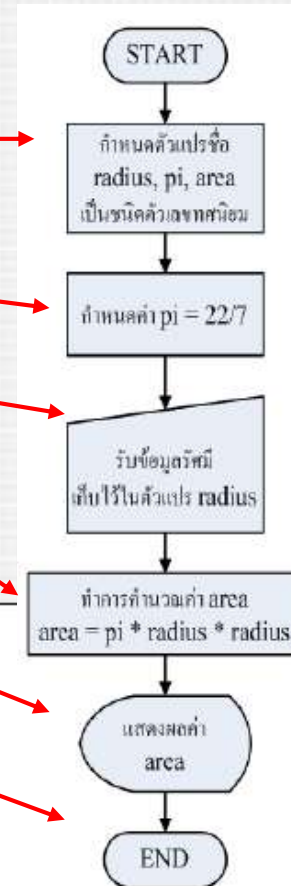
area : เป็นตัวแปรชนิดจำนวนทศนิยมสำหรับเก็บค่าพื้นที่

แนวทางสร้างโปรแกรมคำนวณหาพื้นที่วงกลม



แนวทางสร้างโปรแกรมคำนวณหาพื้นที่วงกลม

```
#include<stdio.h>
int main()
{
    float  radius, pi, area;
    pi = 22.0/7; // pi = 3.14;
    printf ("Enter Radius of Circular : ");
    scanf ("%f",&radius);
    area = pi * radius * radius;
    printf ("Area of Circular : %f",area);
    return 0;
}
```



โจทย์ : โปรแกรมรวมคะแนนเรียน

จงเขียนผังงาน และ โปรแกรมรับ ชื่อ นามสกุล รหัสนักศึกษา
และ คะแนนเก็บ แล้วแสดงผลคำนวณคะแนนรวม

– ตัวอย่างการรันโปรแกรม

```
Enter your Name Surname : Somsak Jaidee
Enter your Student ID   : 40010000
Enter your Score
    Laboratory (15%) = 15
    Midterm (35%)   = 34
    Final (50%)     = 49
Total (100%) = 98
```

4.2 การเขียนโปรแกรมแบบกำหนดเงื่อนไข

- ในการเขียนโปรแกรมสำหรับงานส่วนใหญ่ จำเป็นต้องมีการทดสอบเงื่อนไขบางอย่างก่อน เพื่อตัดสินใจเลือกการทำงานของโปรแกรมในอันดับถัดไป
- คำสั่งควบคุมการทำงานของโปรแกรมที่นิยมใช้มีอยู่ 2 คำสั่ง คือ คำสั่ง if และ คำสั่ง if ... else ซึ่งจะพิจารณาเลือกกระทำหรือไม่กระทำจากการพิสูจน์นิพจน์ว่าเป็น จริง หรือ เท็จ
- โดยจะใช้ควบคู่กับเครื่องหมายเปรียบเทียบ และเครื่องหมายทางตรรกศาสตร์

4.2.1 การเปรียบเทียบ

เครื่องหมาย	การเปรียบเทียบ	ตัวอย่าง
==	เท่ากับ	x == y
!=	ไม่เท่ากับ	x != y
>	มากกว่า	x > y
>=	มากกว่าหรือเท่ากับ	x >= y
<	น้อยกว่า	x < y
<=	น้อยกว่าหรือเท่ากับ	x <= y

ผลของการเปรียบเทียบจะได้ค่าจริง (ค่าที่ไม่ใช่ 0) หรือค่าเท็จ (ค่าที่เป็น 0)

การใช้งานเครื่องหมายการเปรียบเทียบ

การเปรียบเทียบ	ผลที่ได้
$7 == 9$	False
$7 != 9$	True
$8 > 8$	False
$8 >= 8$	True
$(10+9) < 7$	False
$4 <= 3$	False

การเปรียบเทียบ	ผลที่ได้
$22 == 22$	True
$(3+5) != 8$	False
$9 > 7$	True
$7 >= 9$	False
$7 < (10+9)$	True
$3 <= 4$	True

ไม่ควรใช้เครื่องหมายเท่ากับ $==$ หรือไม่เท่ากับ $!=$ สำหรับข้อมูลทศนิยม

4.2.2 เครื่องหมายทางตรรกศาสตร์

เครื่องหมาย	ความหมาย	ตัวอย่าง
&&	และ (and)	$x \ \&\& \ y$
	หรือ (or)	$x \ \ y$
!	ไม่ หรือ ตรงกันข้าม (not)	$!x$



การใช้งานเครื่องหมายทางตรรกศาสตร์

การดำเนินการ	ผลที่ได้
T & T	T
T & F	F
F & T	F
F & F	F

การดำเนินการ	ผลที่ได้
T T	T
T F	T
F T	T
F F	F

การดำเนินการ	ผลที่ได้
! T	F
! F	T

การใช้งานเครื่องหมายทางตรรกศาสตร์

```
int num1 = 10, num2 = 20, num3 = 30;
```

```
num1 == num2
```

```
num1 > num2
```

```
(num1 < num2) && (num2 < num3)
```

```
(num1 > num2) || (num1 > num3)
```

```
(num1 > num2) || (num2 < num3)
```

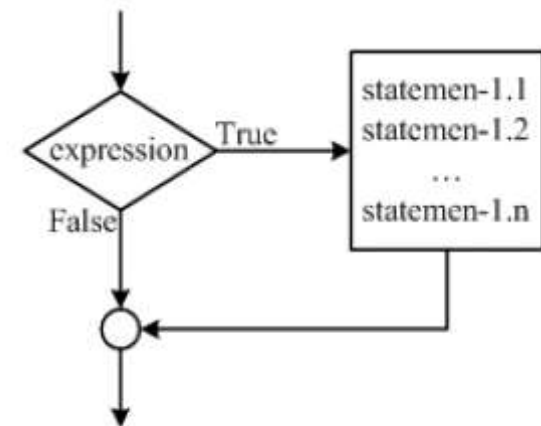
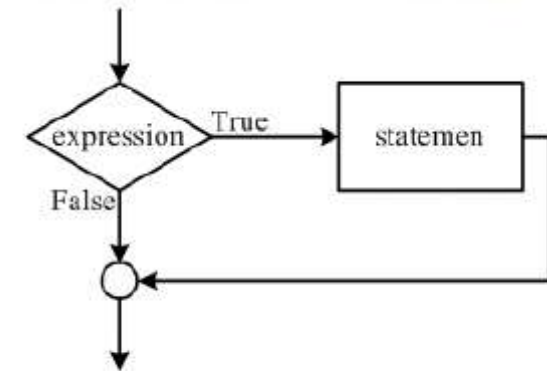
4.3 การใช้คำสั่ง **if**

คพว211 วิทยาการคอมพิวเตอร์ 1

if (*expression*)
statement,

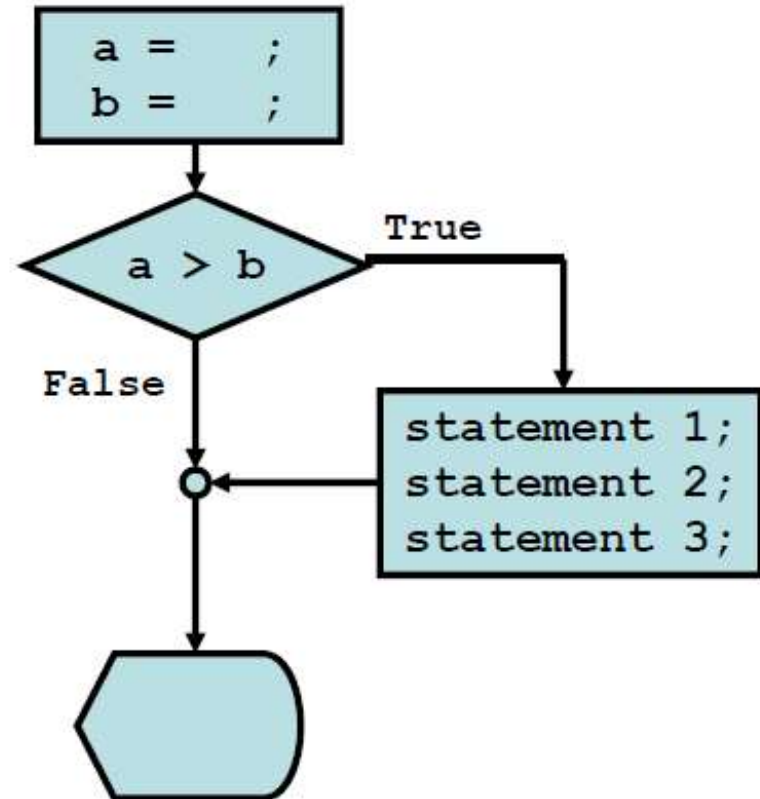
```

if (expression)
{
    statement-1.1;
    statement-1.2;
    ...
    statement-1.n;
}
  
```



การใช้คำสั่ง **if** กับนิพจน์

```
int a = ;  
int b = ;  
if (a>b)  
{  
    statement 1;  
    statement 2;  
    statement n;  
}  
printf();
```



โปรแกรม 4.2 ตรวจสอบคะแนน | **if**

จงเขียนผังงานและ โปรแกรมรับชื่อนามสกุล รหัสนักศึกษา
คะแนนสอบรวม และคะแนนเต็ม หากนักศึกษาสอบได้
มากกว่า 60% ให้แสดงผลชื่อ นามสกุล รหัสนักศึกษา คะแนน
และผลสอบว่าผ่าน

- Output Analysis
 - แสดงผลชื่อ-สกุล รหัสนักศึกษา คะแนนสอบ ผลสอบ

โปรแกรมตรวจสอบคะแนน | if

- Input Analysis
 - ชื่อ / นามสกุล / รหัสนักศึกษา / คะแนนสอบ / คะแนนเต็ม
- Process Analysis
 - โปรแกรมรอรับชื่อ / นามสกุล / รหัสนักศึกษา / คะแนนสอบ / คะแนนเต็ม
 - ตรวจสอบว่าคะแนนมากกว่าหรือเท่ากับ 60 % หรือไม่
 - ถ้าจริง แสดงผลชื่อ-สกุล รหัสนักศึกษา คะแนน และแสดงว่าสอบผ่าน

โปรแกรมตรวจสอบคะแนน | if

- Variable Define

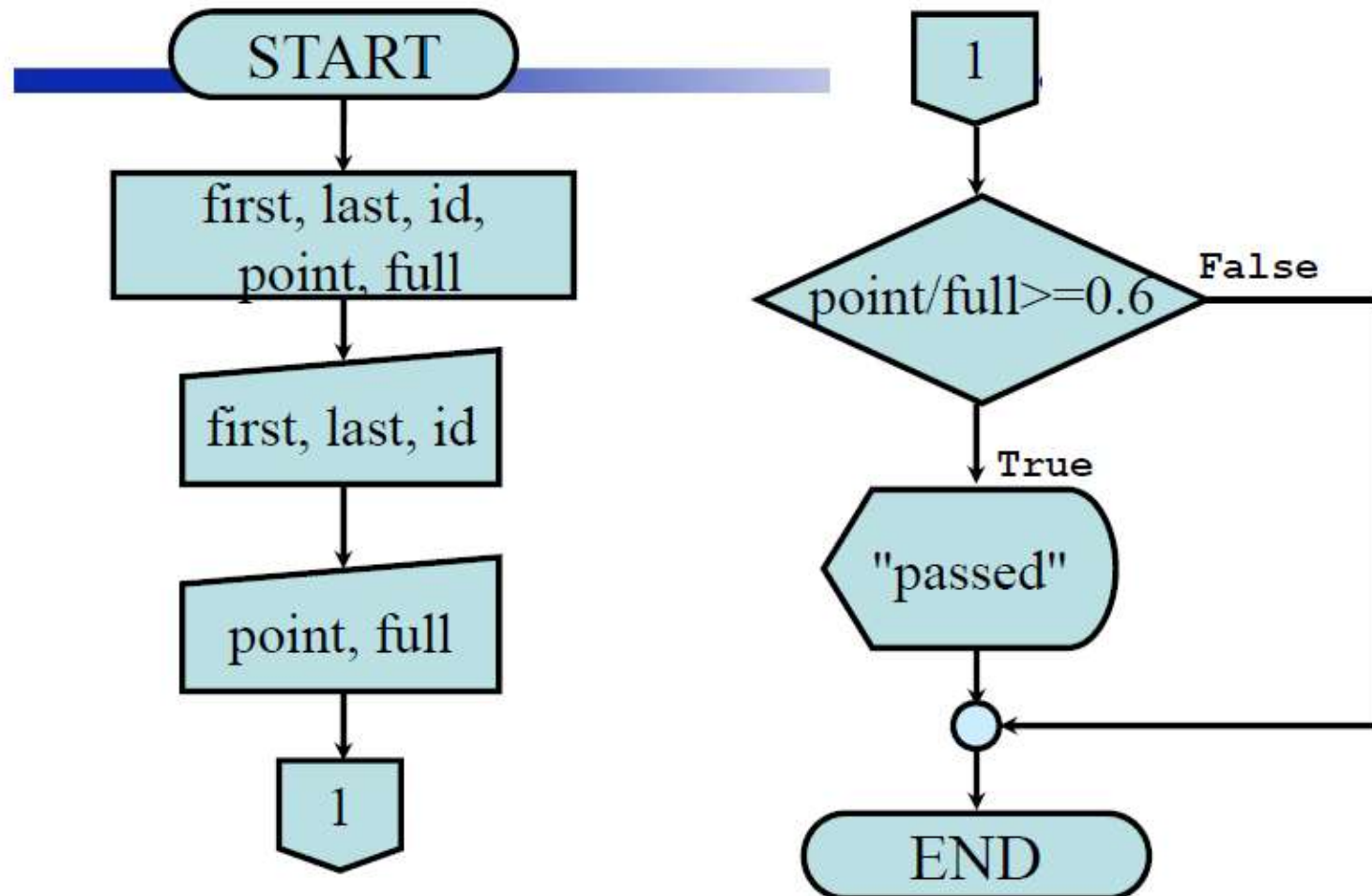
first : ตัวแปรชนิดข้อความสำหรับเก็บชื่อขนาด 20

last : ตัวแปรชนิดข้อความสำหรับเก็บนามสกุลขนาด 20

id : ตัวแปรชนิดข้อความสำหรับเก็บรหัสนักศึกษาขนาด 9

point : ตัวแปรชนิดจำนวนทศนิยมสำหรับเก็บคะแนนรวม

full : ตัวแปรชนิดจำนวนทศนิยมสำหรับเก็บคะแนนเต็ม



โปรแกรมตรวจสอบคะแนน | if

```
#include<stdio.h>
int main()
{
    char    first[20], last[20], id[9];
    float   point,full;

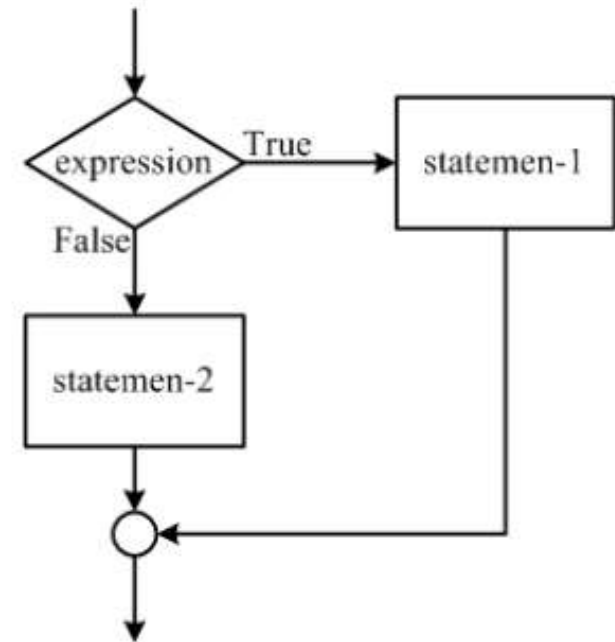
    printf ("Enter your Name      : ");
    scanf  ("%s",first);
    printf ("Enter your Surname : ");
    scanf  ("%s",last);
    printf ("Enter your ID        : ");
    scanf  ("%s",id);
```

โปรแกรมตรวจสอบคะแนน | if

```
printf ("Enter your examination points : ");
scanf ("%f",&point);
printf ("Enter your total points          : ");
scanf ("%f",&full);
if ((point/full) >= 0.6)
{
    printf ("Name : %s %s\n",first,last);
    printf ("ID   : %s\n",id);
    printf ("Examination points : %f / %f\n",point,full);
    printf ("You passed, Congratulation\n");
}
return 0;
}
```

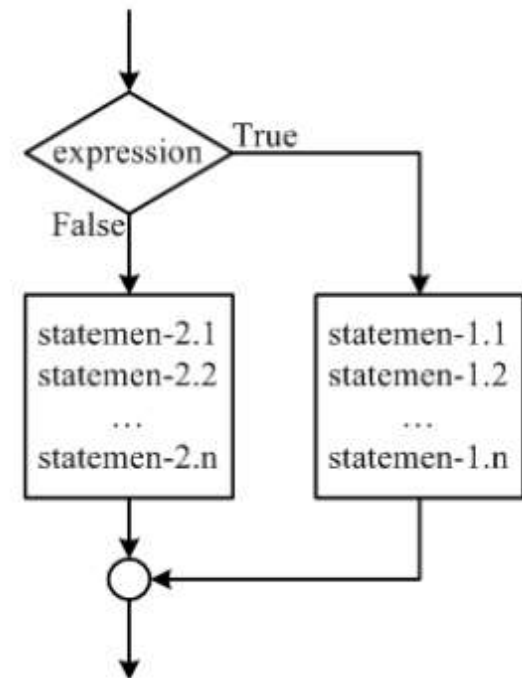
4.4 การใช้คำสั่ง **if-else** | ภายในเงื่อนไขมีคำสั่งเดียว

```
if (expression)  
    statement-1;  
else  
    statement-2;
```



4.4 การใช้คำสั่ง **if-else** | ภายในเงื่อนไขมีหลายคำสั่ง

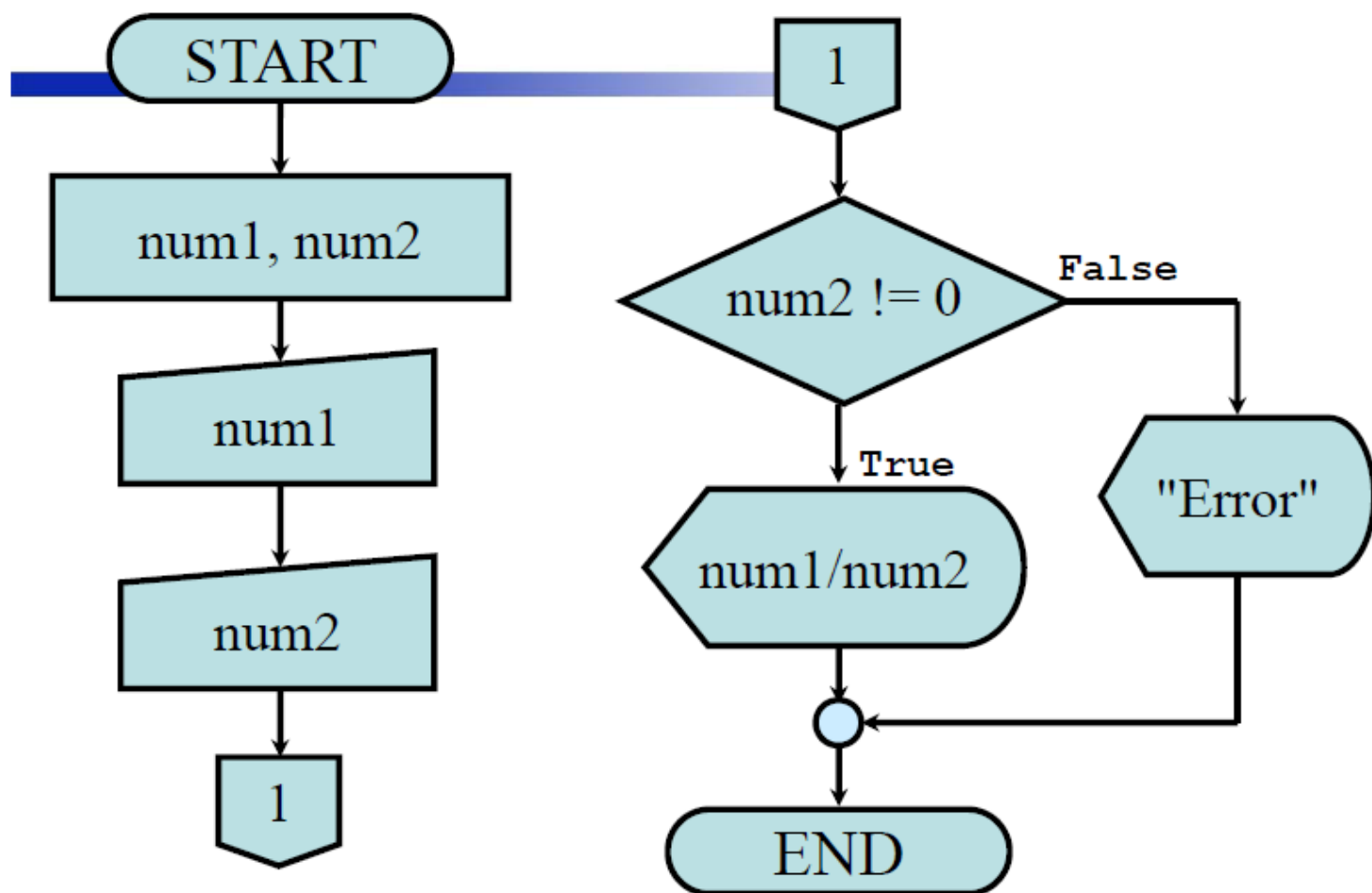
```
if (expression)
{
    statement-1.1;
    statement-1.2;
    ...
    statement-1.n;
}
else
{
    statement-2.1;
    statement-2.2;
    ...
    statement-2.n;
}
```



โปรแกรม4.3 หารเลข 2 จำนวน

จงเขียนผังงานและโปรแกรมหารเลข 2 จำนวน โดยโปรแกรมต้องตรวจสอบได้ว่าตัวหารเป็น "0" หรือไม่

- Output Analysis
 - แสดงผลหารของเลข 2 จำนวน
 - แสดงผลว่าไม่สามารถหารได้เพราะตัวหารเป็นศูนย์
- Input Analysis
 - ตัวตั้ง และตัวหาร



โปรแกรมหารเลข | if-else

```
#include<stdio.h>
int main()
```

```
{
```

```
float num1,num2;
```

```
printf ("Enter number 1 : ");
```

```
scanf ("%f",&num1);
```

```
printf ("Enter number 2 : ");
```

```
scanf ("%f",&num2);
```

```
if (num2 != 0)
```

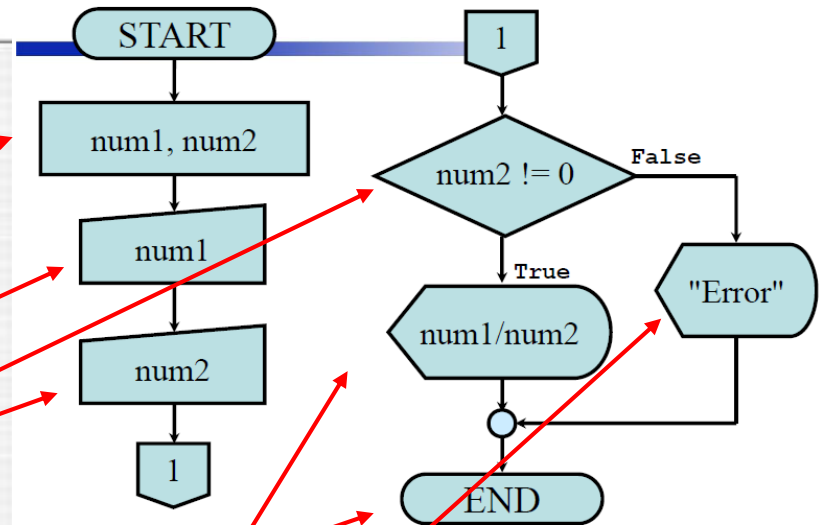
```
    printf (".2f / .2f = .2f", num1, num2, num1/num2);
```

```
else
```

```
    printf ("Error divided by zero\n");
```

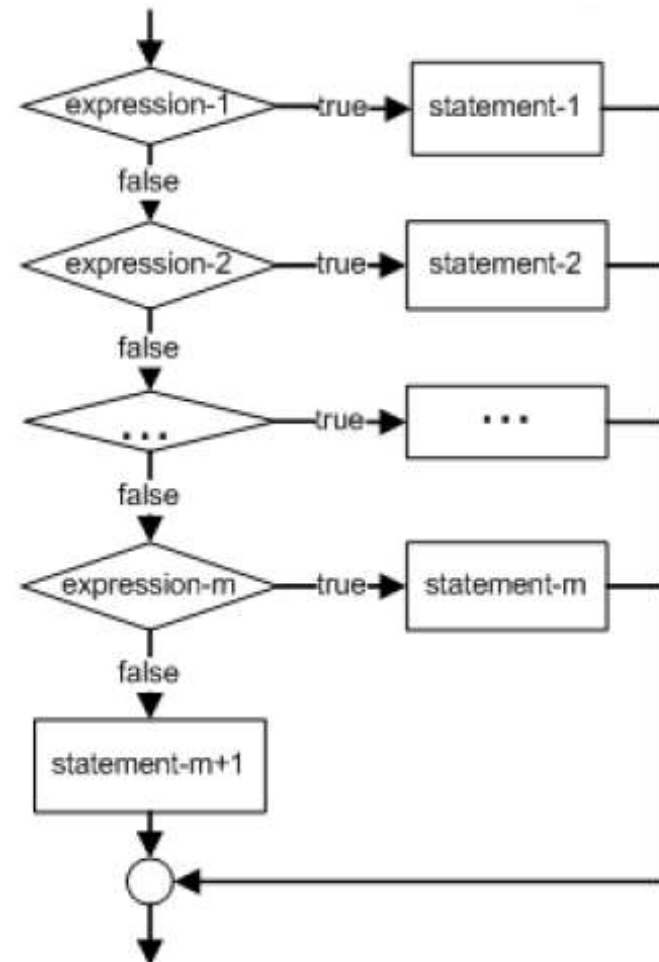
```
return 0;
```

```
}
```



4.4 การใช้คำสั่ง **if-else if** | ภายในเงื่อนไขมีคำสั่งเดียว ลักษณะผูกกัน

```
if (expression-1)  
    statement-1;  
else if (expression-2)  
    statement-2;  
...  
...  
else if (expression-m)  
    statement-m;  
else  
    statement-m+1;
```



การใช้คำสั่ง **if-else if** | ภายในเงื่อนไขมีหลายคำสั่ง

```
if (expression-1)
{
    statement-1.1;
    ...
    statement-1.n;
}
else if (expression-2)
{
    statement-2.1;
    ...
    statement-2.n;
}
...
```

...

```
...
else if (expression-m)
{
    statement-m.1;
    ...
    statement-m.n;
}
else
{
    statement-m+1.1;
    ...
    statement-m+1.n;
}
```

โปรแกรม 4.4 ตรวจสอบเกรด A ถึง F

จงเขียนผังงานและโปรแกรมสำหรับรับชื่อ นามสกุล รหัส
นักศึกษา และคะแนนวิชา Computers and Programming เพื่อ
ตรวจสอบว่านักศึกษาได้เกรดระดับใด โดยใช้เกณฑ์ดังนี้

คะแนน 90 – 100	ได้เกรด A	คะแนน 80 – 89.99	ได้เกรด B
คะแนน 70 – 79.99	ได้เกรด C	คะแนน 60 – 69.99	ได้เกรด D
คะแนน 0 – 59.99	ได้เกรด F		

แล้วแสดงผลลัพธ์ ชื่อ สกุล รหัสนักศึกษา คะแนน และเกรด

โปรแกรมตรวจสอบเกรด | if-else if

- Output Analysis
 - แสดงชื่อ นามสกุล รหัสนักศึกษา คะแนน และเกรดที่ได้
- Input Analysis
 - ชื่อ / นามสกุล / รหัสนักศึกษา / คะแนน
- Process Analysis
 - โปรแกรมรอรับชื่อ / นามสกุล / รหัสนักศึกษา / คะแนนสอบ
 - แสดงผลชื่อ-สกุล รหัสนักศึกษา และคะแนน

โปรแกรมตรวจสอบเกรด | if-else if

- Process Analysis (ต่อ)
 - ตรวจสอบคะแนนสอบ
 - ถ้ามากกว่าหรือเท่ากับ 90 แสดงผลว่าได้เกรด A
 - ถ้ามากกว่าหรือเท่ากับ 80 แสดงผลว่าได้เกรด B
 - ถ้ามากกว่าหรือเท่ากับ 70 แสดงผลว่าได้เกรด C
 - ถ้ามากกว่าหรือเท่ากับ 60 แสดงผลว่าได้เกรด D
 - ถ้าไม่ตรงเงื่อนไขที่ผ่านมาทั้งหมด แสดงผลว่าได้เกรด F

โปรแกรมตรวจสอบเกรด | if-else if

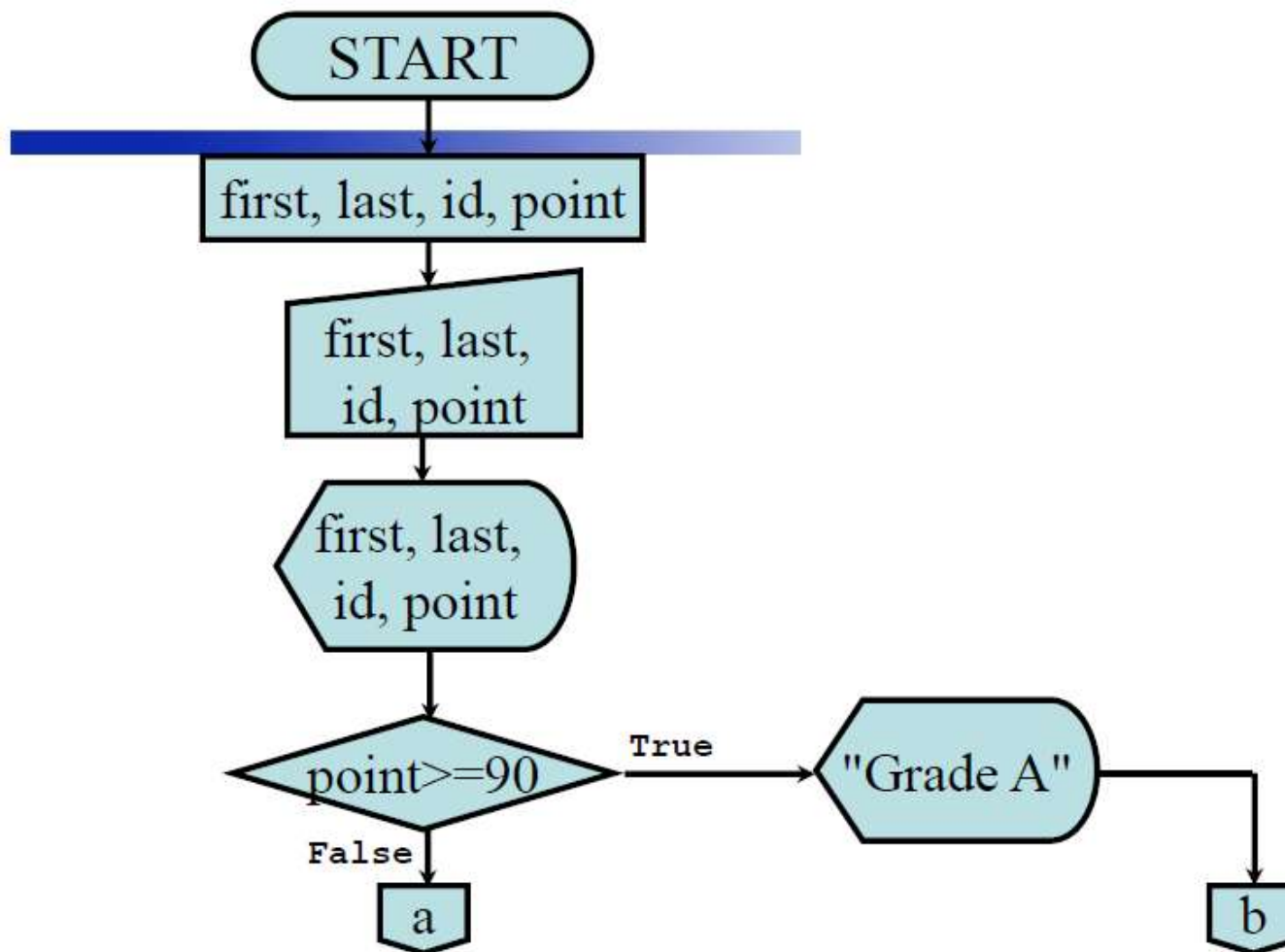
- Variable Define

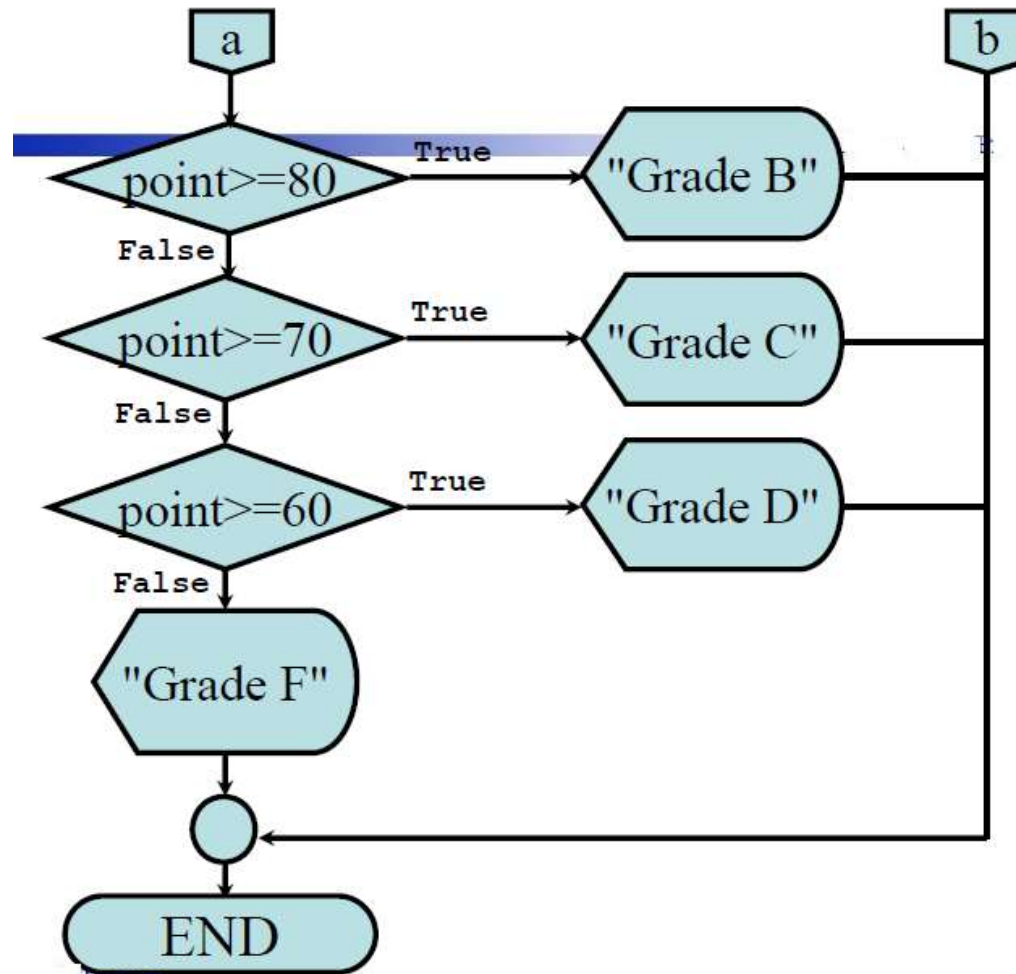
first : ตัวแปรชนิดข้อความสำหรับเก็บชื่อขนาด 20

last : ตัวแปรชนิดข้อความสำหรับเก็บนามสกุลขนาด 20

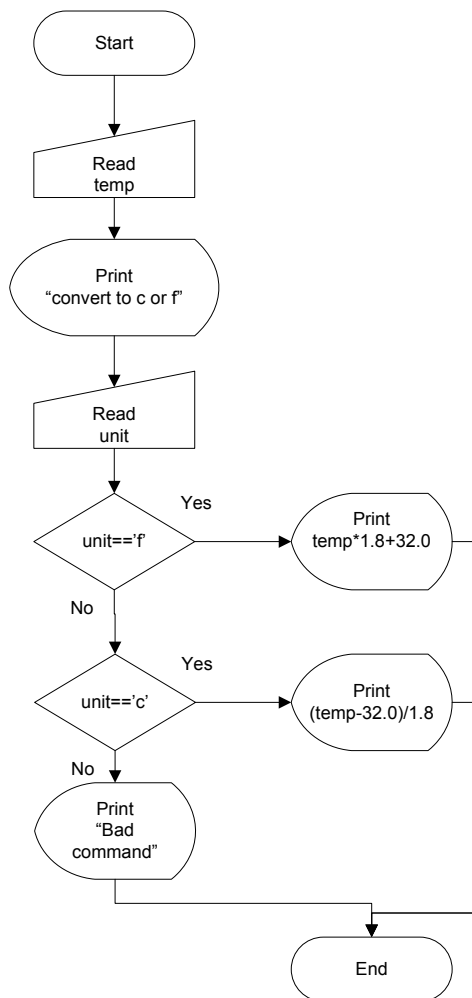
id : ตัวแปรชนิดข้อความสำหรับเก็บรหัสนักศึกษาขนาด 9

point : ตัวแปรชนิดจำนวนเต็มสำหรับเก็บคะแนน

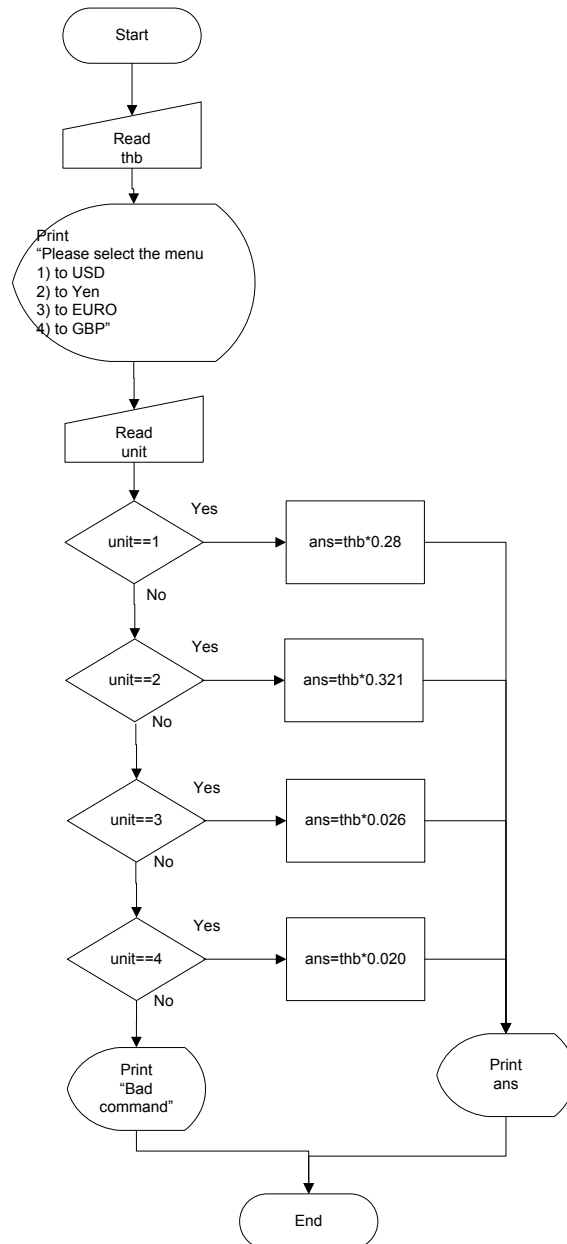




โปรแกรมแปลงหน่วยอุณหภูมิ



โปรแกรมแปลงค่าสกุลเงิน

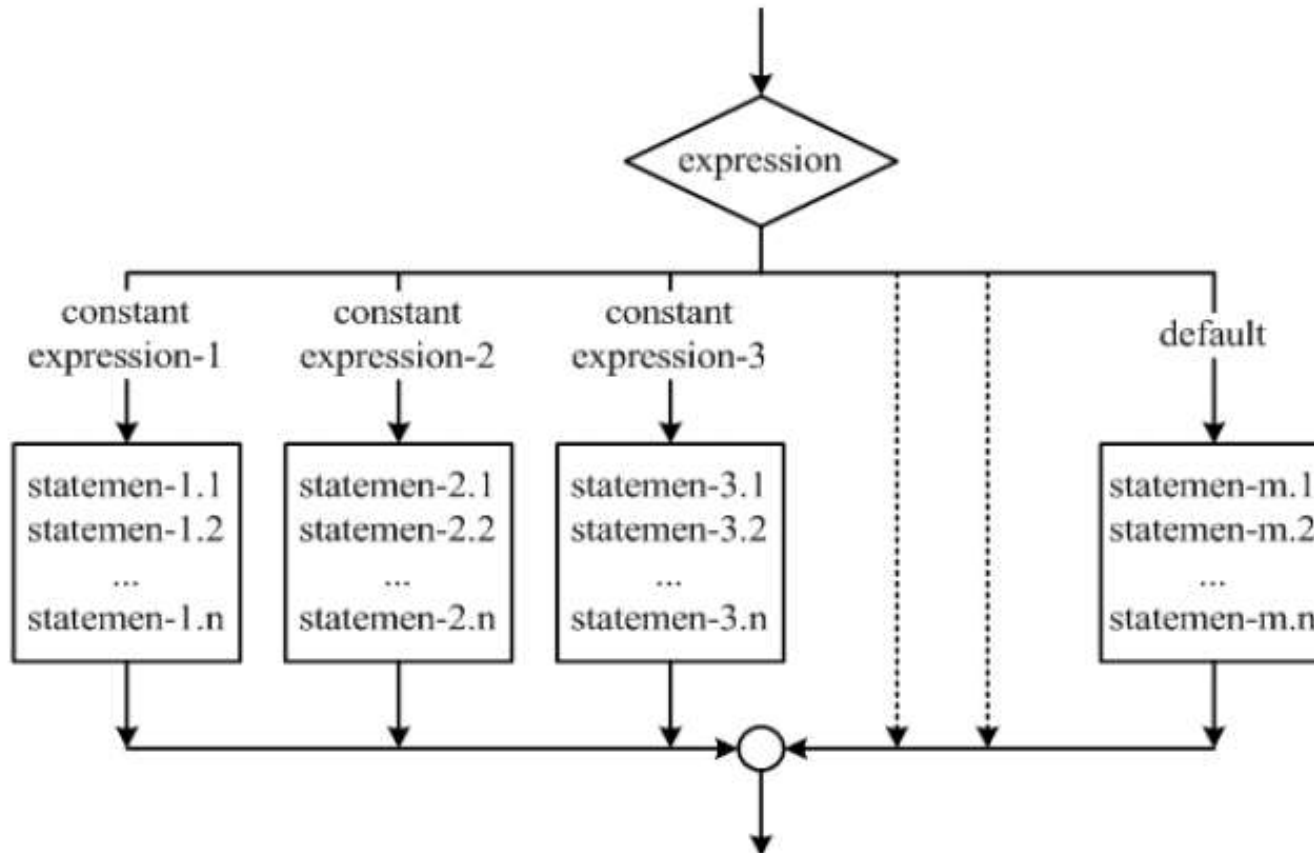


4.4 การใช้คำสั่ง switch

```
switch (expression-1)
{
case constant-expr-1:
    statement-1.1;
    statement-1.2;
    ...
    statement-1.n;
    break;
case constant-expr-2:
    statement-2.1;
    statement-2.2;
    ...
    statement-2.n;
    break;
...
...
}
```

```
...
case constant-expr-m:
    statement-m.1;
    statement-m.2;
    ...
    statement-m.n;
    break;
default:
    statement-d.1;
    statement-d.2;
    ...
    statement-d.n;
}
```

การใช้คำสั่ง switch | Flowchart



ตัวอย่างการใช้คำสั่ง **switch**

```
#include<stdio.h>
int main()
{
    int d;
    printf("Enter a number from 1 to 9: ");
    scanf("%d", &d);
    switch (d)
    {
        case 1: puts("A stitch in time saves nine.");
                break;
        case 2:
        case 6:
        case 9: puts("Handsome is as handsome does.");
                break;
        default: puts("Very clever. Try again.");
    }
    return 0;
}
```


ตัวอย่างการใช้คำสั่ง **switch**

```
#include<stdio.h>
int main()
{
    int d;
    printf("Enter a number from 1 to 9: ");
    scanf("%d", &d);
    switch (d)
    {
        case 1: puts("A stitch in time saves nine.");
                break;
        case 2: break;
        case 6: break;
        case 9: puts("Handsome is as handsome does.");
                break;
        default: puts("Very clever. Try again.");
    }
    return 0;
}
```


โจทย์ : โปรแกรมตรวจสอบสถานะการเรียน

จงเขียนผังงาน และโปรแกรมรับค่า GPA ของภาคการศึกษาที่ผ่านมาแล้วแสดงผลสถานะการเรียน โดยมีเงื่อนไขดังนี้

- $GPA_{(เดิม)} < 1.00$ แสดงสถานะ Retried (ฟื้นสภาพ) จบโปรแกรม
- $GPA_{(เดิม)} < 2.00$ แสดงสถานะ Probation (ภาคทัณฑ์)
- $GPA_{(เดิม)} \geq 2.00$ แสดงสถานะ Normal (ปกติ)

หลังจากแสดงผลสถานะแล้วให้โปรแกรมรับค่า GPS และ GPA ของภาคการศึกษาปัจจุบัน แล้วแสดงผลสถานะการเรียน

โจทย์ : โปรแกรมตรวจสอบสถานะการเรียน

โดยมีเงื่อนไขดังนี้

- $GPA_{(ใหม่)} < 1.00$ แสดงสถานะ **Retried**
- $GPA_{(ใหม่)} \geq 2.00$ แสดงสถานะ **Normal**
- $GPA_{(เดิม)} < 2.00$
 - $GPS_{(ใหม่)} \geq 2.00$ แสดงสถานะ **Probation**
 - $GPS_{(ใหม่)} < 2.00$ แสดงสถานะ **Retried**

โจทย์ : โปรแกรมตรวจสอบสถานะการเรียน

– ตัวอย่างการรันโปรแกรม

```
Enter GPA (past) : 0.74  
Your Status : Retired  
Sorry, Try again.
```

```
Enter GPA (past) : 1.50  
Your Status : Probation  
Enter GPS (present) : 2.00  
Enter GPA (present) : 1.75  
Your Status : Probation
```

```
Enter GPA (past) : 1.50  
Your Status : Probation  
Enter GPS (present) : 3.0  
Enter GPA (present) : 2.25  
Your Status : Normal
```

```
Enter GPA (past) : 2.50  
Your Status : Normal  
Enter GPS (present) : 1.00  
Enter GPA (present) : 1.75  
Your Status : Probation
```

```
Enter GPA (past) : 1.50  
Your Status : Probation  
Enter GPS (present) : 1.00  
Enter GPA (present) : 1.25  
Your Status : Retired  
Sorry, Try agin.
```

4.5 คำถามท้ายบท

1. จงเขียนผังงาน โปรแกรมเครื่องคิดเลขที่มีตัวอย่างผลการรันดังต่อไปนี้

```
Enter Num1 : 3
Enter Num2 : 6
Calculator Menu :
  1. +
  2. -
  3. *
  4. /
  5. %
Choose menu : 1
Ans: Num1 + Num2 = 9
```

คำถามท้ายบท (ต่อ)

2. จงเขียนโปรแกรมเพื่อแยกสาร 5 ชนิดซึ่งแต่ละชนิดมีคุณสมบัติดังต่อไปนี้

ชนิดที่ 1 มีคาร์บอนเป็นองค์ประกอบ, มี 5 อะตอม, เป็นก๊าซ

ชนิดที่ 2 มีคาร์บอนเป็นองค์ประกอบ, มี 6 อะตอม, เป็นของเหลว

ชนิดที่ 3 มีไนโตรเจนเป็นองค์ประกอบ, มี 6 อะตอม, เป็นก๊าซ

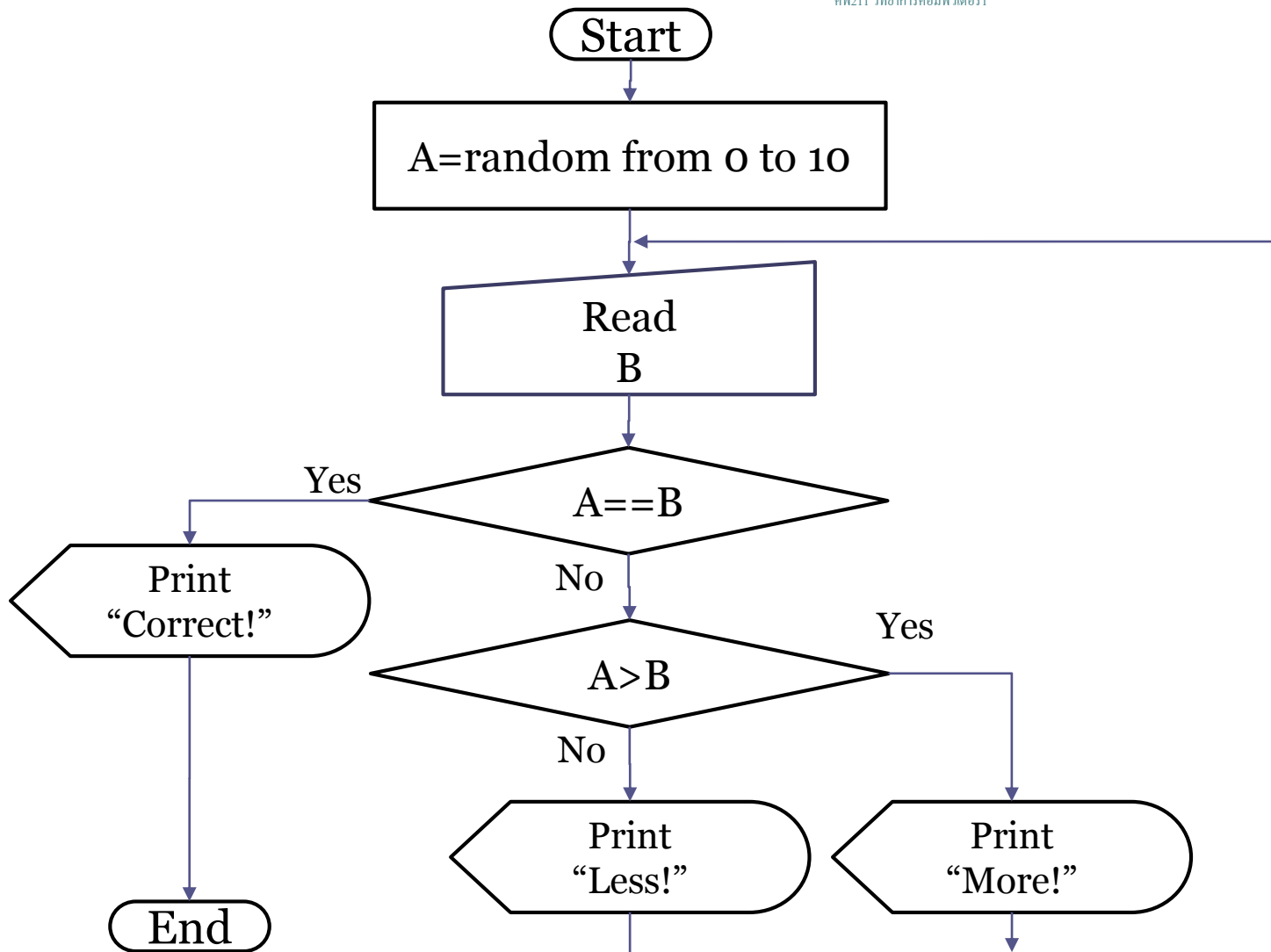
ชนิดที่ 4 มีไนโตรเจนเป็นองค์ประกอบ, มี 4 อะตอม, เป็นของแข็ง

ชนิดที่ 5 เป็นสารชนิดที่ 1 และมี ไฮโดรเจนเป็นองค์ประกอบ

คำถามท้ายบท (ต่อ)

3. มีส่วนของคำสั่ง switch อยู่ให้หาค่า x , y และ z หลังจากผ่านส่วนของคำสั่ง switch นี้ โดยกำหนดให้ $x = 1$, $y = 0$ และ $z = 0$

```
switch (x%2)
{
    case 0    : x = 2; y = 3;
    case 1    : x = 4; break;
    default   : y = 3; x = z;
}
```



ซาลามัต ดาตัง หมายถึง สวัสดีในภาษามลายูเซีย

