

CP 111 Computer Programming (2/2554 Section B01-B02)

Handout 2^{*}

หัวข้อ

- ทบทวน
- Operators in C (Chapter 12.3)
- Problem solving using operators (Chapter 12.5)
- Conditional constructs (Chapter 13.2)

ทบทวน

1. ภาษา C เป็นภาษาโปรแกรมคอมพิวเตอร์แบบ Procedural (Procedural programming language)
 - โปรแกรมภาษา C หนึ่งๆ จะประกอบไปด้วยหลายๆ โปรซีเยอร์ (หรือ Subroutine)
 - ภาษา C จะเรียก โปรซีเยอร์ (หรือ Subroutine) ว่า Function
2. โครงสร้างโปรแกรมภาษา C เบื้องต้น: Variables, Literals, Operators, Expression, Statement, printf, scanf
 - **Variables:** ใช้สำหรับเก็บค่าในการประมวลผลของโปรแกรม การอ้างถึงตัวแปร (อ่าน/เขียนค่า) ทำได้โดยอ้างชื่อ
 - รูปแบบ (syntax) การประกาศตัวแปร: `<data_type> <var_name>;`
 - Three basic data types: int, float (double), char
 - Type modifiers: unsigned, signed, short, long
 - **Literals:** ข้อความที่มีค่าอยู่ในตัว เช่น 87, 0x57, 1.345, 454.12e-10, 40.0e12, 'c'
 - **Operators:** ใช้สำหรับระบุการประมวลผล (manipulate) ค่าของตัวแปร เช่น + - * / % = > >= < <= != == || && !
 - **Expressions:** ข้อความที่ประกอบไปด้วย variables และ literals และ operators และต้องสามารถคำนวณค่า (evaluate) ออกมาได้ เช่น x*y, x>2, x+2
 - **Statements:** ข้อความที่ประกอบด้วย variables และ literals และ operators เพื่อแสดงสิ่งที่สั่งให้คอมพิวเตอร์ทำ โดยทั่วไป 1 unit of work ต่อ 1 statement และจบท้ายด้วย ; (semicolon) เช่น `c = (b/d) + (a*d);`
3. `printf("Hello %d %f %c\n", an_int_var, a_float_or_double, a_char);`
4. `scanf("%d %c %f %lf", &an_int, &a_char, &a_float, &a_double);`
5. เพิ่มเติม printf: `printf("%f\n", a); /* Display 1.144691 %f จำนวนจุดทศนิยมที่จะแสดงขึ้นอยู่กับ compiler */`

ในกรณีที่ต้องการใช้ printf แสดงค่าแบบ floating point โดยที่จะระบุจำนวนจุดทศนิยมที่จะแสดง

```
double a=1.144961;
/* %(ตัวเลขระบุจำนวนจุด)f */
printf("%.1f\n",a); /* Display 1.1 */
printf("%.2f\n",a); /* Display 1.14 */
printf("%.3f\n",a); /* Display 1.145 */
```

```
double a=41.144961;
/* %(ตัวเลขระบุจำนวนตัวอักษรทั้งหมด).(ตัวเลขระบุจำนวนจุด)f */
printf("%.1f\n",a); /* Display xx41.1 */
printf("%.2f\n",a); /* Display x41.14 */
printf("%.3f\n",a); /* Display 41.145 */
```

Note: x = ช่องว่าง (space)

^{*} Handout นี้ เรียงเรียงจาก Slide วิชา ECE190 UIUC by V.Kindratenko (<https://wiki.engr.illinois.edu/display/ece190/Lectures>)

Operators in C

6. สิ่งที่สำคัญที่จำเป็นต้องรู้เกี่ยวกับ Operator หนึ่งๆ

- หน้าที่ของ Operator ว่ามันเอาไว้ทำอะไร

- Order of evaluation (ลำดับการคำนวณค่า)

- (i) Expression ในวงเล็บจะถูกคำนวณค่าก่อน เริ่มจากวงเล็บในสุดไปนอกสุด เช่น $(a*b) + ((c*d)/4)$

- (ii) Precedence: ลำดับการทำงานของ operator ที่อยู่ด้วยกันเป็นอย่างไร

- เช่น $a*b + c*d$ จะเหมือนกับ $(a*b) + (c*d)$ นั่นคือ operator $*$ จะมีลำดับการทำงานก่อน $+$

- (iii) Associativity: ลำดับการทำงานของ operator ที่มี Precedence ระดับเดียวกันมาอยู่ด้วยกันเป็นอย่างไร

- เช่น $a - b - c$ จะเหมือนกับ $(a - b) - c$

7. Assignment operator (=): ใช้กำหนดค่าด้านขวาของ operator ให้กับตัวแปรที่ระบุทางด้านซ้าย

- เช่น $a=b+c;$

8. Arithmetic operator ($*$ / $\%$ + $-$): ดู Slide 12-15 สำหรับ Precedence และ Assoc.

- Precedence: ลำดับการทำงานของ operator $*$ / $\%$ จะมาก่อน $+$ -

- เช่น Expression $a+b*c$ จะเทียบเท่ากับ $a+(b*c)$ เนื่องจาก operator $*$ จะถูกคำนวณค่าก่อน $+$

- Associativity: ลำดับการทำงานของ operator ที่มี Precedence ระดับเดียวกันมาอยู่ด้วยกันจะถูกกระทำจากซ้ายไปขวา

- เช่น $a+b-c+d$ จะเทียบเท่ากับ $((a+b)-c)+d$

9. ข้อควรระวัง

- กรณีที่ใช้ operator กับ mixed data type จะปรับ data type ให้เป็นชนิดที่รองรับค่าได้มากกว่า เช่น ถ้า x เป็น `int` แล้วใน expression $x+4.3$ ค่าของ x จะถูกแปลงเป็น `double` แล้วบวกกับ 4.3 ได้ผลลัพธ์เป็นชนิดข้อมูลแบบ `double`

- Integer division เช่น $5/9$ มีค่าเท่ากับ 0 หรือถ้า x เป็น `int` และมีค่าปัจจุบันเท่ากับ 5 แล้ว $x/3$ จะเท่ากับ 1

- กรณีที่ Data type ของ Expression ด้านขวาของ operator ไม่เหมือนกับ Data type ของ Variable ด้านซ้าย

ตัวอย่าง

```
int main()  
{  
    int x,a;  
    double y,z;
```

```
    x=1;
```

```
    y=2.3;
```

```
    z=x+y;
```

```
    /* copy ของค่า x จะถูกแปลงให้เป็นชนิด double (ตัวแปร x ยังเป็นชนิด int และไม่ได้เปลี่ยนค่า)
```

```
    ก่อนที่ expression x+y จะถูกคำนวณค่า */
```

```
    printf("%f", z);
```

```
    a=x+y;
```

```
    /* ค่าของ expression x+y ซึ่งมีชนิดข้อมูลแบบ double จะถูกแปลงเป็นชนิดข้อมูลแบบ int ก่อนเก็บลงใน a */
```

```
    /* โดยทั่วไป compiler จะแปลงเป็น machine code ที่ตัดเลขวหลังทศนิยมทิ้ง */
```

```
    /* เกิด Bug บ่อย: Programmer ควรจะต้องระวังตรงจุดนี้ด้วย */
```

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

```
}
```

- ควรใช้วงเล็บปิดเปิดในการจัดกลุ่ม operator ใน Expression และ statement เพื่อไม่ให้เกิดความสับสน

10. Increment/Decrement Operators: ++, --

- โดยทั่วไป x++ จะเทียบเท่ากับการเขียน x=x+1 และ y-- จะเทียบเท่ากับการเขียน y=y-1
- สามารถใช้ในรูปแบบ preincrement., postincrement, predecrement, post decrement ดู Slide 12-21

ตัวอย่าง

```
{
    x=4;
    y=x++; /* x++ is in postincrement form */
    /*Note: x++ is executed after y=... then y=4, x=5 */

    z=++x; /* ++x is in preincrement form */
    /*Note: ++x is executed before z=... then z=6, x=6 */
}
```

11. Relational Operators

- ได้แก่ > (มากกว่า), >= (มากกว่าหรือเท่ากับ), <, <=, != (ไม่เท่ากับ), == (เท่ากับ)
- ใช้สร้าง Expression ที่มีการเปรียบเทียบ โดยที่ผลลัพธ์ของการเปรียบเทียบ (ค่าของ Expression) จะมีค่า 1 หรือค่าไม่เท่ากับ 0 สำหรับ Expression ที่เป็นจริง (TRUE) หรือค่า 0 สำหรับ Expression ที่เป็นเท็จ (False)
- Note: operator == คือการเปรียบเทียบว่าเท่ากันหรือไม่ แต่ operator = คือ Assignment operator
- ตัวอย่าง

```
q=(111==121); /*q มีค่า 0 เนื่องจาก expression (111==121) มีค่าเป็น False เสมอ */
z=(x==y);      /*z มีค่า 1 เมื่อค่าของ x เท่ากับค่าของ y มิฉะนั้น z มีค่า 0 */
h=(f<=g);      /* h มีค่า 1 เมื่อค่าของ f น้อยกว่าหรือเท่ากับค่าของ g มิฉะนั้น h มีค่า 0 */
```

12. Logical Operators

- ได้แก่ || (OR), && (AND), ! (NOT)
- ใช้สร้าง Expression สำหรับเป็นเงื่อนไขในการเปรียบเทียบต่างๆ
- มอง variables หรือ expression ที่ต้องอยู่กับ operator ให้มีค่า 2 แบบ TRUE (1 or non-zero) หรือ FALSE (0)
- ตัวอย่าง

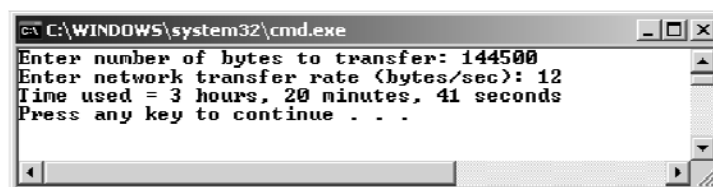
```
y=(10<=x) && (x<=20); /*y มีค่า TRUE ถ้า 10<=x<=20*/
z=(10<=x) || (x<=5); /*z มีค่า TRUE เมื่อค่าของ x >= 10 หรือ x<=5*/
h=(10<=x) && !(x%2==0); /* h มีค่า TRUE เมื่อค่า x >= 10 และ x%2==0 เป็นเท็จ (หารด้วย 2 ไม่ลงตัว)
```

13. Precedence และ Associativity ของ Operator ทั้งหมดในภาษา C

- ดู Table 12.5 หน้า 323 ของ Textbook สำหรับ

Problem solving using operators

- 14 จงทำโปรแกรมข้างล่างให้สมบูรณ์เพื่อให้โปรแกรมนี้อำนาจการคำนวณเวลาที่ต้องใช้ (จำนวนชั่วโมง, นาที, วินาที) ในการส่งข้อมูลปริมาณขนาดหนึ่ง (ระบุโดยจำนวนไบต์) ผ่าน Network ที่มีความเร็วที่กำหนด (ระบุเป็นค่าจำนวนไบต์/วินาที)



```

#include <stdio.h>
int main()
{
    int numbytes,transf_rate,time,hr,min,sec;

    printf("Enter number of bytes to transfer: ");
    .....

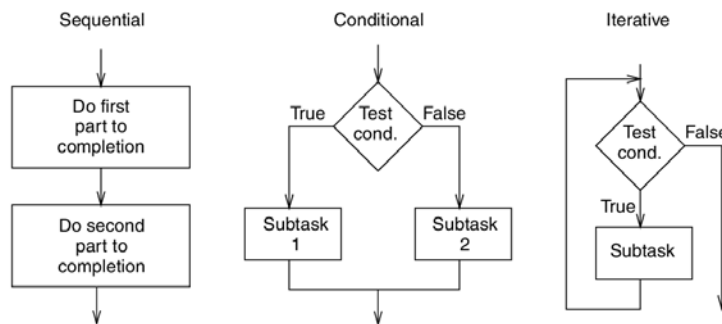
    printf("Enter network transfer rate (bytes/sec): ");
    .....

    time=
    hr=
    min=
    sec=
    printf("Time used =
}

```

Conditional constructs

15. Recall: ใน CP121 เราศึกษาการเขียนโปรแกรมภาษาแอสเซมบลี LC-3 โดยมี 3 constructs ที่ได้เรียนดังรูปข้างล่าง



ใน CP111 เราจะศึกษาการเขียนโปรแกรมทั้ง 3 รูปแบบด้วยภาษา C

- Sequential: เขียน statement เรียงต่อกันจากบนลงล่าง
- Conditional: if statement, if-else statement, switch statement
- Iterative: while statement, do-while statement, for statement (Section 13.3)

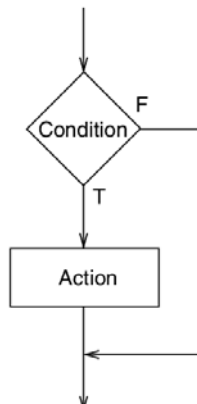
16. if statement

C syntax

```

if(condition)
    action;

```



/* ถ้า condition (expression) มีค่าเท่ากับ TRUE (ไม่เท่ากับ 0)
โปรแกรมจะกระทำ action (statement)
Note: action สามารถเป็น compound statements อยู่ใน { }
*/

ตัวอย่าง

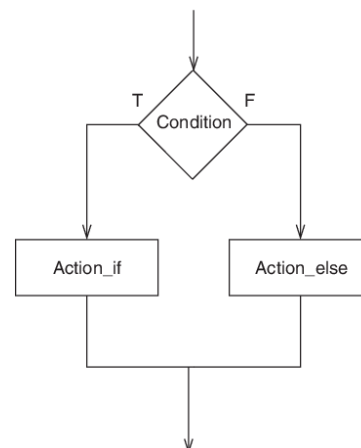
<pre>if(x<0) x=-x;</pre>	<pre>; compound statements if(x>5 && x < 25) { y=x*x+5; z=(2*y)/3; printf("%d %d\n",y,z); }</pre>	ระวัง ถ้าไม่ใส่วงเล็บ { } <pre>if(x>5 && x < 25) y=x*x+5; z=(2*y)/3; printf("%d %d\n",y,z);</pre>
---------------------------------	---	--

17. if-else statement

C syntax

```
if(condition)
    action_when_condition_is_true;
else
    action_when_condition_is_false;
```

ถ้า condition (expression) มีค่าเท่ากับ TRUE (ไม่เท่ากับ 0) โปรแกรมจะกระทำ
 action_when_condition_is_true
 มิฉะนั้น โปรแกรมจะกระทำ
 action_when_condition_is_false



ตัวอย่าง

<pre>if(x<0) x=-x; else x=2*x; ; compound statements if(x>5 && x < 25) { y=x*x+5; printf("%d\n",y); } else { printf("%d\n",x); }</pre>	<u>มี if statement ซ้อนเข้าไปได้</u> <pre>if(x!=0) { if(y>3) z=z/2; else z=z+2; }</pre>	<u>มี if-else หลายอัน</u> <pre>if(x>1) z=z+1; else if(x<0) z=z-1; else if(y==1) z=z*2; else z=z/2;</pre>
--	---	---

18. โจทย์ตัวอย่าง (ข้อสอบเก่าปี 2552):

กำหนดให้ a และ b เป็นตัวแปรแบบ int จงเขียน C expression เพื่อใช้เป็นเงื่อนไขในการทดสอบ a และ b ตามข้อความภาษาไทยดังต่อไปนี้

(i) a เป็นเลขคู่ และ b เป็นเลขคี่ แต่ a ต้องหารด้วย b ลงตัว

.....

จงยกตัวอย่างค่าที่เป็นไปได้ของ a และ b มา 1 คู่ที่ทำ expression ข้างบนมีค่าไม่เท่ากับ 0 (non-zero value):

(ii) a หรือ b ต้องเป็นค่าที่มีเครื่องหมายไม่เหมือนกัน (a เป็นค่าลบ b ต้องเป็นค่าบวก หรือในทางกลับกัน) และ ผลรวมของ a และ b มีค่ามากกว่า 0

จงยกตัวอย่างค่าที่เป็นไปได้ของ a และ b มา 1 คู่ที่ทำ expression ข้างบนมีค่าไม่เท่ากับ 0 (non-zero value)

(iii) a ต้องเป็นค่าที่สามารถหารด้วย 17 หรือ 19 ลงตัว และ b ต้องเป็นค่าที่หารด้วย 11 แล้วเหลือเศษ 3

จงยกตัวอย่างค่าที่เป็นไปได้ของ a และ b มา 1 คู่ที่ทำ expression ข้างบนมีค่าเท่ากับ 0 (zero value)

19. โจทย์ตัวอย่าง: จงแสดงผลลัพธ์ของโปรแกรมเมื่อผู้ใช้ป้อนตัวเลข 2 ตัวเป็นค่าต่างๆ

```
#include <stdio.h>
int main()
{
    int x,y,a,b;
    scanf("%d %d",&x,&y);
    if(x>2*y)
    {
        a=x+y;
        b=x/y;
    }
    else
    {
        a=x*y;
        b=x-y;
    }
    printf("%d %d\n",a,b);
    return 0;
}
```

(ถ้าผู้ใช้ป้อน 5 4-)

(ถ้าผู้ใช้ป้อน 8 4-)

(ถ้าผู้ใช้ป้อน 10 4-)

20. โจทย์ตัวอย่าง: จงเขียนโปรแกรมรับข้อมูลแบบจำนวนเต็มบวกผ่านทางคีย์บอร์ด หลังจากนั้นให้ตรวจสอบว่าตัวเลขนั้นเป็นที่หารด้วย 5 หรือ 7 ลงตัว แต่หารด้วย 11 แล้วเหลือเศษ 3 ใช่หรือไม่ ถ้าใช่ให้แสดงข้อความ YES ออกสู่หน้าจอ ถ้าไม่ใช่ให้แสดง NO

21. โจทย์ตัวอย่าง: จงเขียนโปรแกรมเพื่อตรวจสอบว่าจุดในระนาบ 2 มิติอยู่ใน Quadrant อะไร หรืออยู่บน แกน X, Y หรือเป็นจุด origin โดยโปรแกรมจะรับค่าตัวเลขแบบจำนวนเต็มจากผู้ใส่ 2 ค่าเว้นด้วยช่องว่างในบรรทัดเดียวกัน ซึ่งแทนค่า X และ Y coordinate ของจุดหนึ่งๆ หลังจากนั้นโปรแกรมต้องแสดงผลเป็นข้อความ Q1 หรือ Q2 หรือ Q3 หรือ Q4 หรือ X-AXIS หรือ Y-AXIS หรือ ORIGIN ออกไปสู่หน้าจอ ตามค่า XY-coordinate ที่ถูกป้อนเข้ามา

22. โจทย์ตัวอย่าง: จงเขียนโปรแกรมเพื่อรับข้อมูลแบบจำนวนเต็ม 3 จำนวนที่ละบรรทัด หลังจากนั้นให้แสดงค่าน้อยสุด (Minimum value) และค่ามากที่สุด (Maximum value) ในบรรดา 3 จำนวนนี้ แล้วแสดงออกสู่หน้าจอที่ละบรรทัด

23. โจทย์ตัวอย่าง: ถ้าลูกบอลลูกหนึ่งถูกปล่อยที่ความสูง h เมตร ความเร็ว u ในการตกกระทบพื้นจะมีค่าเท่ากับ

$u = \sqrt{2gh}$ เมตร/วินาที โดยที่ $g=9.8$ เมตร/วินาที² และลูกบอลจะเร่งขึ้นไปอยู่ที่ระดับความสูง $\frac{2}{3}$ เท่าของความสูงของการตกครั้งที่แล้ว จงเขียนโปรแกรมที่รับค่าความสูงที่จะเริ่มปล่อยลูกบอล หลังจากนั้นให้แสดงค่าความเร็วในการตกกระทบพื้นยกกำลังสอง (u^2) และความสูงที่ลูกบอลจะเร่งขึ้นไปของการตกและเร่งขึ้นของลูกบอลสามครั้งแรก กำหนดให้โปรแกรมทำการแสดงค่าความเร็วยกกำลังสอง (u^2) และความสูงของการตกและเร่งขึ้นในแต่ละครั้งคู่กันที่ละบรรทัดและให้แสดงเป็นเลขทศนิยม 3 หลัก