

迴圈敘述

5

CHAPTER

迴圈敘述（loop statement）的功能就是可以重複執行敘述。迴圈敘述主要有 for、while 和 do...while 三種不同語法迴圈敘述。

5-1 for 迴圈

我們先來介紹 for 迴圈，其語法如下：

```
for (初值設定; 條件運算式; 更新運算式) {  
    條件運算式為真時，執行的主體敘述;  
}
```

此迴圈由三個部分所組成，分別為初值設定、條件運算式及更新運算式，茲說明如下：

for 迴圈	功 用
初值設定	設定變數的初值。
條件運算式	若為真，則執行其所對應的敘述；若為假，則結束迴圈。
更新運算式	更新變數值。

此三部份是用分號 (;) 隔開，而不是逗號 (,)。注意，for 敘述小括弧 () 的後面不可加分號 (;)，若執行敘述的部份為單一敘述時，則大括弧 {} 是可以省略的，若是複合敘述時，則不可以省略。但有些人不管單一敘述或是複合敘述，皆加上大括號，因為以後較易維護，如範例 5-1a 所示。

範例 5-1a

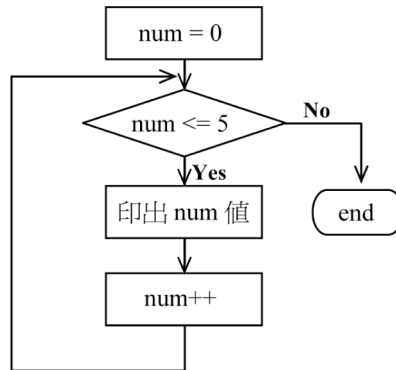
```
01  /* ex5-1a.c */
02  #include <stdio.h>
03  int main()
04  {
05      int num;
06      for (num=0; num<=5; num++) {
07          printf("Num is %d\n", num);
08      }
09      return 0;
10  }
```

輸出結果

```
Num is 0
Num is 1
Num is 2
Num is 3
Num is 4
Num is 5
```

【程式剖析】

第 6 行 for 迴圈先將 num 的初值設為 0，判斷 num 是否小於等於 5，若是，則印出 num 值（第 7 行）。之後，再對 num 加 1，並判斷 num 是否小於等於 5。若是，則印出 num 值。週而復始，直到 num 不是小於等於 5，結束迴圈為止。注意！迴圈結束時，num 的值是 6。以流程圖表示如下：



以下我們來探討 for 迴圈的一些變形寫法。程式中 num 變數的初值設定，可以在定義變數就給定之，此時 for 迴圈內的初值設定敘述就可以省略，但其後面的「;」不可省略。如範例 5-1b 第 6 行所示。

範例 5-1b

```
01  /* ex5-1b.c */
02  #include <stdio.h>
03  int main()
04  {
05      int num = 0; /* 設定 num 初值為 0 */
06      for ( ; num<=5; num++) { /* 省略初值的設定 */
07          printf("Num is %d\n", num);
```


輸出結果

```
Num is 0
Num is 1
Num is 2
Num is 3
Num is 4
Num is 5
```

【程式剖析】

此程式在定義變數時，也同時設定其初值。

其次，也可以將 for 迴圈的更新運算式移至他處，如範例 5-1c 第 8 行所示。

 範例 5-1c

```
01  /* ex5-1c.c */
02  #include <stdio.h>
03  int main()
04  {
05      int num = 0;
06      for ( ; num<=5; ) {
07          printf("Num is %d\n", num);
08          num++;
09      }
10      return 0;
11  }
```


輸出結果

```
Num is 0
Num is 1
Num is 2
Num is 3
Num is 4
Num is 5
```

【程式剖析】

for 迴圈的更新運算式移至執行的主體敘述內，但不可以省略前面的分號。輸出結果同上一範例。

與上一章的 if...else 敘述一樣，當 for 迴圈所要執行的主體敘述是複合敘述（指兩個或兩個以上的敘述），必須以左、右大括號括起來。上一範例的 for 迴圈若沒加左、右大括號，將會形成無窮迴圈，如範例 5-1d 第 6-8 行所示。

 範例 5-1d

```
01  /* ex5-1d.c */
02  #include <stdio.h>
03  int main()
04  {
05      int num = 0;
06      for ( ; num<=5; )
07          printf("Num is %d\n", num);
08          num++;
09      return 0;
10  }
```

輸出結果

這是一無窮迴圈，請您親身體驗一下。

【程式剖析】

因為 for 迴圈只對它的下一條敘述有效而已，所以並沒有執行 num++; 敘述，導致 num 都是 0，因此，形成了無窮迴圈。

在 for 迴圈中，若省略條件運算式，其結果將會如何？因為 C 語言認為人性本善，所以視此條件永遠為真，導致形成無窮迴圈，此時必須利用 break 敘述來終止迴圈的執行，如範例 5-1e 所示。有關 break 敘述的詳細說明，請參閱 5-5 節。



範例 5-1e

```
01  /* ex5-1e.c */
02  #include <stdio.h>
03  int main()
04  {
05      /* 將範例 5-1a 的設定初值、條件判斷式及更新運算式省略 */
06      int num = 0;
07      for ( ; ; ) {
08          if (num > 5) /* 若 num 大於 5，則執行 break，以結束 for 迴圈 */
09              break;
10          printf("Num is %d\n", num);
11          num++;
12      }
13      return 0;
14  }
```

輸出結果

```
Num is 0
Num is 1
Num is 2
Num is 3
Num is 4
Num is 5
```

【程式剖析】

在第 7 行 for 迴圈中，省略了條件判斷式，表示其條件為真，所以是一無窮迴圈。此時必須利用第 8 行 if 敘述，判斷 num 是否大於 5，若是，則執行第 9 行 break，以結束迴圈。

上述幾個範例都已給定迴圈所要執行的次數，此稱為**定數迴圈**。若沒有給定所要執行次數，則稱之為**不定數迴圈**，這種迴圈由使用者決定是否要繼續執行。範例 5-1f 就是一典型不定數迴圈的寫法。



範例 5-1f

```
01  /* ex5-1f.c */
02  #include <stdio.h>
03  int main()
04  {
05      /* 要求使用者不斷輸入數字，直到數字為負 */
06      int num = 0;
07      for ( ; num>=0; ) {
08          printf("請輸入一整數(當此數為負時，將結束迴圈): ");
09          scanf("%d", &num);
10      }
11      printf("輸入結束!!");
12      return 0;
13  }
```

輸出結果

```
請輸入一整數(當此數為負時，將結束迴圈): 77
請輸入一整數(當此數為負時，將結束迴圈): 30
請輸入一整數(當此數為負時，將結束迴圈): 25
請輸入一整數(當此數為負時，將結束迴圈): 66
請輸入一整數(當此數為負時，將結束迴圈): 12
請輸入一整數(當此數為負時，將結束迴圈): -20
輸入結束!!
```

【程式剖析】

在第 7 行 for 迴圈中只有條件運算式，其它都省略了，當輸入 num 的數值為負時，將結束迴圈。

5-7-8 計算以下的公式值

試撰寫一程式，用以計算下列公式的值。請參閱應用範例 5-7-8。

$$\frac{1}{1+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + \frac{1}{\sqrt{3}+\sqrt{4}} + \dots + \frac{1}{\sqrt{999}+\sqrt{1000}}$$



應用範例 5-7-8

```

01  /* ex5-7-8.c */
02  #include <stdio.h>
03  #include <math.h>
04  int main()
05  {
06      int i;
07      double sum = 0.0;
08      for (i=1; i<=999; i++) {
09          sum += 1/(sqrt(i)+sqrt(i+1));
10      }
11      printf("sum = %.3f\n", sum);
12      return 0;
13  }
```

輸出結果

```
sum = 30.623
```

5-7-9 將公哩轉換為公里

試撰寫一程式，將 50~100 公哩（mile）轉換為公里（km）。提示：1 公哩約等於 1.6 公里。請參閱應用範例 5-7-9。



應用範例 5-7-9

```

01  /* ex5-7-9 */
02  #include <stdio.h>
03  int main() {
04      int i;
05      double km;
06      for (i=50; i<=100; i+=5) {
07          km = i * 1.6;

```

```
08     printf("%3d 公哩等於 %.2f\n", i, km);
09     }
10     return 0;
11 }
```

輸出結果

```
50 公哩等於 80.00
55 公哩等於 88.00
60 公哩等於 96.00
65 公哩等於 104.00
70 公哩等於 112.00
75 公哩等於 120.00
80 公哩等於 128.00
85 公哩等於 136.00
90 公哩等於 144.00
95 公哩等於 152.00
100 公哩等於 160.00
```

5-7-10 將攝氏溫度轉換為華氏溫度

試撰寫一程式，將 0~100 攝氏溫度(C)轉換為華氏溫度(F)。提示：攝氏溫度轉換為華氏溫度的公式如下：

$$F = C * (9/5) + 32$$

請參閱應用範例範例 5-7-10。



應用範例 5-7-10

```
01  /* ex5-7-10.c */
02  #include <stdio.h>
03  int main() {
04      int i;
05      double F;
06      for (i=0; i<=100; i+=5) {
07          F = i * (9./5) + 32;
08          printf("攝氏溫度%3d 等於F華氏溫度 %.2f\n", i, F);
09      }
10      return 0;
11 }
```


輸出結果

攝氏溫度	0	等於 F	華氏溫度	32.00
攝氏溫度	5	等於 F	華氏溫度	41.00
攝氏溫度	10	等於 F	華氏溫度	50.00
攝氏溫度	15	等於 F	華氏溫度	59.00
攝氏溫度	20	等於 F	華氏溫度	68.00
攝氏溫度	25	等於 F	華氏溫度	77.00
攝氏溫度	30	等於 F	華氏溫度	86.00
攝氏溫度	35	等於 F	華氏溫度	95.00
攝氏溫度	40	等於 F	華氏溫度	104.00
攝氏溫度	45	等於 F	華氏溫度	113.00
攝氏溫度	50	等於 F	華氏溫度	122.00
攝氏溫度	55	等於 F	華氏溫度	131.00
攝氏溫度	60	等於 F	華氏溫度	140.00
攝氏溫度	65	等於 F	華氏溫度	149.00
攝氏溫度	70	等於 F	華氏溫度	158.00
攝氏溫度	75	等於 F	華氏溫度	167.00
攝氏溫度	80	等於 F	華氏溫度	176.00
攝氏溫度	85	等於 F	華氏溫度	185.00
攝氏溫度	90	等於 F	華氏溫度	194.00
攝氏溫度	95	等於 F	華氏溫度	203.00
攝氏溫度	100	等於 F	華氏溫度	212.00

5-7-11 使用 while 迴圈敘述求 n^3 大於等於 15000 的最小整數

試撰寫一程式，利用 while 迴圈敘述，求出 n^3 大於等於 15000 的最小整數。
請參閱應用範例 5-7-11。



應用範例 5-7-11

```
01  /* ex5-7-11.c */
02  #include <stdio.h>
03  int main()
04  {
05      int n=1;
06      while (n*n*n < 15000) {
07          n++;
08      }
09      printf("%d 的 3 次方為大於等於 15000 的最小整數 \n", n);
10      return 0;
11  }
```

輸出結果

25 的 3 次方為大於等於 15000 的最小整數

5-8 問題演練

1. 試問下一程式的輸出結果為何？

```
#include <stdio.h>
int main()
{
    int i = 0;
    while (i < 3 ) {
        switch(i++) {
            case 0: printf("Merry");
            case 1: printf("Merr");
                    break;
            case 2: printf("Mer");
                    default: printf("Christmas");
        }
    }
    return 0;
}
```

2. 試問下一程式的輸出結果為何？請以下列資料測試之 1、1、1、2、3、0、1、4、1、3。

```
#include <stdio.h>
int main()
{
    int i = 0, num = 0, total = 0;
    printf("input 10 numbers, or type 0 to quit\n");
    while (i++ < 10) {
        printf(" %2d:", i);
        scanf("%d", &num);
        if (num == 0)
            break;
        total = total + num;
    }
    printf("total = %d\n", total);
    return 0;
}
```

3. 試問下一程式的輸出結果為何？請以下列資料測試之 1、1、1、2、3、0、1、4、1、3。

```
#include <stdio.h>
int main()
{
    int i = 0, num = 0, total = 0;
    printf("input 10 numbers, or type 0 to continue\n");
    while (i++ < 10) {
        printf(" %2d:", i);
        scanf("%d", &num);
        if (num == 0)
            continue;
        total = total + num;
    }
    printf("total = %d\n", total);
    return 0;
}
```

4. 試問以下程式的輸出結果為何？

```
#include <stdio.h>
int main()
{
    int i, num, total=0;
    printf("累加正整數\n");
    for(i=1; i<=10; i++) {
        printf("請輸入#%d 個整數: ", i);
        scanf("%d", &num);
        if( num >= 0)
            total += num;
        else
            continue;
    }
    printf("Total=%d\n", total);
    return 0;
}
```

5. 試問下列片段程式的輸出結果為何？

(a)

```
int x=0;
while (++x < 3)
    printf("%4d\n", x);
```

(b)

```
int x=100;
while (x++ < 103)
    printf("%4d\n", x);
printf("%4d\n", x);
```

```
(c) char ch='s';
    while (ch < 'w') {
        printf("%c", ch);
        ch++;
    }
    printf("%c\n", ch);
```

6. 以下有二個程式，請回答其問題。

```
(a) #include <stdio.h>
int main()
{
    int i, i1=0, i2=0, i3=0, i4=0, others=0, k;
    for (k=1; k<=5; k++){
        printf("Please input a number : ");
        scanf("%d", &i);
        switch(i) {
            case 1: i1++;
                    break;
            case 2: i2++;
                    break;
            case 3: i3++;
                    break;
            case 4: i4++;
                    break;
            default: others++;
        }
        printf("i1=%d, i2=%d, i3=%d, i4=%d,
               others=%d\n", i1, i2, i3, i4, others);
    }
    return 0;
}
```

試問分別輸入 1、2、3、4、5 的結果為何？

```
(b) #include <stdio.h>
int main()
{
    int i, i1=0, i2=0, i3=0, i4=0, others=0, k;
    for (k=1; k<=5; k++) {
        printf("Please input a number:");
        scanf("%d", &i);
        switch(i) {
            case 1: i1++;
            case 2: i2++;
                    break;
        }
    }
}
```

```

        case 3: i3++;
        case 4: i4++;
                break;
        default: others++;
    }
    printf("i1=%d,i2=%d,i3=%d,i4=%d,others=%d\n",i1,i2,
        i3,i4,others);
    }
    return 0;
}

```

試問分別輸入 1、2、3、4、5 的結果為何？

5-9 程式實作

1. 利用迴圈敘述，印出以下的圖形。

(a) *

```

***
*****
*****
*****
*****
*****
*****
***
*

```

(b) *

```

**
***
****
*****
*****
*****
*****
*****

```

2. 先要求使用者輸入等差數列的首項與公差，再輸入項數，最後利用迴圈敘述計算該數列的和。
3. 利用迴圈敘述，印出攝氏溫度與華氏溫度的對照表。從攝氏溫度 -50° 到 100° ，每間隔 10° 印出（提示：華氏溫度 = $9/5 \times$ 攝氏溫度 + 32.0）。
4. 印出費氏數列 (Fibonacci) 的前 15 次（提示：某一項是其前兩項之和，第一項為 1，第二項為 1）。
5. 以 10 進位印出 11×11 之乘法表，輸出格式如下所示。

```

<< Multiplication table 11 * 11 >>
1
2   4
3   6   9
4   8  12  16
5  10  15  20  25

```

6	12	18	24	30	36					
7	14	21	28	35	42	49				
8	16	24	32	40	48	56	64			
9	18	27	36	45	54	63	72	81		
10	20	30	40	50	60	70	80	90	100	
11	22	33	44	55	66	77	88	99	110	121

6. 試撰寫一程式，以下列公式計算 π 值。

$$\pi = 4(1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \dots + (-1)^{i+1}/(2i - 1))$$

試以 i 等於 10000、20000、30000， \dots ，以及 200000 時，其 π 值為何？
(註：系統內建的 $\pi = 3.141592653589793$)

7. 試撰寫一程式，依下表求分數所對應的 GPA (Grade Point Average)。以不定數迴圈提示使用者輸入其分數，之後程式將顯示其 GPA。注意，當分數為負數時，則結束程式執行。

分數	GPA
80~100 分	A
70~79 分	B
60~69 分	C
50~59 分	D
49 分以下	F

8. 試撰寫一程式，印出以下的九九乘法表。

```
*****
1*1= 1  2*1= 2  3*1= 3  4*1= 4  5*1= 5  6*1= 6  7*1= 7  8*1= 8  9*1= 9
1*2= 2  2*2= 4  3*2= 6  4*2= 8  5*2=10  6*2=12  7*2=14  8*2=16  9*2=18
1*3= 3  2*3= 6  3*3= 9  4*3=12  5*3=15  6*3=18  7*3=21  8*3=24  9*3=27
1*4= 4  2*4= 8  3*4=12  4*4=16  5*4=20  6*4=24  7*4=28  8*4=32  9*4=36
1*5= 5  2*5=10  3*5=15  4*5=20  5*5=25  6*5=30  7*5=35  8*5=40  9*5=45
1*6= 6  2*6=12  3*6=18  4*6=24  5*6=30  6*6=36  7*6=42  8*6=48  9*6=54
1*7= 7  2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49  8*7=56  9*7=63
1*8= 8  2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64  9*8=72
1*9= 9  2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81
*****
```