

w10-Lab

Iterative Control Statements

Part III

Adapted for 204111
by Areerat Trongratsameethong

The EOF Character Sentinels [2]

```
#include <stdio.h>
int main() {
    double grade;
    double total = 0.01; /* note the initialization here */

    printf("\nTo stop entering grades, press either the F6 key");
    printf("\n or the ctrl and Z keys simultaneously on PCs");
    printf("IBM compatible computers");
    printf("\n or the ctrl and D keys for UNIX OS.\n\n");
    printf("Enter a grade: ");

    while (scanf("%lf", &grade) != EOF) {
        total += grade;
        printf("Enter a grade: ");
    }

    printf("\nThe total of the grades is %f\n", total);
    return 0;
}
```

Sentinel

The EOF Character Sentinels

- EOF ย่อมาจาก End Of File เป็นหนึ่งใน sentinel ที่ใช้ในภาษา C
- ค่า EOF ขึ้นอยู่กับ Compiler ที่จะกำหนดค่าอะไรให้กับ EOF แต่ทั้งนี้ค่าที่กำหนดให้จะไม่ซ้ำกับค่าที่ใช้โดย character ตัวอื่น ๆ
- EOF ถูกกำหนดอยู่ใน `stdio.h`

เมื่อผู้ใช้กด CTRL Z
ฟังก์ชัน `scanf()` จะส่งค่า EOF
กลับมาให้ผู้ที่เรียกใช้ฟังก์ชัน



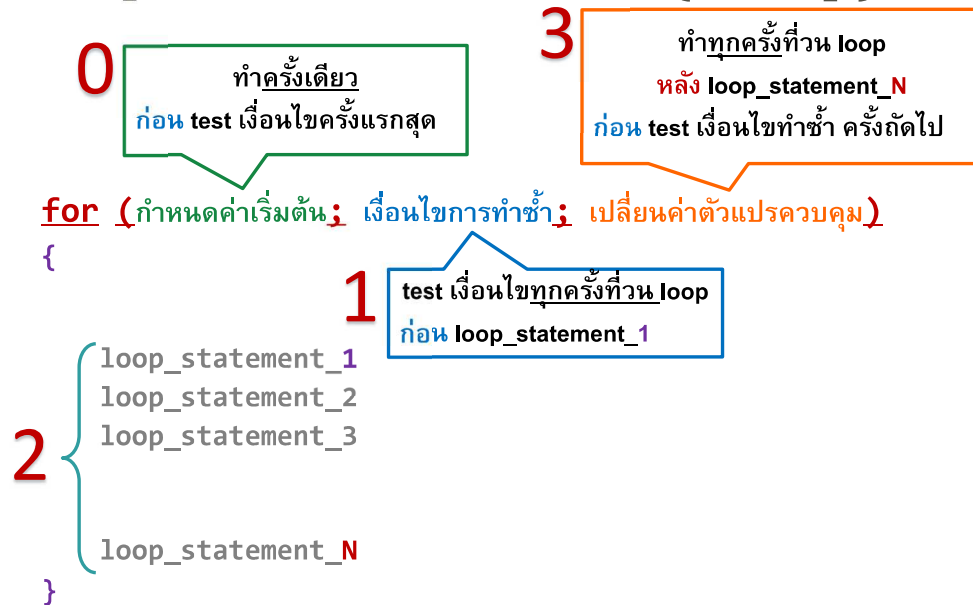
Figure 5.7 Generation of the EOF constant by the `scanf()` function

The for Statement (Recap)

- คำสั่ง `for` เป็นคำสั่งที่รวมองค์ประกอบของ loop ทั้ง 4 ข้อไว้ในบรรทัดเดียว
- เราไม่จำเป็นต้องระบุทั้ง 3 ค่าภายใน บรรทัดที่มีคำสั่ง `for` (เช่นสามารถกำหนด ค่าเริ่มต้นที่บรรทัดอื่น ก่อนคำสั่ง `for` ได้)
 - แต่จำเป็นต้องใส่ semi-colon ให้ครบทั้งสองอันภายในวงเล็บ
 - `for (; count <= 20;) // valid syntax`
 - `for (; ;) // also valid syntax`

การละเงื่อนไขการทำซ้ำ ทำให้เป็น infinite loop.. วิธีแก้?

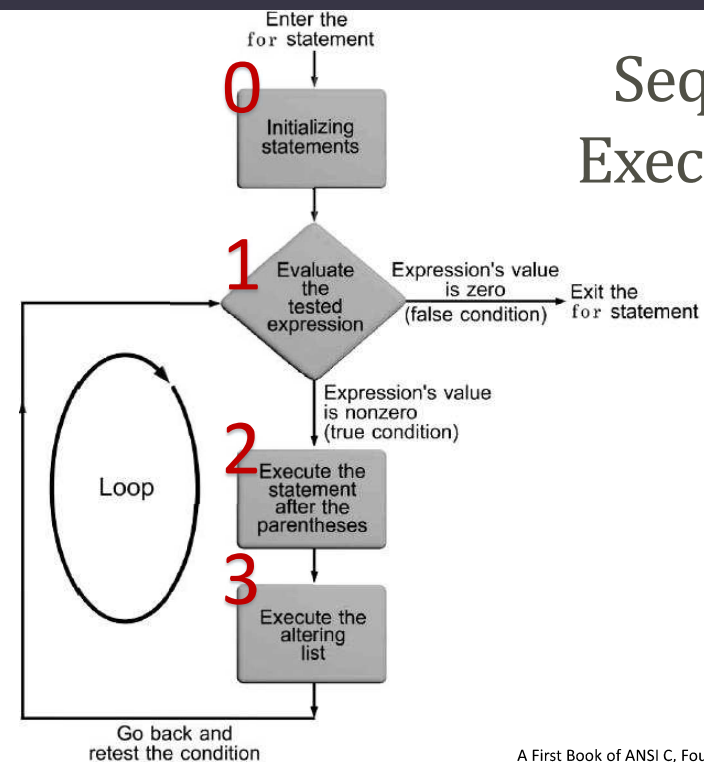
Sequence of Execution (Recap)



A First Book of ANSI C, Fourth Edition

5

Sequence of Execution [2]



A First Book of ANSI C, Fourth Edition

6

The Null Statement

- A semicolon with nothing preceding it is also a valid statement, called the null statement
- Use the null statement where a statement is syntactically required, but no action is needed
- Null statements typically are used either with while or for statements

;

A First Book of ANSI C, Fourth Edition

7

Sums and Averages with for Loop

```
#include <stdio.h>
#define MAXCOUNT 5
int main() /* This program calculates the average */
{
    int count;
    double num, total, average;
    total = 0.0;
    for (count = 0; count < MAXCOUNT; count++){
        printf("\nEnter a number: ");
        scanf("%lf", &num);
        total += num;
    }
    average = total / MAXCOUNT;

    printf("\n\nThe average of the %d numbers", MAXCOUNT);
    printf("entered is %f\n", average);
    return 0;
}
```

A First Book of ANSI C, 4th Edition

8

Case Studies: Loop Programming Techniques

- Technique 1: Selection within a loop
- Technique 2: Input data validation
- Technique 3: Interactive loop control
- Technique 4: Evaluating equations

2: Input Data Validation

```

01 #include <stdio.h>
02
03 int main() {
04     int month;
05     printf("\nEnter a month between 1 and 12: ");
06     scanf("%d", &month);
07
08     while (month < 1 || month > 12) {
09         printf("Error - the month you entered is not valid.\n");
10         printf("\nEnter a month between 1 and 12: ");
11         scanf("%d", &month);
12     }
13     printf("The month accepted is %d\n", month);
14     return 0;
15 }

```

Same code used
in lines 5-6!

1: Selection within a Loop

```

01 #include <stdio.h>
02 #define MAXNUMS 5
03
04
05 int main()          /* this program computes the positive and negative */
06 {                  /* sums of a set of MAXNUMS user entered numbers */
07     int i;
08     double number;
09     double postotal = 0.0f;
10     double negtotal = 0.0f;
11
12     for (i = 1; i <= MAXNUMS; i++) {
13         printf("Enter a number (positive or negative)");
14         scanf("%lf", &number);
15         if (number > 0)
16             postotal += number;
17         else
18             negtotal += number;
19     }
20
21     printf("\nThe positive total is %f", postotal);
22     printf("\nThe negative total is %f\n", negtotal);
23
24     return 0;
25 }

```

2: Input Data Validation [2]

```

01 #include <stdio.h>
02 #define TRUE 1
03
04 int main()
05 {
06     int month;
07     while (TRUE) { /* this is always true */
08
09         printf("\nEnter a month between 1 and 12: ");
10         scanf("%d", &month);
11         if (month >= 1 && month <= 12) /* the test is made here */
12             break;
13         printf("Error - the month you entered is not valid.\n");
14     }
15     printf("The month accepted is %d\n", month);
16     return 0;
17 }

```

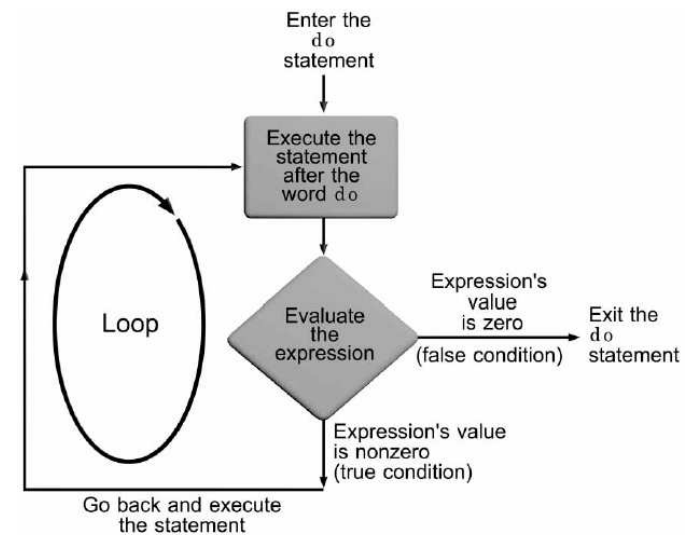
The `do-while` Statement (Recap)

- Post test loop

- เหมาะกับการตรวจสอบข้อมูลที่รับเข้ามา

```
do {
    printf("Input score (-1 to terminate): ");
    scanf("%le",&score);
} while (score != -1);
```

The `do-while` Statement (Recap) [2]

Figure 5.10 `do-while` statement flow of control

3: Interactive Loop Control

```
01 #include <stdio.h>
02 /* this program displays a table of numbers, */
03 /* their squares and cubes */
04 /* starting from the number 1. */
05 /* The final number in the table is */
06 /* input by the user */
07
08 int main()
09 {
10     int num, final;
11     printf("Enter the final number for the table: ");
12     scanf("%d", &final);
13
14     printf("Number Square Cube\n");
15     printf("-----\n");
16     for (num = 1; num <= final; num++)
17         printf("%3d %7d %6d\n", num, num*num, num*num*num);
18     return 0;
19 }
```

4: Evaluating Equations

```
01 #include <stdio.h>
02 #include <math.h>
03 int main() {
04     int x, y;
05
06     printf("x value y value\n");
07     printf("-----\n");
08
09     for (x = 2; x <= 6; x++) {
10         y = 10 * pow(x,2) + 3 * x - 2;
11         printf("%4d %10d\n", x, y);
12     }
13
14     return 0;
15 }
```

4: Evaluating Equations [2]

```

01 #include <stdio.h>
02 #include <math.h>
03 int main()
04 {
05     float x, y;
06     printf("x value          y value\n");
07     printf("          \n");
08     for (x = 2.0; x <= 6.0; x += 0.5)
09     {
10         y = 10.0 * pow(x,2) + 3.0 * x - 2.0;
11         printf("%8.6f %13.6f\n", x, y);
12     }
13     return 0;
14 }

```

Common Programming Errors [2]

- Placing a semicolon at the end of the for's parentheses (creates a do-nothing loop)

การวาง ; ไว้หลังคำสั่ง for เช่น for(i=0; i<n; i++); คำสั่งที่ทำให้ซ้ำจะไม่มีการทำงานใดๆ

- Using commas to separate the items in a for statement instead of the required semicolons

วาง , คั่นคำสั่งใน for loop แทนที่จะใช้ ; เช่น

```
for(i=0, i<n, i++)
```

- Omitting the final semicolon from the do statement

ลืมใส่ ; ตัวสุดท้ายของคำสั่ง do-while loop เช่น

```
do { ... } while ()
```

Common Programming Errors

- “Off by one” error, in which the loop executes either one too many or one too few times than intended
ทำเกินไป 1 Loop หรือ ทำขาดไป 1 Loop (ให้เช็คเงื่อนไขที่ใช้ในการควบคุม Loop ให้ดี)
- Using the assignment operator, =, instead of the equality operator, ==, in the tested expression
ใช้สัญลักษณ์สำหรับการเปรียบเทียบผิด คือ ใช้ = (สำหรับกำหนดค่าให้กับตัวแปร) แทนที่จะใช้ == (สำหรับเปรียบเทียบ)
- As with the if statement, repetition statements should not use the equality operator, ==, when testing single-precision or double-precision operands
ไม่ควรใช้การเปรียบเทียบในลักษณะเท่ากับ สำหรับการเปรียบเทียบตัวเลขทศนิยม เพราะโอกาสที่จะเท่ากันยากมาก (เนื่องจากทศนิยมมีหลายหลัก)

Common Compiler Errors

Error	Typical Unix-based Compiler Error Message	Typical Windows-based Compiler Error Message
Separating the statements in a for loop with commas rather than semicolons. For example, <code>for (init, cond, alt)</code>	(S) Syntax error: possible missing ';' or ','?	error: syntax error : missing ';' before '}'
Omitting the parenthesis in a while statement. For example, <code>while condition { statement; }</code>	(S) Syntax error: possible missing '('?	error: syntax error : missing ';' before '{'
Omitting the ; at the end of the do-while statement. For example, <code>do { statement; }while(condition)</code>	(S) Syntax error. (This error tends to lead programmers astray. You would expect to get the error generated by a missing semicolon or comma, but instead you get a syntax error.)	error: syntax error : missing ';' ;'
Omitting the second + or - in a post increment or decrement statement. For example, <code>val+; or val-;</code>	(S) Syntax error. (Note that <code>+val;</code> and <code>-val;</code> do not generate a compiler error because these are valid expressions)	error: syntax error : ';

Summary

- A section of repeating code is called a loop
- The three C repetition statements are while, for and do-while
- Loops are also classified as to the type of tested condition
- The most commonly used syntax for a while loop is

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

The `math.h` library (Recap) `#include <math.h>`

- เราสามารถเรียกใช้ function ทางคณิตศาสตร์ได้จาก library `math.h` ในปฏิบัติการนี้จะกล่าวถึงเฉพาะบาง function ที่พบบ่อย
- `double pow(double x, double y)`
 - Returns x^y
- `double sqrt(double x)`
 - Returns \sqrt{x}
- `double ceil(double x)`
 - Returns $\lceil x \rceil$
- `double floor(double x)`
 - Returns $\lfloor x \rfloor$
- `double fab(double x)`
 - Returns $|x|$

ข้อสังเกต: ค่า return ของ function เหล่านี้มีชนิด เป็น `double` หากนำมาใช้กับ ข้อมูลประเภท `integer` จำเป็นจะต้องมีการ `casting` ที่เหมาะสม

Summary [2]

- A for statement performs the same functions as the while statement, but uses a different form
- The for statement is extremely useful in creating counter-controlled loops
- The do-while statement is used to create posttest loops because it checks its expression at the end of the loop

Practice 1

ให้นำเข้าข้อมูลประเภทจำนวนเต็ม 2 ค่า คือค่าแรก (first) และค่าสุดท้าย (last) ผ่านทาง keyboard และนับว่ามีจำนวนเฉพาะกี่จำนวน อะไรบ้างเช่น

7 ถึง 21: มี จำนวนเฉพาะ 5 ตัวคือ: 7, 11, 13, 17, 19

- ตัวอย่างการ run 1

```
Input first value: 7
Input last value: 21
The prime number(s) between 7 - 21 are
7 11 13 17 19
Total: 5 numbers
```

Practice 2

ตัวอย่าง Output

Average of Sale Amount of Week (1) = 12800.00
 Average of Sale Amount of Week (2) = 13050.00
 Average of Sale Amount of Week (3) = 13728.57
 Average of Sale Amount of Week (4) = 14500.00
 Average of Sale Amount for All = 13519.64

ให้เขียนโปรแกรมภาษา C เพื่อหาค่าเฉลี่ยของยอดขายทั้งเดือน
 ค่าเฉลี่ยของยอดขายต่อสัปดาห์ ให้สอดคล้องกับตัวอย่างการ run
 โปรแกรมด้านล่าง โดยที่ค่าข้อมูลให้นำเข้าผ่านทาง keyboard

Week	Su	Mo	Tu	We	Th	Fr	Sa	Average (Per Week)
1	15000.00	10000.00	11500.00	9500.00	12600.00	14500.00	16500.00	12800.00
2	14500.00	11000.00	12500.00	9600.00	12650.00	14550.00	16550.00	13050.00
3	14800.00	11500.00	13500.00	9800.00	13500.00	15500.00	17500.00	13728.57
4	16000.00	12450.00	13550.00	10500.00	14500.00	16000.00	18500.00	14500.00
						Average (All)		13519.64

25

Practice 4

ให้นำเข้าข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
 ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
* * 1
* 2 2
3 3 3
```

- ตัวอย่างการ run 2

```
Input row: 4
* * * 1
* * 2 2
* 3 3 3
4 4 4 4
```

27

Practice 3

ให้นำเข้าข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
 ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
1
1 2
1 2 3
```

- ตัวอย่างการ run 2

```
Input row: 4
1
1 2
1 2 3
1 2 3 4
```

26

Practice 5

ให้นำเข้าข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
 ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
* * *
* *
*
```

- ตัวอย่างการ run 2

```
Input row: 4
* * * *
* * *
* *
*
```

28