Chapter5 Loop: while, for

max換人做? (可以用else if嗎?)

Input (scanf_s) 5 integers a, b, c, d, e. Print the maximum, minimum, sum and average (print the two digits after the decimal point)。

Please show two different outputs.

Output example:

Enter a=10
Enter b= 8
Enter c= 2
Enter d=88
Enter e=-1
Max=
Min=
Sum=
Average=

Output example:

Enter a=
Enter b=
Enter c=
Enter d=
Enter e=
Max=
Min=
Sum=
Average=

Try below method:

```
...
max=a;
if(max<b)
    max=b;
if(max<c)
    max=c;
....
```

Switch and if-else-if control structure

- Using if-else-if control structures
- Using switch statements
- Comparing switch and if-else-if control structures

Department of Electrical Engineering, Yuan Ze University

if-else-if control structure

要有expression!!

```
if (rational_expression_1)
   { statement_block_1     }
else if (rational_expression_2)
   { statement_block_2     }
.
.
.
.
.
else if  (rational_expression_n)
   { statement_block_n     }
else
   { statement_block       }
```

不能有expression!!

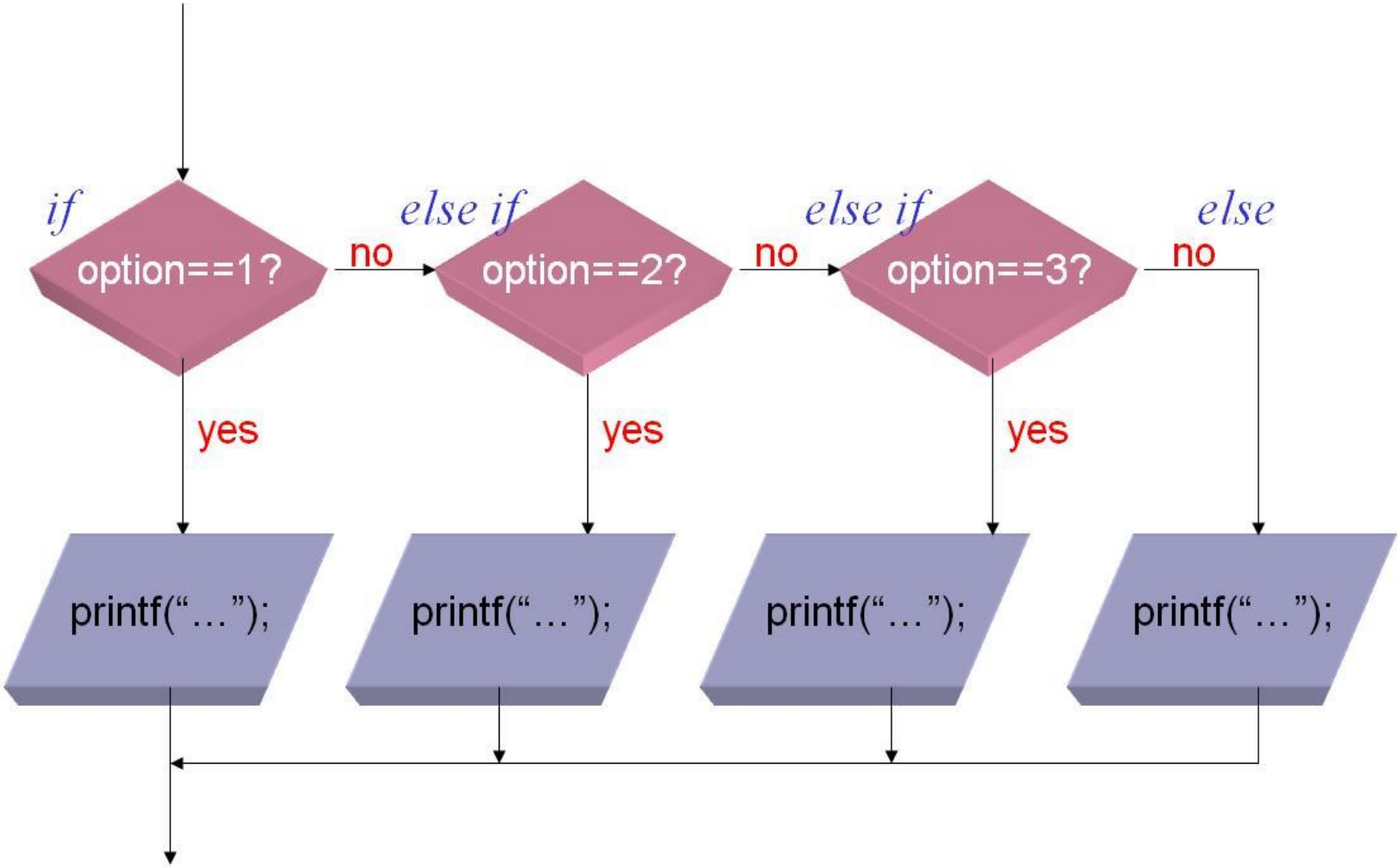
if-else-if control structure

```
#include <stdio.h>
 void main(void)
{
   int option;

   printf("Please type 1, 2, or 3\n"
          "1. Breakfast\n"
          "2. Lunch\n"
          "3. Dinner\n");
   scanf("%d",&option);

      if(option==1)
            {
            printf("Good morning\n");
            printf("Order breakfast\n");
            }
      else if (option==2)
            {
            printf("Order lunch\n");
            }
      else if (option==3)
            {
            printf("Order dinner\n");
            }
      else
            {
            printf("Order nothing\n");
            }
}
```

if-else-if control structure. Only one of the statement blocks (enclosed in braces) is executed.


if
option==1?
no
else if
option==2?
no
else if
option==3?
no
else
yes
yes
yes
printf("…");
printf("…");
printf("…");
printf("…");

switch control structure

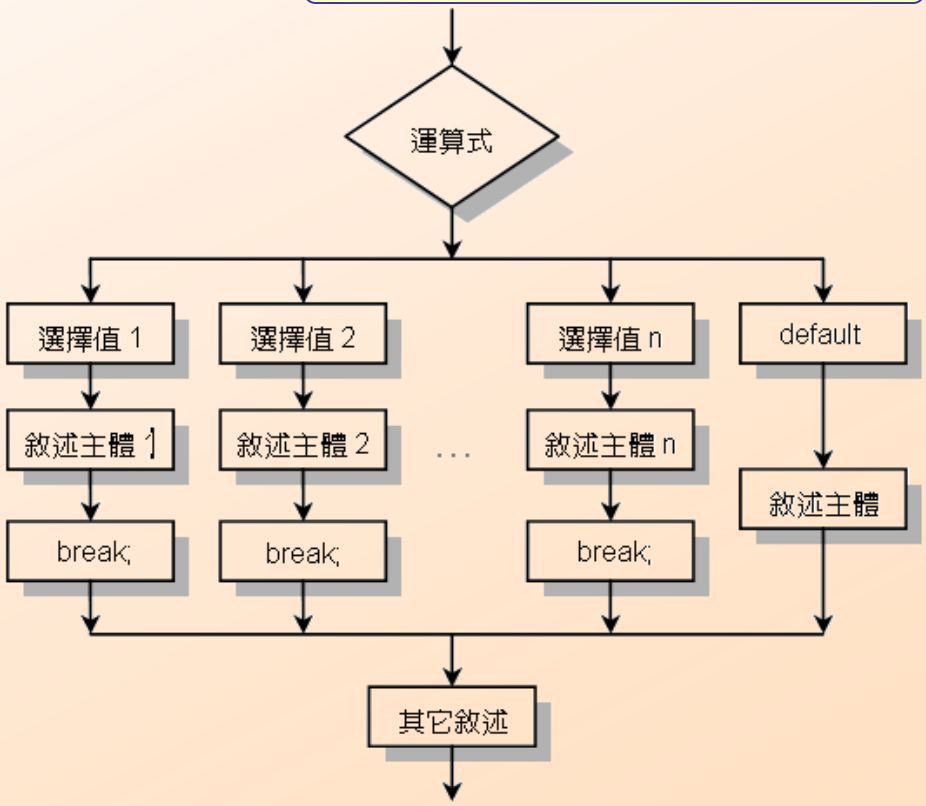
```
switch(expression)
{
case constant1:
        statement1a
        statement1b
        ...
case constant2:
        statement2a
        statement2b
        ...
        ...
default:
        statements
}
```

switch 敘述

- switch 敘述可用來進行多重選擇

switch 敘述的語法

```
switch(運算式)
{
    case 選擇值1:
        敘述主體1;
        break;
    case 選擇值2:
        敘述主體2;
        break;
        ...
    default:
        敘述主體;
}
```



switch control structure

```
#include <stdio.h>
void main(void)
{
 int option;

 printf("Please type 1, 2, or 3\n"
        "1. Breakfast\n"
        "2. Lunch\n"
        "3. Dinner\n");
 scanf("%d",&option);

 switch(option)
 {
 case 1: printf("Good morning\n");
         printf("Order breakfast\n");
         break;

 case 2: printf("Order lunch\n");
          break ;

 case 3: printf("Order dinner\n");
          break;

 default:
         printf("Order nothing\n");
 }

}
```

Labels are followed by colon.

Switch control structure.

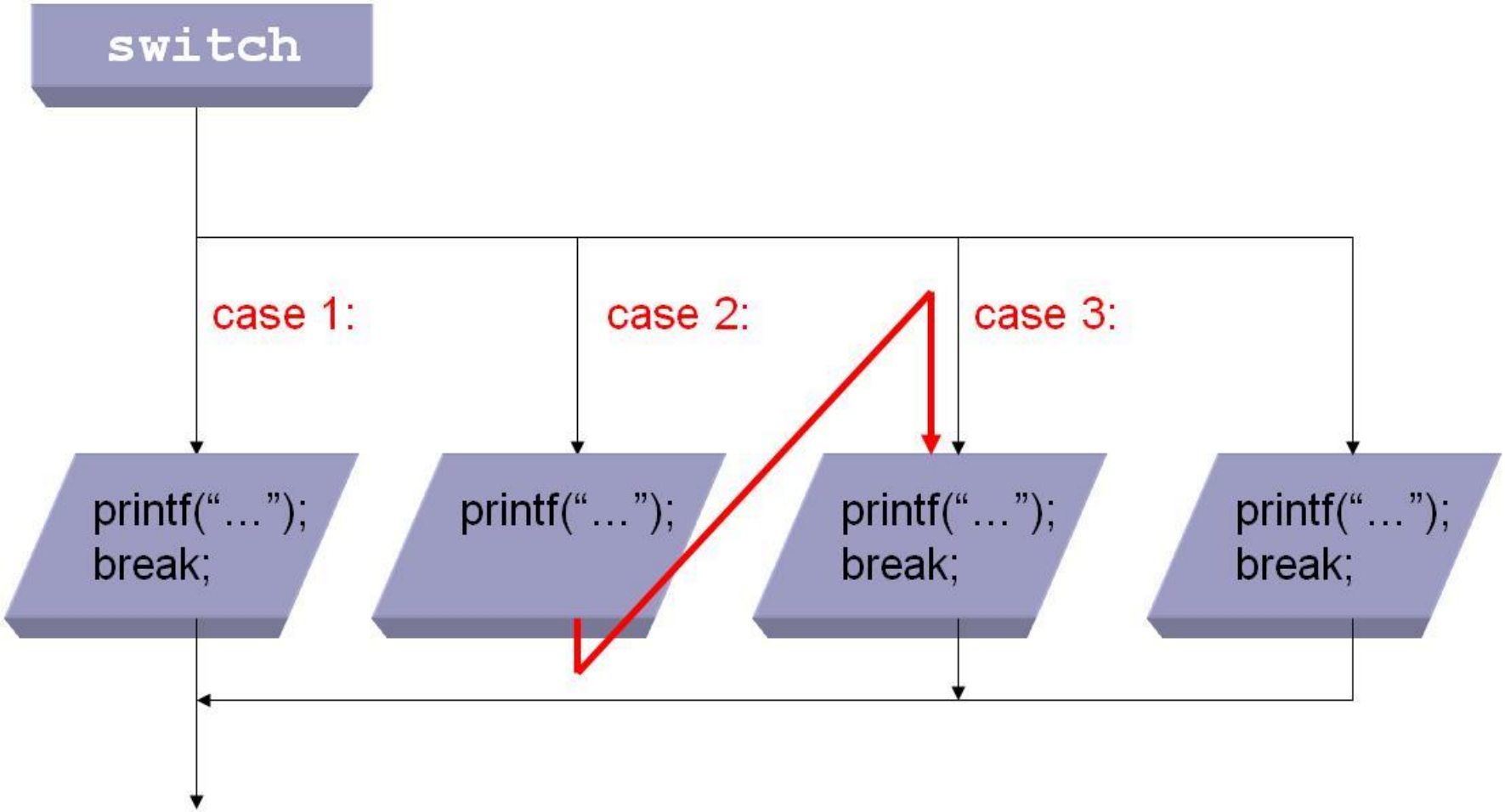
Break statements cause exiting of switch control structure.

break statement

- To terminate the process and process and exit switch.

```
switch(option)
  {
  case 1: printf("Entering case 1\n");
          break;

  case 2: printf("Entering case 2\n");
                                  no break statement here
  case 3: printf("Entering case 3\n");
          break;
  }
```


switch
case 1:
case 2:
case 3:
printf("…");
break;
printf("…");
printf("…");
break;
printf("…");
break;

將不同的選擇值並列

```
01  /* prog6_11, switch 敘述—以不同的選擇值來處理相同的敘述 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     char grade;
07     printf("Input grade:");
08     scanf("%c",&grade);
09
10     switch(grade)
11     {
12        case 'a':   /* 輸入 a 或 A 時印出 Excellent! */
13        case 'A':
14           printf("Excellent!\n");
15           break;
16        case 'b':   /* 輸入 b 或 B 時印出 Good! */
17        case 'B':
18           printf("Good!\n");
19           break;
```

```
20        case 'c':   /* 輸入 c 或 C 時印出 Be study hard! */
21        case 'C':
22           printf("Be study hard!\n");
23           break;
24        default:    /* 輸入其他字元時印出 Failed! */
25           printf("Failed!\n");
26     }
27     system("pause");
28     return 0;
29  }

/* prog6_11 OUTPUT---

Input grade:B
Good!
----------------------*/
```

Loops

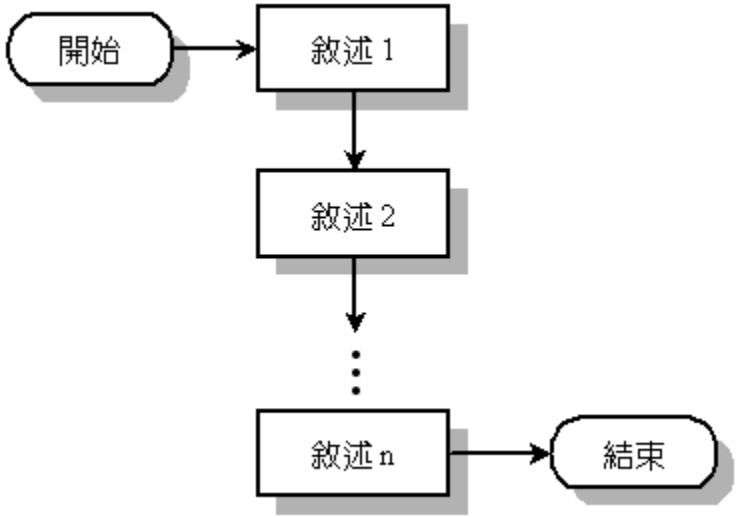
- C provides for a number of iterative control structures, known as *looping*.
- **`while`** loop
- **`do-while`** loop
- **`for`** loop

結構化程式設計

- 結構化的程式設計包含有下面三種結構：
  - 循序性結構（**sequence structure**）
  - 選擇性結構（**selection structure**）
  - 重複性結構（**iteration structure**）

結構化程式設計

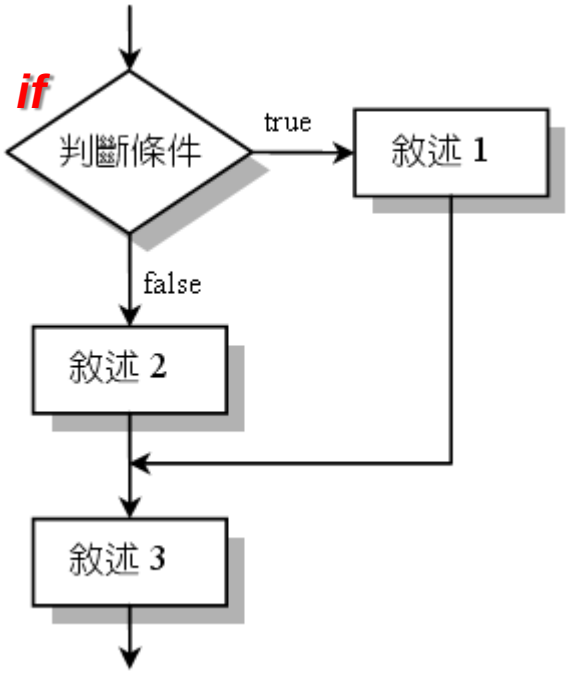
- 循序性結構（**sequence structure**）
  - 程式的執行流程是由上而下，一個接著一個敘述依序執行



結構化程式設計

```
if(數學成績>=90)
        印出"成績為A"
```

- 選擇性結構（selection structure）
  - 依條件判斷的結果來改變程式執行的流程




```
if (判斷條件成立)
        do something;
else
        do the other thing;
```

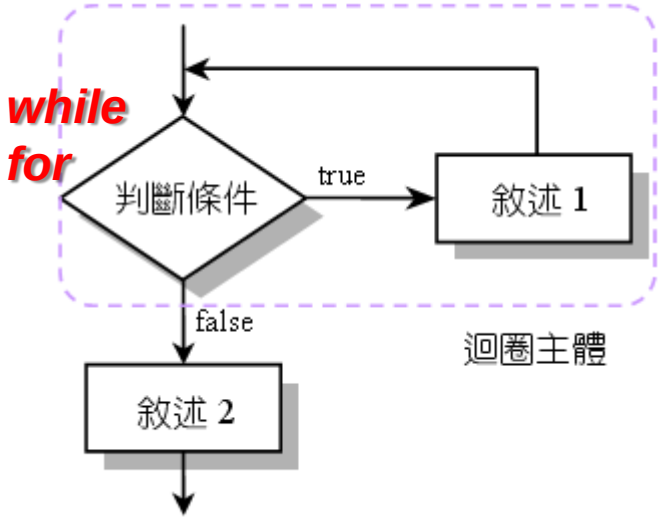
⇩

只需要做一次時!

結構化程式設計

```
while(計數器還沒數到0){
    印出"*";
    ....
}
```

- 重複性結構（iteration structure）
  - 程式在某些敘述區塊反覆執行，直到符合測試條件時才離開




```
while (判斷條件成立)
        do something;
```

⇩

當可能需要做多次時!

使用while 迴圈:
while 最適合用在迴圈執行次數為未知時

NO ";"

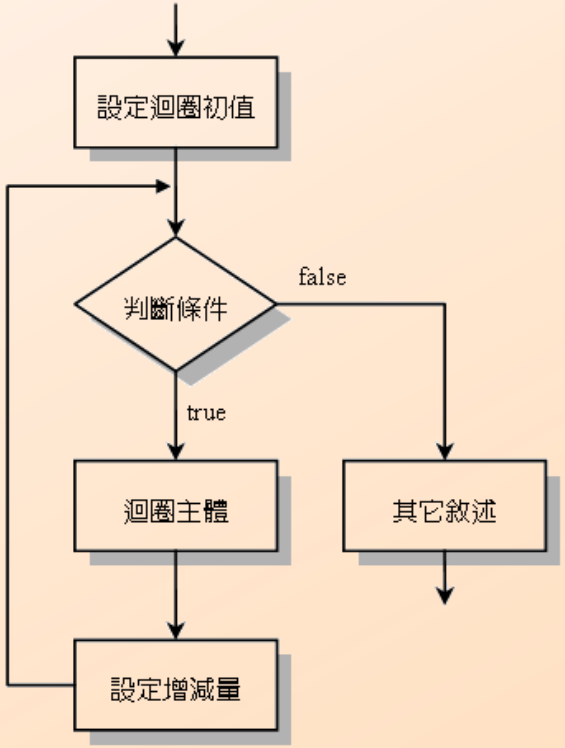
Compiler do not detect this error.
It thinks "end" of while !

TRUE or FALSE

```
設定迴圈初值;
while (判斷條件)
{
    迴圈主體;
    設定增減量;
}
```

這兒不可以加分號

while 迴圈的語法



while迴圈

- 下面列出while迴圈執行的流程
  - 第一次進入while迴圈前，就必須先設定迴圈控制變數的起始值
  - 根據判斷條件的內容，檢查是否要繼續執行迴圈
  - 執行完迴圈主體內的敘述後，重新設定（增加或減少）迴圈控制變數的值，再回到 while 判斷條件

```
設定迴圈初值;
while(判斷條件)  ⟶ 這兒不可以加分號
{
    迴圈主體;
    設定增減量;
}  ⟶ 這兒不可以加分號
```

while loop

Source code

```
#include <stdio.h>
 void main(void)
 {
     int  i;

     i = 1;

     while (i<= 5 )
        {
        printf (" Loop number %d in the while_loop\n",i);
        i++;
        }
 }
```

Test expression.

Statement block is repeatedly executed until test expression becomes False.

Incrementing counter variable.

Output

```
Loop number 1 in the while_loop
Loop number 2 in the while_loop
Loop number 3 in the while_loop
Loop number 4 in the while_loop
Loop number 5 in the while_loop
```

while loop

i: The control counter, increase or decrease, the value should be changed at each iteration. 控制迴圈的次數，遞增或遞減，數量一直改變

5: 控制在5次以內，也可以是變數 **(within 5 times, it can be changed to a variable)**

```
#include <stdio.h>
 void main(void)
 {
     int  i;

     i = 1;

     while (i<= 5 )
        {
        printf (" Loop number %d in the while_loop\n",i);
        i++;
        }
}
```

Test expression.

Statement block is repeatedly executed until test expression becomes False.

Incrementing counter variable.

Output

```
Loop number 1 in the while_loop
Loop number 2 in the while_loop
Loop number 3 in the while_loop
Loop number 4 in the while_loop
Loop number 5 in the while_loop
```

infinite loop

- The program can not end
  - Carefully write the expression and the loop counter
  - Use 「**`break`**」

無窮迴圈的範例

```
/* prog7_4 OUTPUT---
i=1
i=2
i=3
... （無窮迴圈的輸出）
--------------------*/
```

```
01  /* prog7_4, 無窮迴圈的說明 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int i=1;
07
08     while (i > 0)      /* 當i>0時執行while迴圈的主體 */
09        printf("i=%d\n",i++);
10
11     system("pause");
12     return 0;
13  }
```

```
...
icnt = 0;
while(k)
{
  printf("old_k=%2d, ",k);
  sum3 = k++;
  printf("new_k=%2d, sum3=%2d\n",k,sum3);
  icnt++;
  if (icnt>30) break;
}
...
```

It terminate execution of the smallest enclosing switch or iteration statement.

→ 跳出迴圈

```
...
icnt = 0;
while(k && icnt<=31)
{
  printf("old_k=%2d, ",k);
  sum3 -= k++;
  printf("new_k=%2d, sum3=%2d\n",k,sum3);
  icnt++;
}
...
```

“break” is not a good way to write a good program, try not to use it.

不使用break指令的方法

→不使用break指令較好(不破壞程式結構)

怎麼設定while()內的判斷式?

1. 當計數器數值還沒數到0時，我列印hi，計數器數值會減1
2. 從1到100，只顯示40~67之間的基數(包含40, 67)，並計算其和
3. 從3到120顯示3的倍數，例如3, 6, 9, …
4. 當還沒有找到數值5時，使用者繼續輸入數值，直到找到數值5
5. 找出最大數。第一個輸入的數值為後面所要輸入的整數的個數，找到其中最大的數字

while

```
/* prog5_5 OUTPUT---
請輸入整數值:10
1+2+...+10=55
--------------------*/
```

- 下面的例子是利用while迴圈計算1累加到10

```
01  // prog5_5, while 迴圈
02  #include <iostream>
03  #include <cstdlib>
04  using namespace std;
05  int main(void)
06  {
07      int num,i=1,sum=0;
08      cout << "請輸入整數值:";  //改寫成printf(...)
09      cin >> num; //改寫成scanf_s(...)
10      while(i<=num)
11      {
12        sum+=i;
13        i++;
14      }
15      cout << "1+2+...+" << num << "=" << sum << endl;
16      system("pause");
17      return 0;
18  }
```

迴圈範例

```
/* prog5_5 OUTPUT---
請輸入整數值:10
1+2+...+10=55
--------------------*/
```

```
07      int num,i=1,sum=0;

10      while(i<=num)
11      {
12        sum+=i;
13        i++;
14      }
```

i 的值	sum 的值	計算 sum+=i 之後，sum 的值
1	0	1
2	1	3
3	3	6
4	6	10
5	10	15
6	15	21
7	21	28
8	28	36
9	36	45
10	45	55

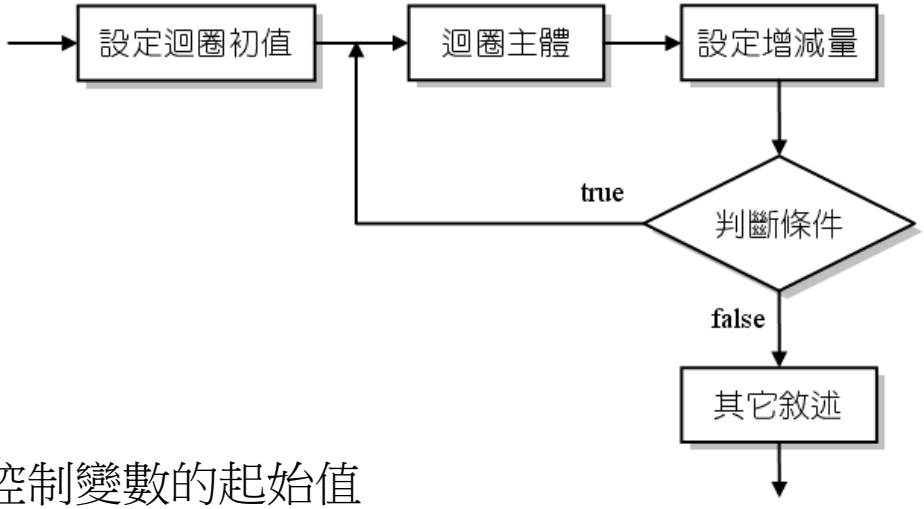
do-while

- Using do-while loops
- Differences between do-while and while loops

do while迴圈

- do while迴圈的格式如下

```
設定迴圈初值;
do
{
    迴圈主體;
    設定增減量;
} while(判斷條件) ;  → 要加分號
```



- 下面列出do while迴圈執行的流程：
  - 進入do while迴圈前，先設定迴圈控制變數的起始值
  - 迴圈主體執行完畢，才開始根據判斷條件的內容，檢查是否繼續執行迴圈
  - 執行完迴圈主體內的敘述後，重新設定（增加或減少）迴圈控制變數的值

do-while loop

- Syntax

```
do
{
   statement1;
   statement2;
   ...
}
while(expression);
```

do while迴圈

```
/* prog5_6 OUTPUT-------
請輸入欲累加的最大奇數值:-6
請輸入欲累加的最大奇數值:7
1+3+...+7=16
-----------------------*/
```

```
01  // prog5_6, do while迴圈
02  #include <iostream>
03  #include <cstdlib>
04  using namespace std;
05  int main(void)
06  {
07      int n,i=1,sum=0;
08      do{
09         cout << "請輸入欲累加的最大奇數值:";
10         cin >> n;
11      }while(n<1 || n%2==0);
12
13      do{
14         sum+=i;
15         i+=2;
16      }while(i<=n);
17      cout << "1+3+...+" << n << "=" << sum << endl;
18
19      system("pause");
20      return 0;
21  }
```

while loop vs. do-while loop

- While loop test expression first, do-while test expression at last
  - do-while execute at least one time
  - while may execute zero time

Debug!

```
int i;
scanf_s("%d", i);
scanf_s("%d\n", &i);
```

Output example:

```
Enter a=10
Enter b= 2
Enter c= 88
Max=88
Min=2
Sum=100
Average=33.33
```

怎樣確認debug時，在哪一個視窗?

不對的方式：

- i+2=5;

“=“號左邊，一定要是變數。WHY?

- a < b;
- a+b;
- a*2+3*c-100;

為什麼這樣寫不可以?

Compiler編譯不會有錯誤。但是，算完之後，對程式的用處是?

哪些用if?哪些用loop?

1. 目前商店正在周年慶折扣，滿1000原有8折的折扣，請問消費金額900, 2500和3300時的付款金額?
2. 從1到100，只顯示40~67之間的基數(包含40, 67)，並計算其和
3. 從3到120顯示3的倍數，例如3, 6, 9, …
4. 計算計程車車資，只需輸入里程數，就可以計算車資。里程數再1500公尺內是80元，每多跑500公尺多5元，不足500公尺以500公尺計算。

for loop

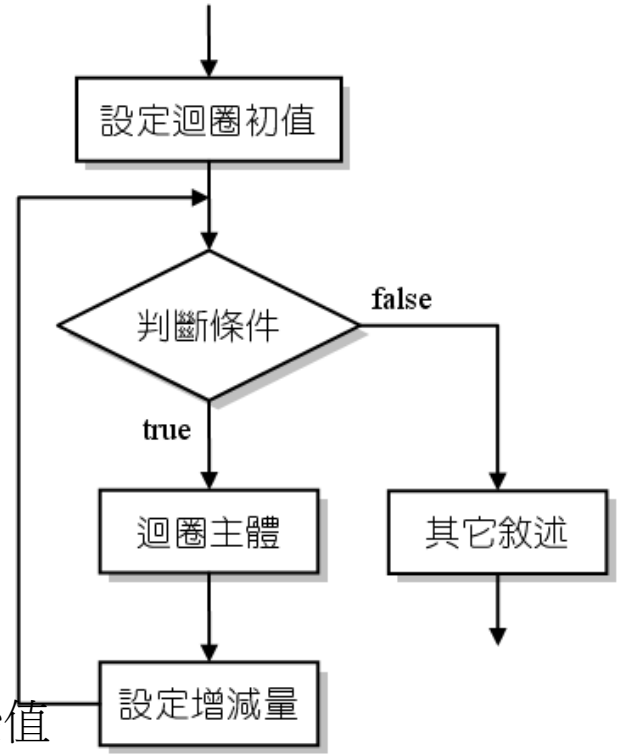
- The for loop control structure
- Structure of a simple for loop
- Difference between for loops and while loops

for迴圈

```
for(設定迴圈初值; 判斷條件; 設定增減量)
{
    迴圈主體;
}
```

這兒不可以加分號

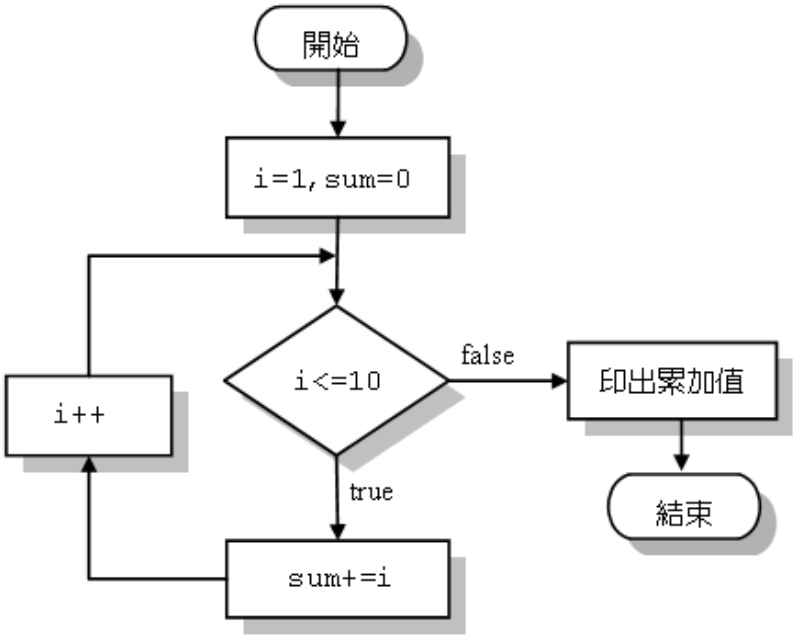
這兒不可以加分號



- for迴圈執行的流程
  - 第一次進入for迴圈時，設定迴圈控制變數的起始值
  - 根據判斷條件的內容，檢查是否要繼續執行迴圈
  - 迴圈控制變數會根據增減量的設定，更改迴圈控制變數的值，再回到上一個步驟重新判斷是否繼續執行迴圈

for迴圈

- 利用for迴圈計算1加到10的總和




```
/* prog7_1, for迴圈的使用 */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
   int i,sum=0;
   for(i=1;i<=10;i++)
      sum+=i;
   printf("1+2+3+...+10=%d\n",sum);

   system("pause");
   return 0;
}
```

```
/* prog7_1 OUTPUT--
1+2+3+...+10=55
---------------------*/
```

	設定迴圈初值		判斷條件		設定增減量	
for(	i=1,sum=0	;	i<=9	;	i+=2	)

```
{
    迴圈主體
}
```

for 迴圈範例

```
06      int i,sum=0;
07      for(i=1;i<=10;i++)
08         sum+=i;
```

表 7.2.1 for 迴圈內，i 與 sum 值變化的情形

i 的值	sum 的值	計算 sum+=i 之後，sum 的值
1	0	1
2	1	3
3	3	6
4	6	10
5	10	15
6	15	21
7	21	28
8	28	36
9	36	45
10	45	55

執行完 for 迴圈之後，sum 的值

UltraEdit-32 - [C:\lect\chap4\c4-10.c]

File Edit Search Project View Format Column Macro Advanced Window Help

c4-10.c

Drive/File List: Refresh

Open Files

Open Files
C:\lect\chap4\c4-10.c

```c
#include <stdio.h>
void main(void)
{
    int day, hour, minutes;

    for(day=1; day<=3; day++)
        printf("Day=%2d\n",day);

    for(hour=5; hour>2; hour--)
    {
        minutes = 60*hour;
        printf("Hour = %2d, Minutes=%3d\n", hour, minutes);
    }
}
```

For Help, press F1 | Ln 15, Col. 1, CW | DOS | Mod: 2001/10/19 09:59:16AM | File Size: 275 | INS

開始 | Microsoft PowerPoint - [Le... | UltraEdit-32 - [C:\lect\chap... | MS-DOS 模式 | En | AM 09:59

```
for (loop_expression)
    single statement for_loop_body;
```

separated from each other by semicolons ;

```
for(day=1; day<=3; day++)
      printf("Day=%2d\n",day);
```

initialize the for loop control variables

serve as a test expression

increase or decrease the control variable

Example of for loops

- The following examples show methods of varying the control variable in a `for` statement.
  - Vary the control variable from 1 to 100 in increments of 1.

    ```
    for ( i = 1; i <= 100; i++ )
    ```

  - Vary the control variable from 100 to 1 in increments of -1 (decrements of 1).

    ```
    for ( i = 100; i >= 1; i-- )
    ```

  - Vary the control variable from 7 to 77 in steps of 7.

    ```
    for ( i = 7; i <= 77; i += 7 )
    ```

  - Vary the control variable from 20 to 2 in steps of -2.

    ```
    for ( i = 20; i >= 2; i -= 2 )
    ```

  - Vary the control variable over the following sequence of values: 2, 5, 8, 11, 14, 17.

    ```
    for ( j = 2; j <= 17; j += 3 )
    ```

  - Vary the control variable over the following sequence of values: 44, 33, 22, 11, 0.

    ```
    for ( j = 44; j >= 0; j -= 11 )
    ```

Counter variable

- It's value could be changed in the body for a loop.
  - Not recommend.

```
for (k=1; k<3; k++) k=1;
```

導致無窮迴圈 (Infinite loops)

The differences between for / while loops

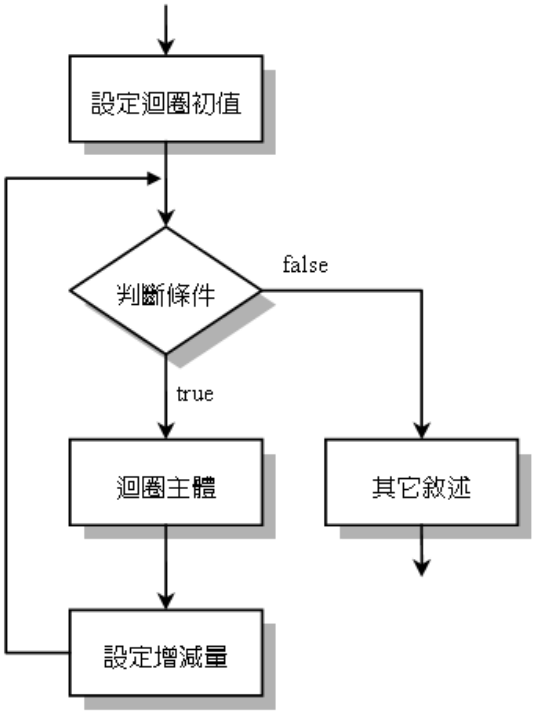
Item	For loop	While loop
Initialization expression	Is one of the loop expressions	Must be given prior to the loop
Test expression	Is one of the loop expressions	Is one of the loop expression
Increment expression	Is one of the loop expressions	Must be in the loop body
When number of iterations is known	Is very convenient and clear to use	Is less convenient and clear
When number of iterations is unknown	Is less convenient and clear	Is more convenient than for loop

for與while迴圈的比較

表 7.3.1　for 迴圈與 while 迴圈的敘述比較

for 迴圈	while 迴圈

for(設定初值;判斷條件;設定增減量)
{
　　敘述 1;
　　敘述 2;
　　⋮
　　敘述 n;
}
	設定初值;
	while(判斷條件)
	{
	　　敘述 1;
	　　敘述 2;
	　　⋮
	　　敘述 n;
	　　設定增減量
	}



空迴圈的範例

```
/* prog7_8 OUTPUT--
i=10001
----------------------*/
```

- 下面的範例是一個不做任何事的空迴圈：

```
01  /* prog7_8, 空迴圈的誤用 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int i;
07     for(i=1;i<=10000;i++);      /* 空迴圈 */
08        printf("i=%d\n",i);
09
10     system("pause");
11     return 0;
12  }
```

Unknown number of loops by using for

- Infinite loops

```
for (;;)
```

- **break**

```
for (;;)
{
  printf("Please type a number, type 0 to quit\n");
  scanf("%d", &number);
  if (number==0) break;
}
```

for vs. while

- The general format of the `for` statement is

```
for ( expression1; expression2; expression3 )
      statement
```

- In most cases, the `for` statement can be represented with an equivalent `while` statement as follows:

```
expression1;
while ( expression2 ) {
      statement
      expression3;
}
```

使用哪一種迴圈？

表 7.6.1　for、while 與 do while 迴圈的比較

迴圈特性	迴圈種類		
	for	while	do while
前端測試判斷條件	是	是	否
後端測試判斷條件	否	否	是
於迴圈主體中需要更改控制變數的值	否	是	是
迴圈控制變數會自動變更	是	否	否
迴圈重複的次數	已知	未知	未知
至少執行迴圈主體的次數	0 次	0 次	1 次
何時重複執行迴圈	條件成立	條件成立	條件成立

迴圈的跳離

- break 敘述：
  - 略過迴圈主體的其餘部分，執行迴圈之後的敘述

break 敘述的語法

```
for(初值設定; 判斷條件; 設定增減量)
{
    敘述 1;
    敘述 2;
    …
    break;
    …
    敘述 n;
}
...
```

若執行break敘述，則此區塊內的敘述不會被執行

continue 敘述 (e.g.,下一個 i 值)

- continue 敘述：
  - 略過迴圈主體的其餘部分，直接開始下一個迴圈循環

continue 敘述的語法

```
for(初值設定; 判斷條件; 設定增減量)
{
    敘述 1;
    敘述 2;
      …
    continue;
      …
    敘述 n;
}
  ...
```

若執行continue敘述，則此區塊內的敘述不會被執行

for vs. while : add 1 to 10

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

void main(void)
{
        int sum=0;
        int i;

        i=1;
        while(i<=10){
                sum += i;
                i++;
        }

        printf("sum=%d\n", sum);
        system("pause");
}
```

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

void main(void)
{
        int sum=0;
        int i;

        for(i=1;i<=10;i++)
                sum +=i;

        printf("sum=%d\n", sum);
        system("pause");
}
```

```
sum=55
請按任意鍵繼續 . . .
```

A Practice

- Enter a number, add from 1 to the number
- E.g.

```
Enter a number=5
1 + 2 + 3 + 4 + 5 = 15
請按任意鍵繼續 . . .
```

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

void main(void)
{
        int sum=0;
        int i;
        int count;

        printf("Enter a number=");
        scanf("%d", &count);

        for(i=1;i<=count;i++){
                sum += i;
                printf("%d", i);
                if(          )
                        printf(" + ");
        }

        printf(" = %d\n", sum);
        system("pause");
}
```

Nested for loops

- Using the **+=** type operator in an increment expression
- Nested for loops

UltraEdit-32 - [C:\lect\chap4\l4-11.c]

File Edit Search Project View Format Column Macro Advanced Window Help

l4-10.c l4-11.c

Drive/File List: Refresh

Open Files

- Open Files
  - C:\lect\chap4\l4-10.c
  - C:\lect\chap4\l4-11.c

```c
#include <stdio.h>
void main(void)
{
    int i,j,k, m=0;

    for(i=1; i<=5; i+=2)
    {
        for(j=1; j<=4; j++)
        {
            k = i+j;
            printf("i=%3d, j=%3d, k=%3d\n", i, j, k);
        }
        m=k+1;
    }
}
```

For Help, press F1 | Ln 6, Col. 1, CW | DOS | Mod: 2001/10/19 11:06:50AM | File Size: 233 | INS

開始 | Microsoft PowerPoint - [... | UltraEdit-32 - [C:\lect\ch... | MS-DOS 模式 | 我的電腦 | En | AM 11:07

UltraEdit-32 - [C:\lect\chap4\l4-11.c]

File Edit Search Project View Format Column Macro Advanced Window Help

l4-10.c l4-11.c

Drive/File List: Refresh

Open Files

- Open Files
  - C:\lect\chap4\l4-10.c
  - C:\lect\chap4\l4-11.c

```c
#include <stdio.h>
void main(void)
{
    int i,j,k, m=0;

    for(i=1; i<=5; i+=2)
    {
        for(j=1; j<=4; j++)
        {
            k = i+j;
            printf("i=%3d, j=%3d, k=%3d\n", i, j, k);
        }
        m=k+1;
    }
}
```

inner for loop

outer for loop

For Help, press F1 | Ln 8, Col. 1, CW | DOS | Mod: 2001/10/19 11:06:50AM | File Size: 233 | INS

開始 | Microsoft PowerPoint - [... | UltraEdit-32 - [C:\lect\ch... | MS-DOS 模式 | 我的電腦 | En | AM 11:08

Practice – 九九乘法表

```
1 *  1 =  1
1 *  2 =  2
1 *  3 =  3
1 *  4 =  4
1 *  5 =  5
1 *  6 =  6
1 *  7 =  7
1 *  8 =  8
1 *  9 =  9

2 *  1 =  2
2 *  2 =  4
2 *  3 =  6
2 *  4 =  8
2 *  5 = 10
2 *  6 = 12
2 *  7 = 14
2 *  8 = 16
2 *  9 = 18

3 *  1 =  3
3 *  2 =  6
3 *  3 =  9
3 *  4 = 12
3 *  5 = 15
3 *  6 = 18
3 *  7 = 21
3 *  8 = 24
3 *  9 = 27

4 *  1 =  4
4 *  2 =  8
4 *  3 = 12
4 *  4 = 16
4 *  5 = 20
4 *  6 = 24
4 *  7 = 28
4 *  8 = 32
4 *  9 = 36
```

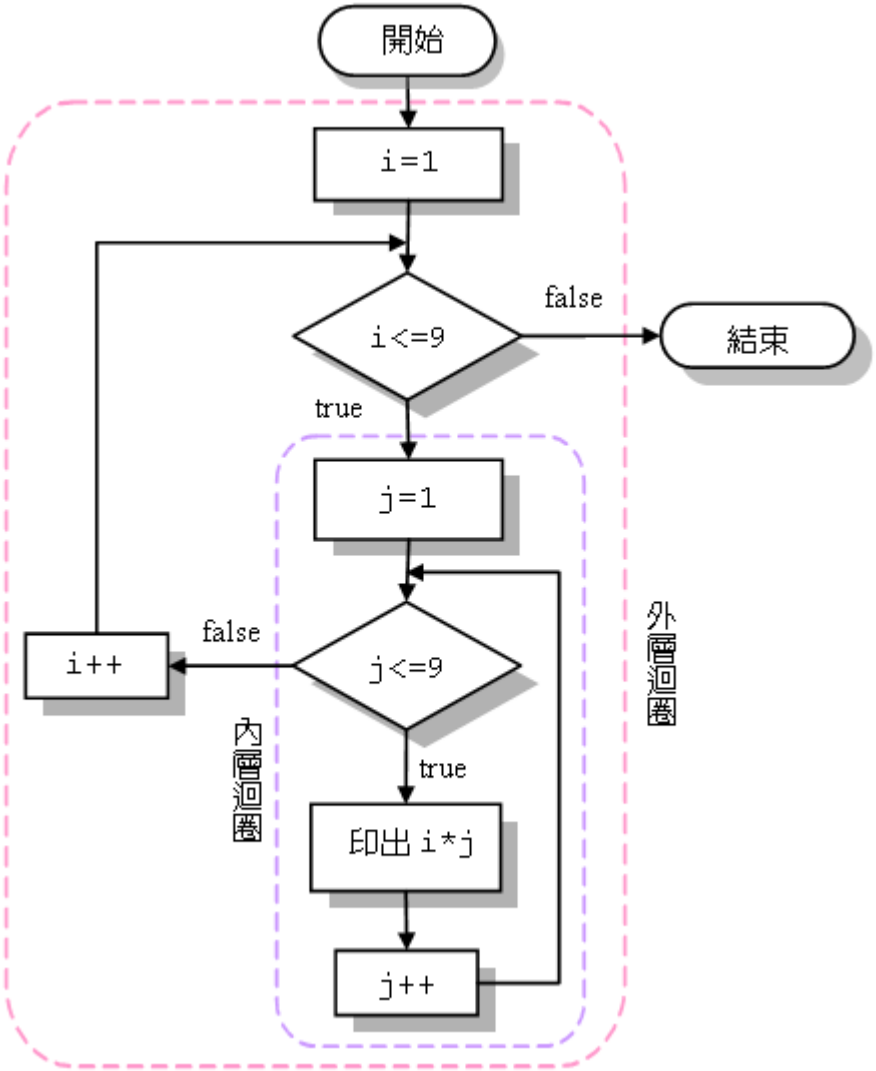
```
1 *  1 =  1     2 *  1 =  2     3 *  1 =  3
1 *  2 =  2     2 *  2 =  4     3 *  2 =  6
1 *  3 =  3     2 *  3 =  6     3 *  3 =  9
1 *  4 =  4     2 *  4 =  8     3 *  4 = 12
1 *  5 =  5     2 *  5 = 10     3 *  5 = 15
1 *  6 =  6     2 *  6 = 12     3 *  6 = 18
1 *  7 =  7     2 *  7 = 14     3 *  7 = 21
1 *  8 =  8     2 *  8 = 16     3 *  8 = 24
1 *  9 =  9     2 *  9 = 18     3 *  9 = 27

4 *  1 =  4     5 *  1 =  5     6 *  1 =  6
4 *  2 =  8     5 *  2 = 10     6 *  2 = 12
4 *  3 = 12     5 *  3 = 15     6 *  3 = 18
4 *  4 = 16     5 *  4 = 20     6 *  4 = 24
4 *  5 = 20     5 *  5 = 25     6 *  5 = 30
4 *  6 = 24     5 *  6 = 30     6 *  6 = 36
4 *  7 = 28     5 *  7 = 35     6 *  7 = 42
4 *  8 = 32     5 *  8 = 40     6 *  8 = 48
4 *  9 = 36     5 *  9 = 45     6 *  9 = 54

7 *  1 =  7     8 *  1 =  8     9 *  1 =  9
7 *  2 = 14     8 *  2 = 16     9 *  2 = 18
7 *  3 = 21     8 *  3 = 24     9 *  3 = 27
7 *  4 = 28     8 *  4 = 32     9 *  4 = 36
7 *  5 = 35     8 *  5 = 40     9 *  5 = 45
7 *  6 = 42     8 *  6 = 48     9 *  6 = 54
7 *  7 = 49     8 *  7 = 56     9 *  7 = 63
7 *  8 = 56     8 *  8 = 64     9 *  8 = 72
7 *  9 = 63     8 *  9 = 72     9 *  9 = 81

請按任意鍵繼續 . . .
```

九九乘法表



程式湊湊看?

還是一點一點地分析步驟?

An example: while

- Input an odd number n, compute the value of 1 + 3 + 5 +… + n
  1. Check whether n is an odd number, please check until input correctly
  2. Output as the example
- 輸入正整數奇數 n,求1 + 3 + 5 +… + n 的值
  1. 檢查是否為奇數(若非奇數，則重新輸入，直至輸入數字為奇數) (請用一個while來檢查)(或do while)
  2. 印出如下結果 (請用while來做加法，並印出過程和結果)

為了程式正確性，

請自行驗證其他數值

Output example 紅字部分請用變數

Enter an odd number=2

Error!

Enter an odd number=11

Sum:1+3+5+7+9+11=36

An example:輾轉相除法(求最大公因數) Flow chart

- 利用輾轉相除法，求num1, num2的最大公因數
  - 餘數求法: e.g. 12 % 7 = 5, 10 % 3 = 1
  - 兩數
    (1) 大數除以小數，求得餘數
    (2) 再以其中小數當成大數，餘數為小數，求餘數
    (3) 直至餘數為零，此時的小數為其gcd

Output example

```
Enter two numbers:
First number=1071
Second number=462
gcd(1071, 462)=21
```

輸入正整數奇數 n, 求1 + 3 + 5 +... + n 的值

Enter an odd number=11

Sum:1+3+5+7+9+11=36

```
4   void main(void)
5   {
6       int num = 11;
7       int i = 1;
8       int sum = 0;
9
10      printf("Sum:");
11      while (          ){
12
13
14
15
16
17
18      }
19      printf("%d\n", sum);
```

輸入正整數奇數 n, 求1 + 3 + 5 +... + n 的值

Enter an odd number=11

Sum:1+3+5+7+9+11=36

```
void main(void)
{
    int num = 11;
    int i = 1;
    int sum = 0;

    printf("Sum:");
    while (i <= num){

        sum += i;
        i += 2;
    }
    printf("%d\n", sum);
```

輸入正整數奇數 n, 求1 + 3 + 5 +... + n 的值

Enter an odd number=11

Sum:1+3+5+7+9+11=36

```c
void main(void)
{
    int num = 11;
    int i = 1;
    int sum = 0;

    printf("Sum:");
    while (i <= num){
        if (i!=num)
            printf("%d+", i);
        else
            printf("%d=", i);
        sum += i;
        i += 2;
    }
    printf("%d\n", sum);
```

More about while !!

輸入一個數字，直到輸入的數字大於0 (while) (1)

Enter a positive number: -5

-5 is not a positive number!

Enter a positive number: 0

0 is not a positive number!

Enter a positive number: -100

-100 is not a positive number!

Enter a positive number: 9

9 is a positive number.

```c
void main(void)
{
    int i;

    printf("Enter a positive number:");
```

```c
void main(void)
{
    int i;

    printf("Enter a positive number:%d", i);
```

?

輸入一個數字，直到輸入的數字大於0 (while) (2)

Enter a positive number: -5

-5 is not a positive number!

Enter a positive number: 0

0 is not a positive number!

Enter a positive number: -100

-100 is not a positive number!

Enter a positive number: 9

9 is a positive number.

```
void main(void)
{
    int i;

    printf("Enter a positive number:");
    scanf_s("%d", &i);
```

哪些程式結果：
如果輸入錯誤，要重複做的事情? (3)

Enter a positive number: -5

-5 is not a positive number!

Enter a positive number: 0

0 is not a positive number!

Enter a positive number: -100

-100 is not a positive number!

Enter a positive number: 9

9 is a positive number.

```
void main(void)
{
    int i;

    printf("Enter a positive number:");
    scanf_s("%d", &i);

    while (      )
    {
        printf("%d is not a positive number!\n", i);
        printf("Enter a positive number:");
        scanf_s("%d", &i);
    }
```

檢測?

所做事項?

跳出迴圈做的事? (4)

Enter a positive number: -5

-5 is not a positive number!

Enter a positive number: 0

0 is not a positive number!

Enter a positive number: -100

-100 is not a positive number!

Enter a positive number: 9

9 is a positive number.

```
4   void main(void)
5   {
6       int i;
7
8       printf("Enter a positive number:");
9       scanf_s("%d", &i);
10
11      while (i <= 0)
12      {
13          printf("%d is not a positive number!\n"
14          printf("Enter a positive number:");
15          scanf_s("%d", &i);
16      }
17
18      printf("%d is a positive number.\n", i);
```

跳出迴圈?

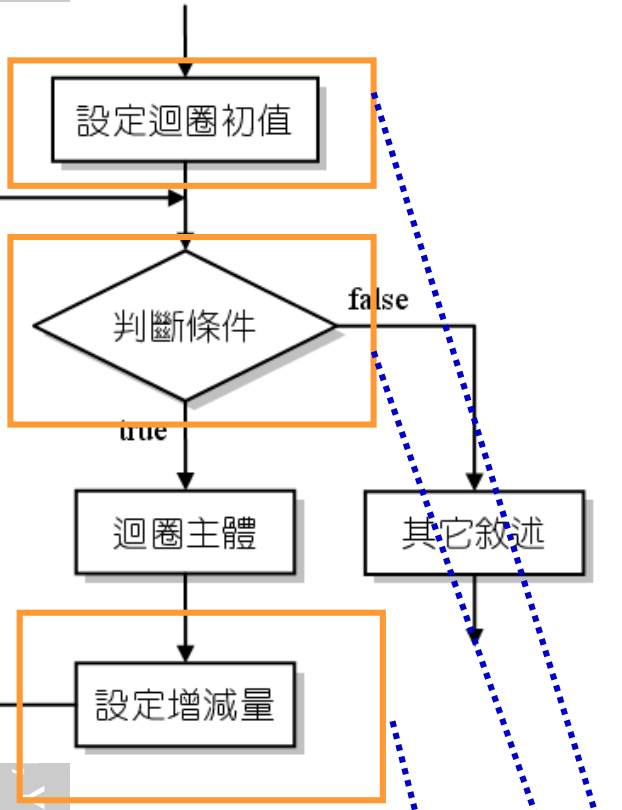
試撰寫一程式，印出從1到100之間，所有可以被18整除的數值。(while, if)

- **Step1:**從**1**到**100**之間 **➔while?**
- **Step2:** 印所有可以被**18**整除的數值**➔if?**

Program:

```
int i=0;
while(i>=1 && i<=100 && i%18==0)
{
    printf("%d\n", i);
}
```

假設有一條繩子長3000公尺，每天剪去一半的長度，請問需要花費幾天的時間，繩子的長度會短於5公尺？



- 甚麼是:判斷條件?

  (做檢查用!!一定是關係運算式)

  – 還沒有<5公尺時，進迴圈

- 甚麼是：迴圈主體?

  – 每天剪去一半的長度

  – 用變數紀錄“天數”

```
4   void main(void)
5   {
6       float length = 3000;
7       int count = 0;
8
9       while (length >= 5.0) {
10          length = length / 2;
11          count++;
12      }
13
14      printf("Total %d times\n", count);
```

要不要有同樣的變數?