

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

Handout 3*

หัวข้อ

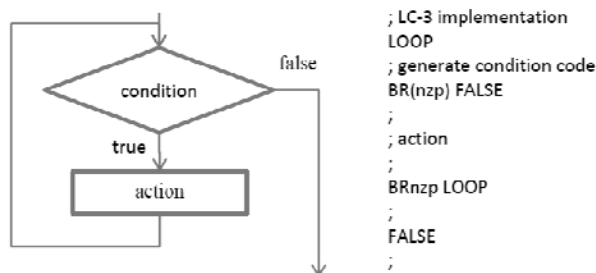
- ทบทวน
- Iterative constructs (Chapter 13.3)
- Problem solving using control structures (Chapter 13.4)
- switch, break, and continue statements (Chapter 13.5)

ทบทวน

- Buzzwords: variables, literals, operators, expression, statement
- Conditional statements: if statement, if-else statement

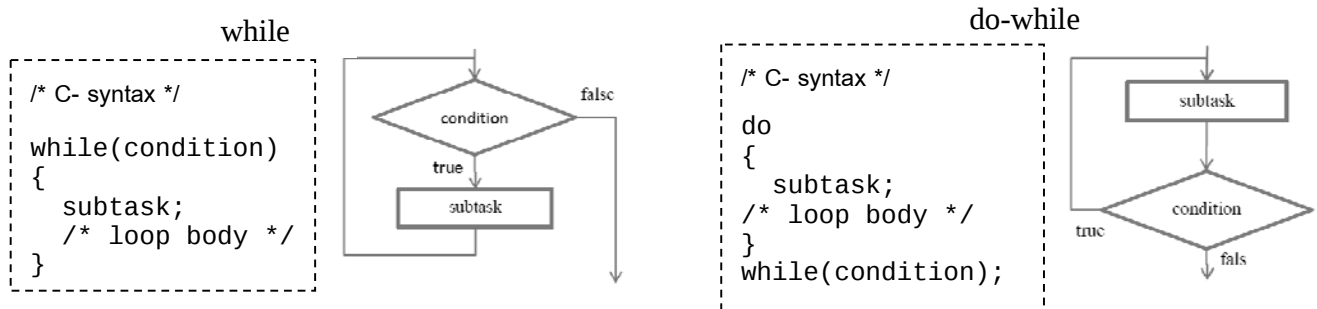
Iterative constructs และ switch, break, continue statements

1. ใน CP121 เราเรียนการเขียน LC-3 assembly สำหรับสร้าง iterative construct ดังตัวอย่างข้างล่าง



ใน CP111 เราจะศึกษาการสร้าง iterative construct ในรูปแบบโปรแกรมภาษา C ซึ่งมี 3 รูปแบบได้แก่ while, do-while, for statements

2. while statement และ do-while statement



- รูปแบบ while จะกระทำ subtask ก็ต่อเมื่อ condition เป็นจริง (อาจจะกระทำหรือไม่กระทำ subtask ก็ได้)
- รูปแบบ do-while จะกระทำ subtask ก่อนแล้วจึงมาตรวจสอบว่า condition เป็นจริงหรือไม่ (กระทำ subtask อย่างน้อย 1 ครั้งแน่ๆ)

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

- ตัวอย่าง จังหวะการทำงานของโปรแกรมและผลลัพธ์ที่แสดงที่หน้าจอ

Program A: while	Program B: do-while
<pre>#include <stdio.h> int main() { int x,y; x=0; /* A1 */ while(x<3) /* A2 */ { printf("%d ",x); /* A3 */ x=x+1; /* A4 */ } y=x; /* A5 */ printf("%d ",y); /* A6 */ return 0; /* A7 */ }</pre>	<pre>#include <stdio.h> int main() { int x,y; x=0; /* B1 */ do { printf("%d ",x); /* B2 */ x=x+1; /* B3 */ } while(x<3); /* B4 */ y=x; /* B5 */ printf("%d ",y); /* B6 */ return 0; /* B7 */ }</pre>

(i) ลำดับการทำงานของบรรทัดต่างๆ ของโปรแกรม A และผลลัพธ์ที่หน้าจอคือ

.....

(ii) ลำดับการทำงานของบรรทัดต่างๆ ของโปรแกรม B และผลลัพธ์ที่หน้าจอคือ

.....

(iii) ถ้าเปลี่ยน code ในบรรทัด A1 ของโปรแกรม A และบรรทัด B1 ของโปรแกรม B จาก x=0; เป็น x=3; จังหวะลำดับการทำงานและผลลัพธ์ที่หน้าจอของโปรแกรมทั้งสอง

.....

.....

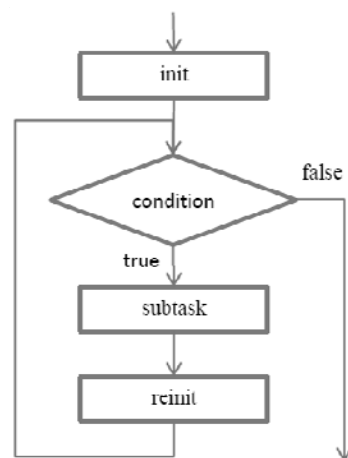
- ข้อควรระวัง: Infinite loop ส่วนย่อยของโปรแกรมข้างล่างลืมเพิ่มค่า x โปรแกรมจะทำงานไม่รู้จบ โดยพิมพ์ตัวเลข 0 ไปเรื่อยๆ

```
x=0; /* A1 */
while(x<3) /* A2 */
{
  printf("%d ",x); /* A3 */
}
```

3. for statement

```
/* C- syntax */

for( init ; condition ; reinit )
{
  subtask; /* loop body */
}
```

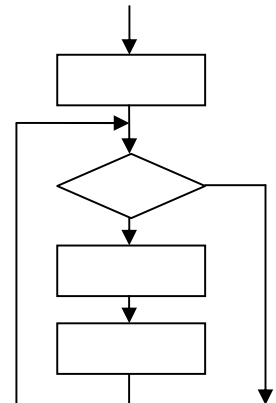


- นิยมใช้สำหรับสั่งให้โปรแกรมกระทำ subtask เป็นจำนวนครั้งที่กำหนดโดยเงื่อนไขใน condition

- ตัวอย่าง จังหวะการทำงานของโปรแกรมและผลลัพธ์ที่แสดงที่หน้าจอ

Flowchart (เฉพาะส่วน for)

```
#include <stdio.h>
int main()
{
    int x,y;
    x=10;                /* A1 */
    for( x=0 ; x<3 ; x++ ) /* A2_1 ; A2_2 ; A2_3 */
    {                    /* Note: x++ is the same as x=x+1 */
        printf("%d ",x); /* A3 */
    }
    y=x;                 /* A4 */
    printf("%d ",y);     /* A5 */
    return 0;           /* A6 */
}
```



4. break and continue statements

- break เป็นการสั่งให้หยุดการทำงานในลูป
- continue เป็นการสั่งให้ข้ามการทำงานที่เหลือในลูป แล้วเริ่มต้น iteration ถัดไปของลูป
- ตัวอย่าง จังหวะการทำงานของโปรแกรมและผลลัพธ์ที่แสดงที่หน้าจอ

Program A: break	Program B: continue
<pre>#include <stdio.h> int main() { int x,y; x=0; /* A1 */ while(x<5) /* A2 */ { if(x==2) /* A3 */ break; /* A4 */ printf("%d ",x); /* A5 */ x=x+1; /* A6 */ } y=x; /* A7 */ printf("%d ",y); /* A8 */ return 0; /* A9 */ }</pre>	<pre>#include <stdio.h> int main() { int x,y; x=0; /* B1 */ while(x<5) /* B2 */ { if(x==2) /* B3 */ { x=x+2; /* B4 */ continue; /* B5 */ } printf("%d ",x); /* B6 */ x=x+1; /* B7 */ } y=x; /* B8 */ printf("%d ",y); /* B9 */ return 0; /* B10 */ }</pre>

(i) ลำดับการทำงานของบรรทัดต่างๆ ของโปรแกรม A และผลลัพธ์ที่หน้าจอคือ

(ii) ลำดับการทำงานของบรรทัดต่างๆ ของโปรแกรม B และผลลัพธ์ที่หน้าจอคือ

- ในกรณี loop for นั้น continue จะทำให้ไปทำที่ reinit -> ตรวจสอบ condition ->)

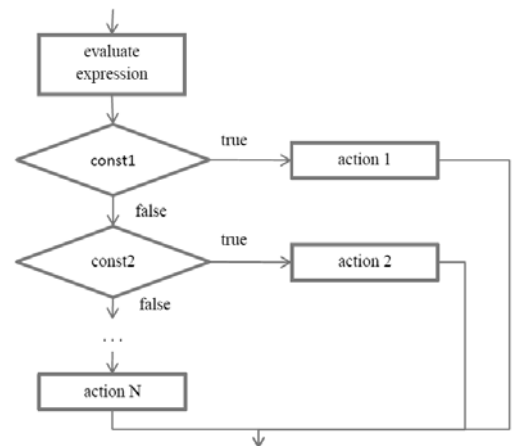
ตัวอย่าง จงระบุการทำงานของส่วนย่อยของโปรแกรมและผลลัพธ์ที่แสดงที่หน้าจอ

<pre> x=10; /* A1 */ for(x=0;x<5;x++) /* A2 */ { if(x==2) /* A3 */ break; /* A4 */ printf("%d ",x); /* A5 */ } y=x; /* A6 */ printf("%d ",y); /* A7 */ </pre>	<pre> x=10; /* B1 */ for(x=0;x<5;x++) /* B2 */ { if(x==2) /* B3 */ continue; /* B4 */ printf("%d ",x); /* B5 */ } y=x; /* B6 */ printf("%d ",y); /* B7 */ </pre>
---	--

5. switch statement

- ใช้เขียนแทนกรณี if-else ต่อกันยาว และ expression คำนวณค่าออกมาได้เป็นค่าแบบ integer (int, char)

เขียนโดยใช้ if-else ต่อกัน	เขียนโดยใช้ switch
<pre> if (expression == const1) action1; else if (expression == const2) action2; else if (expression == const3) action3; ... else actionN; </pre>	<pre> switch (expression) { case const1: action1; break; case const2: action2; break; case const3: action3; break; ... default: actionN; } </pre>



- ตัวอย่างแปลงโปรแกรมที่เขียนโดยใช้ if-else มาอยู่ในรูป switch

<pre> #include <stdio.h> int main() { int a,b,c; char o; scanf("%d %c %d",&a,&o,&b); if(o=='+') c=a+b; else if(o=='-') c=a-b; else if(o=='*') c=a*b; else if(o=='/') c=a/b; else c=a%b; /* Display result */ printf("%d\n",c); return 0; } </pre>	<pre> int main() { int a,b,c; char o; scanf("%d %c %d",&a,&o,&b); switch(o) { case '+': c=a+b; break; case '-': c=a-b; break; case '*': c=a*b; break; case '/': c=a/b; break; default: c=a%b; } /* Display result */ printf("%d\n",c); return 0; } </pre>
---	---

- ข้อควรระวังเกี่ยวกับการใช้ switch statement ในกรณีที่ไม่ใช่ break ที่ตอนท้ายของแต่ละ case แล้วจะทำให้โค้ดใน case ที่อยู่ติดกันจะถูก execute ด้วย

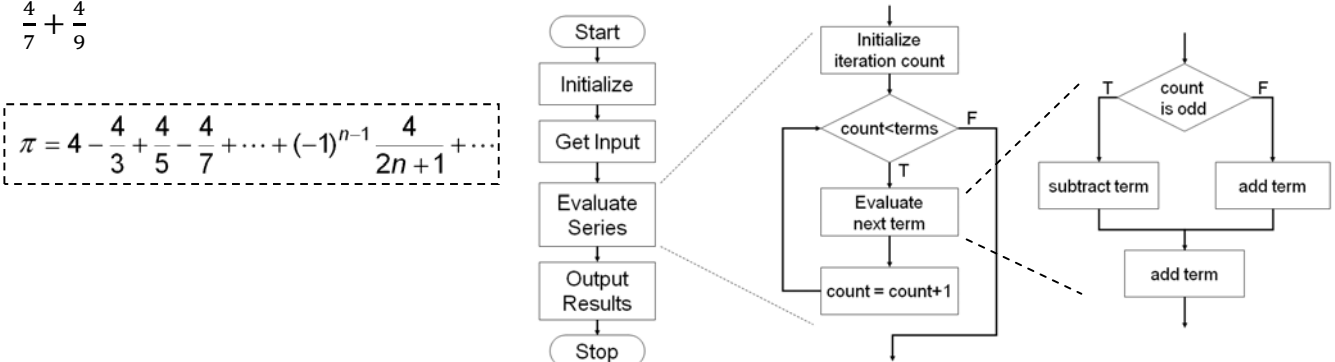
<pre>scanf("%d",&a); switch(a) { case 1: printf("a"); case 2: printf("b"); break; case 3: printf("c"); default: printf("d"); }</pre> ถ้าป้อน 1 (a เท่ากับ 1): แสดง ab ถ้าป้อน 2 (a เท่ากับ 2): แสดง b ถ้าป้อน 3 (a เท่ากับ 3): แสดง cd ถ้าป้อนค่าอื่นๆ แสดง d	ส่วนใหญ่จงใจที่จะใช้ในกรณีที่หลายๆ case มีโค้ดทำงาน เหมือนกัน เช่น โค้ดข้างล่างใช้หาจำนวนวันในแต่ละเดือน <i>/* Note: โค้ดข้างล่างใส่มาบาง case เพราะมีเนื้อที่ไม่พอ */</i> <pre>switch(month) { case 4: case 9: case 11: days=30; break; case 1: case 3: days=31; break; case 2: days=28; break; }</pre>
---	---

6. Nested loop: เราสามารถเขียน loop ซ้อนกับ loop ได้ดังตัวอย่างต่อไปนี้

<pre>#include <stdio.h> int main() { int N,i,j,x; scanf("%d",&N); for(i=1;i<=N;i++) { for(j=1;j<=i;j++) { x=j*i; printf("%d ",x); } printf("\n"); } return 0; }</pre>	จรรยาบรรณผลลัพธ์ที่หน้าจอของโปรแกรมด้านข้าง เมื่อผู้ใช้ป้อนค่า 5
---	---

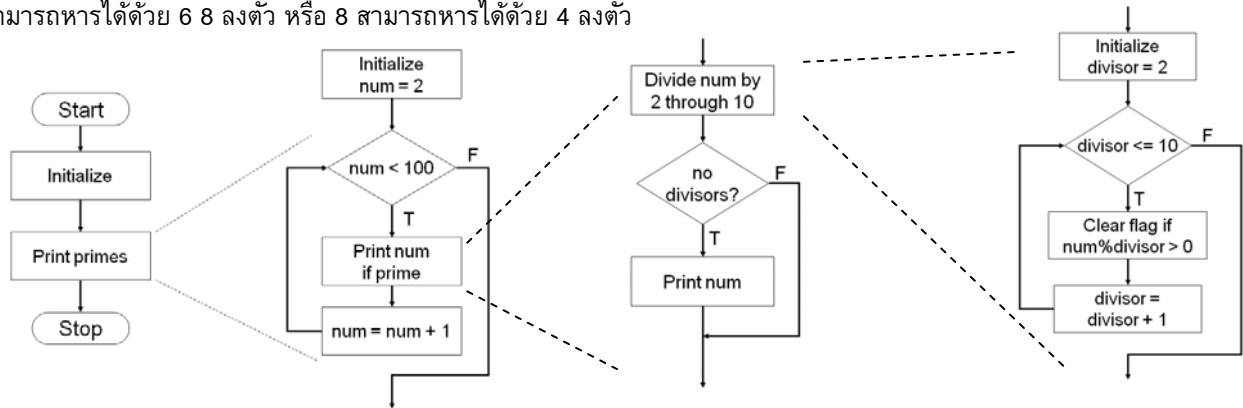
Problem solving using control structures (Chapter 13.4)

7. Textbook หน้า 360 และ Slide หน้า 13-23: จงเขียนโปรแกรมเพื่อคำนวณค่าประมาณของ π ตามสูตรข้างล่าง โดยที่ผู้ใช้จะป้อนจำนวนเทอม ที่จะใช้ในการคำนวณในสูตร เช่น ถ้าผู้ใช้กำหนดจำนวนเทอมเท่ากับ 5 ค่า $\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9}$



8. Textbook หน้า 362 และ Slide หน้า 13-28: จงเขียนโปรแกรมเพื่อแสดงตัวเลขจำนวนเฉพาะ (Prime number) ที่มีค่าน้อยกว่า 100

Fact: จำนวนที่ไม่ใช่จำนวนเฉพาะที่มีค่า 2-100 จะสามารถหารด้วยค่าที่อยู่ในช่วง 2-10 และค่านี้ไม่ใช่ตัวมันเอง เช่น 60 สามารถหารได้ด้วย 6 8 ลงตัว หรือ 8 สามารถหารได้ด้วย 4 ลงตัว



โจทย์ตัวอย่าง

9. ลูกบอลลูกหนึ่งเมื่อตกกระทบพื้นจะกระดอนขึ้นเป็น $2/5$ เท่าของความสูงที่ตกและเมื่อความสูงที่ตกต่ำกว่า 1 เซนติเมตร ลูกบอลจะไม่กระดอนอีก

จงเขียนโปรแกรมเพื่อคำนวณหาระยะทางที่ขึ้นและลงรวมกันทั้งหมด และจำนวนครั้งที่ลูกบอลกระทบพื้นจนลูกบอลหยุดนิ่ง โดยให้โปรแกรมรับค่าความสูงเริ่มต้น (cm) ของลูกบอลจากผู้ใช้งานซึ่งจะมีค่ามากกว่าหรือเท่ากับ 1 เซนติเมตร หลังจากนั้นแสดงผลจำนวนครั้งที่ลูกบอลกระทบพื้นและระยะทางที่ขึ้นและลงรวมกันทั้งหมดในหน่วยเซนติเมตร (เลขทศนิยม 2 ตำแหน่ง) ออกมาที่หน้าจอที่ละบรรทัด ตัวอย่างการทำงานของโปรแกรมแสดงได้ดังนี้

10. จงเขียนโปรแกรมหาค่าเฉลี่ย (Mean) ค่าน้อยสุด (Minimum value) และค่ามากที่สุด (Maximum value) ของข้อมูล M ตัว ($M > 0$) โดยข้อมูลที่ป้อนเข้ามา (input) มีรูปแบบ (format) ดังนี้

- บรรทัดแรกบอกจำนวนของข้อมูลทั้งหมด สมมุติ M
- ตั้งแต่บรรทัดที่ 2 ไปเป็นจำนวน M บรรทัดเป็นข้อมูลแต่ละตัว โดยกำหนดให้ข้อมูลเป็นตัวเลขแบบจำนวนเต็ม

เมื่อได้รับ input ครบแล้ว ให้แสดงค่าเฉลี่ย (แสดงเลขทศนิยม 2 หลัก) ค่าน้อยที่สุด และค่ามากที่สุด ออกมาที่หน้าจอที่ละบรรทัด

11. จงเขียนโปรแกรมเพื่อคำนวณหาพิสัย (Range) ของข้อมูลจำนวนเต็มบวก โดยข้อมูลนำเข้าจะเป็นจำนวนเต็มบวก บรรทัดละตัว และสิ้นสุดด้วย -1 เมื่อสิ้นสุดข้อมูลนำเข้าให้แสดงค่าพิสัยข้อมูลชุดนั้น

นิยาม พิสัย (Range) ของข้อมูลกลุ่มหนึ่งๆ มีค่าเท่ากับ ค่ามากที่สุด (Maximum value) ลบด้วยค่าน้อยสุด (Minimum value)

ตัวอย่างการทำงานของโปรแกรมข้อ 9

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
3	3 6.36

ตัวอย่างการทำงานของโปรแกรมข้อ 10

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
5 325 732 -520 983 881	480.20 -520 983

ตัวอย่างการทำงานของโปรแกรมข้อ 11

ข้อมูลนำเข้า (Input)	ข้อมูลส่งออก (Output)
12 31 10 1 41 32 -1	40