

w07-Lab

Iterative Control Statements

Part I

Adapted for 204111
by Areerat Trongratsameethong

Basic Program Instructions

A few basic instructions appear in just about every language:

- input
- output
- math
- conditional execution
- **repetition**

Topics

- **Repetition**
- Basic Loop Structures
- The while Statement
- Computing Sums and Averages Using a while Loop

Repetition

- **Repeating identical or similar tasks without making errors is something that computers do well and people do poorly.**

การทำงานที่เหมือนกันหรือคล้ายคลึงซ้ำ ๆ อย่ง
ถูกต้องเป็นสิ่งที่ Computer ทำได้ดี ในขณะที่มนุษย์
ทำได้ไม่ดีนัก

Temperature Conversion Revisited

```
#include <stdio.h>
#include <stdlib.h>
// ต้องการสร้างตารางการแปลงอุณหภูมิ จาก องศาฟาเรนไฮต์ เป็นองศาเซลเซียสในช่วง 60 - 75 องศาฟาเรนไฮต์
int main()
{
    double temp_f;
    double temp_c;
    temp_f = 60; // 60 F

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);
    temp_f += 5; // 65 F

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);
    temp_f += 5; // 70 F

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);
    temp_f += 5; // 75 F

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);

    return 0;
}
```

Identical
Tasks

Output

```
60.00 degree F is degree 15.56 C
65.00 degree F is degree 18.33 C
70.00 degree F is degree 21.11 C
75.00 degree F is degree 23.89 C
```

Repetition [2]

- A section of code that is repeated is called Loop
ส่วนของ code ที่ถูกทำซ้ำ เรียกว่า **Loop**
- Because, after the last statement in the code is executed เพราะเมื่อคำสั่งสุดท้ายถูกดำเนินการ
- The program branches, or loops, back to the first statement and starts another repetition through the code โปรแกรมจะวกกลับ (loop back) ไปดำเนินการที่คำสั่งแรก และดำเนินการซ้ำ
- Each repetition is also called an iteration or pass through the loop ในแต่ละรอบของการทำซ้ำเรียกว่า iteration หรือการวนลูป

Temperature Conversion Revisited [2]

```
#include <stdio.h>
#include <stdlib.h>
// ต้องการสร้างตารางการแปลงอุณหภูมิ จาก องศาฟาเรนไฮต์ เป็นองศาเซลเซียสในช่วง 60 - 75 องศาฟาเรนไฮต์
int main()
{
    double temp_f;
    double temp_c;
    temp_f = 60; // 60 F

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);
    temp_f += 5; // 65 F

    while (temp_f <= 75) {
        temp_c = (5.0/9.0) * (temp_f - 32);
        printf("%.2f degree F is degree %.2f C\n",
            temp_f, temp_c);
        temp_f += 5;
    }

    temp_c = (5.0/9.0) * (temp_f - 32);
    printf("%.2f degree F is degree %.2f C\n", temp_f, temp_c);

    return 0;
}
```

With repetition
(Loop)

6

Topics

- Repetition
- Basic Loop Structures
- The while Statement
- Computing Sums and Averages Using a while Loop

Basic Loop Structures

- Constructing a repeating section of code requires that four elements be present: การสร้าง Loop ประกอบด้วย 4 องค์ประกอบหลักคือ

- Repetition statement:** คำสั่งที่ใช้สำหรับการทำซ้ำ
 while statement
 for statement
 do-while statement } Later in the course
- Condition:** เงื่อนไขของการทำซ้ำ
- A statement that initially sets the condition being tested:**
 การตั้งค่าเริ่มต้นให้กับตัวแปรที่ใช้ในการควบคุมเงื่อนไข
- A statement within the repeating section of code that alters the condition so that it eventually becomes false:**
 การเปลี่ยนแปลงค่าตัวแปรที่ใช้ในการควบคุมเงื่อนไข ในการวนแต่ละครั้ง เพื่อให้ เงื่อนไขเป็นเท็จในที่สุด

Basic Loop Structures [2]

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

int main()
{
    double temp_f;
    double temp_c;

    temp_f = 60;

    while (temp_f <= 75) {
        temp_c = (5.0/9.0) * (temp_f - 32);
        printf("%.2f degree F is degree %.2f C\n",
               temp_f, temp_c);
        temp_f += 5;
    }

    return 0;
}
```

Initializing Condition (3)
 Condition (2)
 Repetition Statement (1)
 Condition altering (4)

Types of Loops

- Counter-Controlled Loops**
 - Loop ที่ทำการวนซ้ำตามจำนวนครั้งที่กำหนด
- Condition-Controlled Loops**
 - Loop ที่ทำการวนซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ

Counter-Controlled Loops

- Counter-controlled loop:** the condition is used to keep track of the number of repetitions
 - Also known as a fixed-count loop
 ทำด้วยจำนวนครั้งที่กำหนดตายตัว ตัวแปรที่ใช้นับจะเป็นตัวที่ใช้ตรวจสอบเงื่อนไข

```
int count;
count = 0;

while (count < 10) {
    printf("count = %d", count);
    count++;
}
```

Repetition Statement
 Initially sets the condition
 Condition
 Alters the condition

Condition-Controlled Loops

- **Condition-controlled loop: the tested condition does not depend on a count being achieved, but rather on a specific value being encountered**

เงื่อนไขไม่ได้ขึ้นอยู่กับตัวที่ใช้นับ แต่ขึ้นอยู่กับค่าอื่น ๆ ที่กำหนด

```
printf("Input scores 0 - 100, others to exit loop: ");
scanf("%f", &scores);
while (scores >= 0 && scores <= 100) {
    ...
    printf("Input scores 0 - 100, others to exit loop: ");
    scanf("%f", &scores);
}
```

The `while` Statement

- The general form of the while statement is

```
while (expression) {
    statement;
}
```

// ถูกดำเนินการก็ต่อเมื่อ `expression` เป็นจริง และ
// ไปเรื่อย ๆ จนกระทั่ง `expression` เป็นเท็จ

- The transfer of control back to the start of a while statement to reevaluate the expression is known as a program loop
การวนซ้ำกลับไปยังคำสั่งแรกของ while เพื่อตรวจสอบเงื่อนไข เรียกว่า *program loop*

- The following is a valid but ***infinite loop*** (Loop ที่ทำงานต่อไปเรื่อย ๆ ไม่รู้จบ):

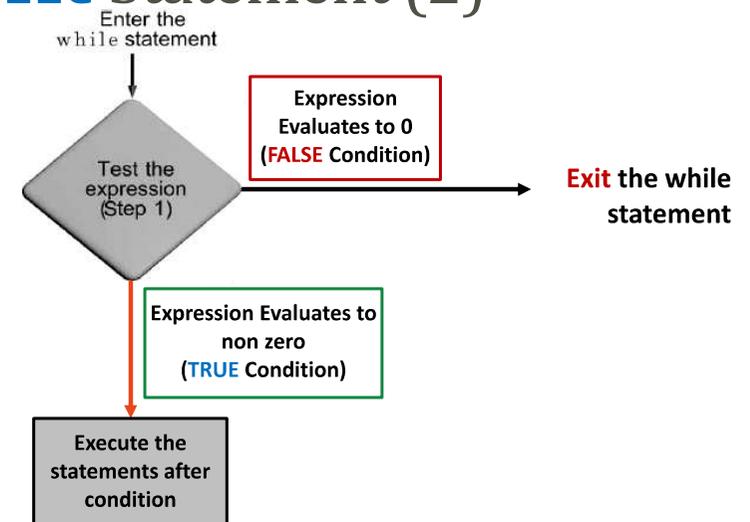
```
while (count <= 10) {
    printf("%d ", count);
}
```

ทำไมถึงเป็น infinite loop ?

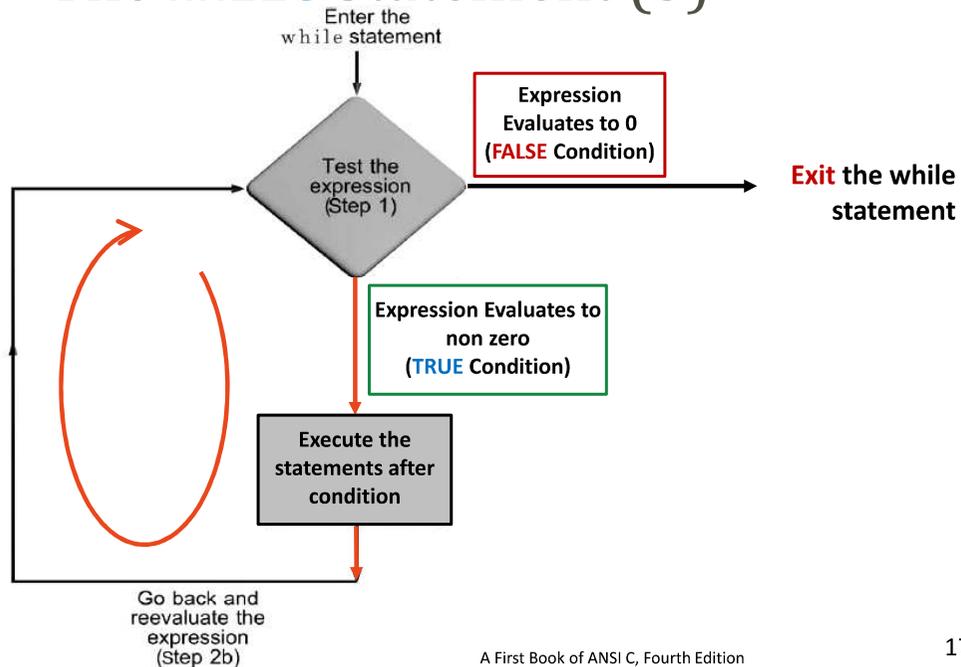
Topics

- Repetition
- Basic Loop Structures
- The `while` Statement
- Computing Sums and Averages Using a `while` Loop

The `while` Statement (2)



The `while` Statement (3)



The `while` Statement (4)

```

01 #include <stdio.h>
02 #include <stdlib.h>
03
04 int main()
05 {
06     int count;    /* initialize count */
07     count = 1;
08
09     while (count <= 10) {
10         printf("%d", count);
11         count++; /* add 1 to count */
12     }
13
14     printf("\n"); /*print a blank line */
15
16     return 0;
17 }
  
```

Output is:

1 2 3 4 5 6 7 8 9 10

The `while` Statement (5)

```

01 #include <stdio.h>
02 #include <stdlib.h>
03
04 int main()
05 {
06     int i;
07     i = 10;
08
09     while (i >= 1) {
10         printf("%d", i);
11         i--; /* subtract 1 from i */
12     }
13
14     printf("\n"); /*print a blank line */
15
16     return 0;
17 }
  
```

Output is:

10 9 8 7 6 5 4 3 2 1

The `while` Statement (6)

```

01 #include <stdio.h>
02 #include <stdlib.h>
03 #define TABLESIZE 10
04
05 int main()
06 {
07     int num;
08
09     printf("NUMBER  SQUARE  CUBE\n");
10     printf("-----  -----  -----\n");
11     num = 1;
12
13     while (num <= TABLESIZE) {
14         printf("%3d %7d %6d\n", num, num*num, num*num*num);
15         num++; /* add 1 to num */
16     }
17
18     return 0;
19 }
  
```

Output is:

NUMBER	SQUARE	CUBE
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

The `while` Statement (7)

```

01 #include <stdio.h>
02 #include <stdlib.h>
03
04 #define ENDVALUE 50
05 int main() /*Program to convert Celsius to Fahrenheit */
06 {
07     double dCelsius;
08     double dFahren;
09
10     /* display the heading lines */
11     printf("DEGREE  DEGREE\n");
12     printf("CELSIUS  FAHRENHEIT\n");
13     printf("-----  -----\n");
14
15     // now fill in the table using a while loop
16     dCelsius = 5;      /* starting Celsius value */
17
18     while (dCelsius <= ENDVALUE) {
19         dFahren = (9.0 / 5.0) * dCelsius + 32.0;
20         printf("%7.2f%11.2f\n",dCelsius ,dFahren);
21         dCelsius = dCelsius + 5;
22     }
23
24     return 0;
25
26 }

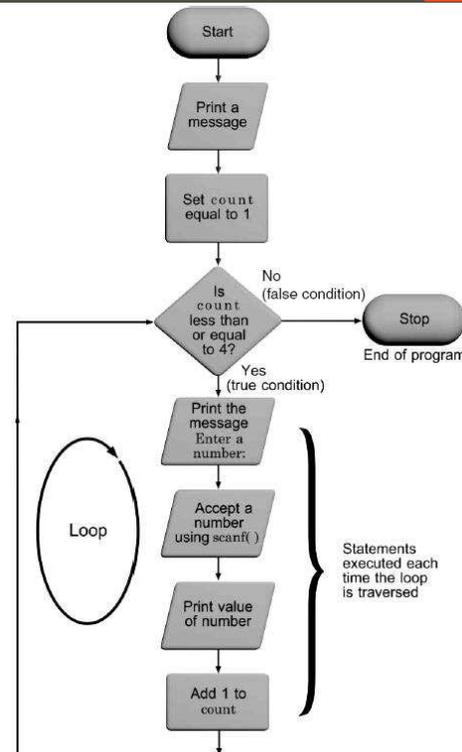
```

Output is:

DEGREE CELSIUS	DEGREE FAHRENHEIT
5.00	41.00
10.00	50.00
15.00	59.00
20.00	68.00
25.00	77.00
30.00	86.00
35.00	95.00
40.00	104.00
45.00	113.00
50.00	122.00

Condition-controlled loop

Entering Numbers Using a `while` Loop



Topics

- Repetition
- Basic Loop Structures
- The `while` Statement
- Computing Sums and Averages Using a `while` Loop

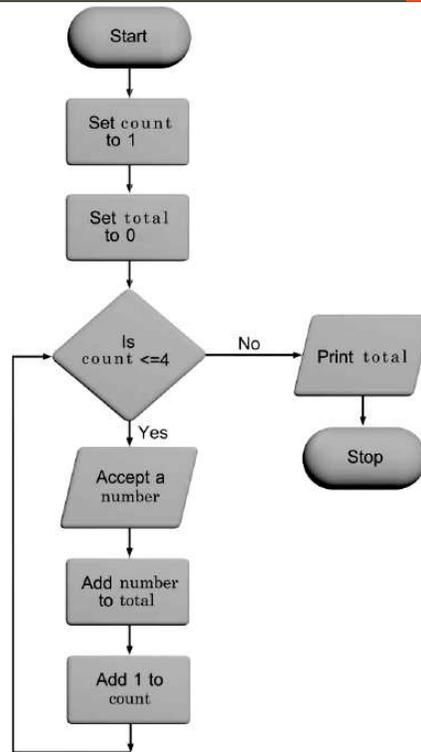
Entering Numbers Using a `while` Loop [2]

```

01 #include <stdio.h>
02 #define MAXCOUNT 4
03
04 int main()
05 {
06
07     int count;
08     double num;
09
10     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
11     count = 1;
12
13     while (count <= MAXCOUNT) {
14         printf("Enter a number: ");
15         scanf("%f", &num);
16         printf("The number entered is %f\n", num);
17         count++;
18     }
19     return 0;
20 }

```

Computing Sums Using a **while** Loop



25

Computing Sums Using a **while** Loop [2]

```

01 #include <stdio.h>
02 #include <stdlib.h>
03
04 #define MAXCOUNT 4
05
06 int main()
07 {
08     int count;
09     double num, total;
10
11     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
12
13     count = 1;
14     total = 0.0;
15
16     while (count <= MAXCOUNT) {
17
18         printf("Enter a number: ");
19         scanf("%le", &num);
20         total += num;
21         printf("The total is now %f\n", total);
22         count++;
23         printf("\n\nThe final total of the %d numbers is %f\n", MAXCOUNT, total);
24
25     }
26     return 0;
27 }
  
```

Ensures that any previous value present in the storage locations assigned to the variable **total** is overwritten and the **total** starts at a correct value

Accumulating statement

26

Computing Sums and Average Using a **while** Loop

```

01 #include <stdio.h>
02 #define MAXCOUNT 4
03
04 int main()
05 {
06     int count;
07     double num, total, average;
08
09     printf("\nThis program will ask you to enter %d numbers.\n\n", MAXCOUNT);
10
11     count = 1;
12     total = 0.0;
13
14     while (count <= MAXCOUNT) {
15         printf("Enter a number: ");
16         scanf("%le", &num);
17         total += num;
18         count++;
19     }
20     average = total / MAXCOUNT;
21     printf("\n\nThe average of the %d numbers is %8.4f\n", MAXCOUNT, average);
22
23     return 0;
24 }
  
```

Calculating an average

27

Practice 1

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อหาค่าผลบวกของตัวเลขจำนวนเต็มเริ่มจาก 1 ถึง n โดยที่ n เป็นจำนวนเต็มบวกใดๆ

ไม่อนุญาต ให้ใช้สูตรผลบวกเลขคณิต: $n * (n + 1) / 2$

• ตัวอย่างการ run

Input an integer: 25
Sum of integers from 1 to 25 is: 325

28

Practice 2

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อหาค่า Factorial (!) ของค่า integer n ($0 \leq n \leq 13$) เช่น $4! = 24$

Note: $0! = 1$

- ตัวอย่างการ run 1

```
Input an integer (0 - 13): 9
The factorial of 9 is: 362880
```

- ตัวอย่างการ run 2

```
Input an integer (0 - 13): 26
Input out of range
```

29

Practice 4

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อหาค่า x^y ใดๆ โดยที่ x เป็นเลขจำนวนจริง และ y เป็นจำนวนเต็มใดๆ (ลบ, 0, หรือ บวก) เช่น $2.5^{-2} = 0.16$, $100^0 = 1$, $2.3^3 = 12.167$

- ตัวอย่างการ run

```
Input the base number: 2
Input the exponent: -3
2 to the power of -3 is: 0.125
```

31

Practice 3

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อหาค่า x^y ใดๆ โดยที่ x เป็นเลขจำนวนจริง และ y เป็นจำนวนเต็มมากกว่าหรือเท่ากับ 0 เช่น $2.5^3 = 15.625$

- ตัวอย่างการ run

```
Input the base number: 2.5
Input the exponent: 3
2.5 to the power of 3 is: 15.625
```

30

Practice 5

แปลงเลขจำนวนเต็มฐาน 10 เป็นเลขฐาน 2

เช่น $45_{10} = 101101_2$ โดย input จะอยู่ในช่วง (0 - 1023)

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อแก้ปัญหาดังกล่าวข้างต้น

- ตัวอย่างการ run

```
Input integer in base 10 (0 - 1023): 28
Base 2 representation of 28 is: 11100
```

32

Practice 5 (2)

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

Step 1: Problem Solving

ยกตัวอย่าง แปลงเลข 45_{10} เป็นฐาน 2

ให้ **result** คือค่าผลลัพธ์ ให้ค่าตั้งต้นเป็น 0

	$\text{result} = \text{result} + (\text{เศษ} \times 10^i), i = 0, 1, 2, \dots$
$45 / 2 = 22$	เศษ 1 $\rightarrow \text{result} = 0 + (1 \times 10^0)$
$22 / 2 = 11$	เศษ 0 $\rightarrow \text{result} = 1 + (0 \times 10^1)$
$11 / 2 = 5$	เศษ 1 $\rightarrow \text{result} = 01 + (1 \times 10^2)$
$5 / 2 = 2$	เศษ 1 $\rightarrow \text{result} = 101 + (1 \times 10^3)$
$2 / 2 = 1$	เศษ 0 $\rightarrow \text{result} = 1101 + (0 \times 10^4)$
$1 / 2 = 0$	เศษ 1 $\rightarrow \text{result} = 1101 + (1 \times 10^5)$
	= 101101

33

Practice 6

1. Statement	while
2. Condition	
3. Initializing Condition	
4. Modifying Condition	

ให้เขียน Algorithm และโปรแกรมภาษา C เพื่อ
แปลงเลขทศนิยมฐาน 10 ใดๆ ให้เป็นเลขฐาน 2
พร้อมทศนิยม 6 ตำแหน่ง เช่น $44.1875_{10} =$

101100.001100_2

- ตัวอย่างการ run

Input a number in base 10: 44.1875
Base 2 representation of 44.1875 is: 101100.001100

34