

迴圈

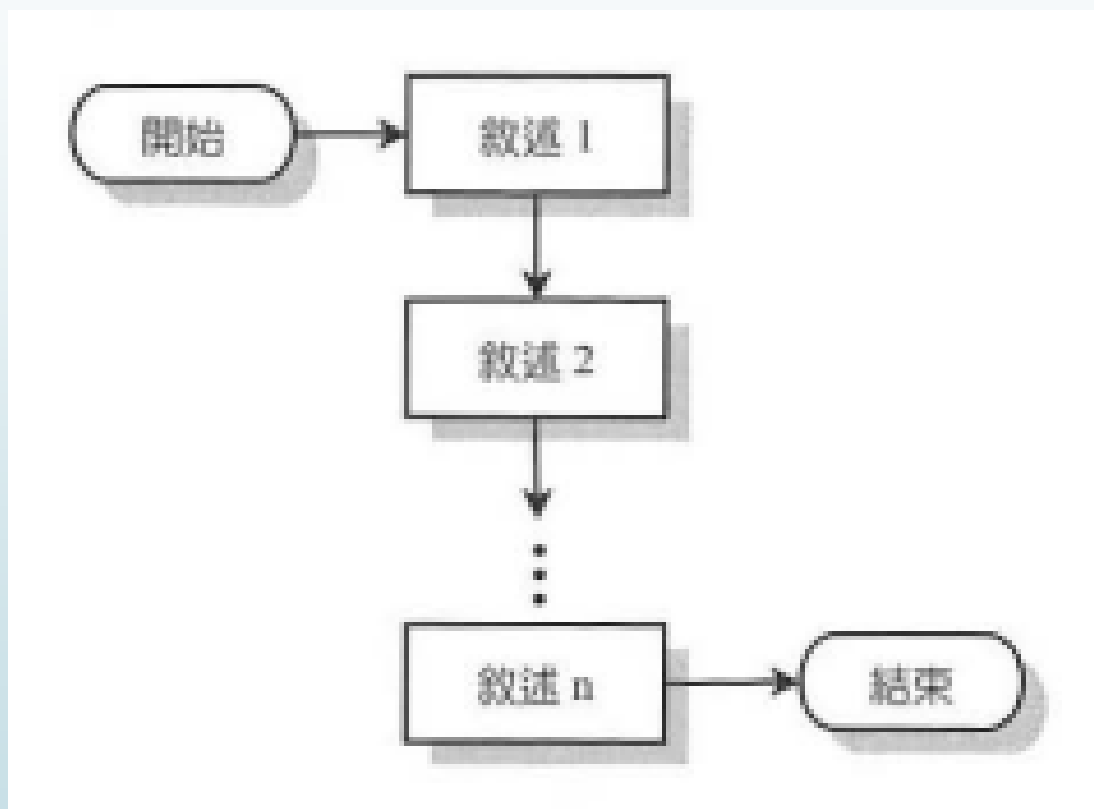
結構化程式設計

1

- 結構化的程式設計
 - 循序性的結構 (sequence structure)
 - 選擇性的結構 (selection structure)
 - 重複性的結構 (iteration structure)
- 有一個共通點，只有一個進入點，也只有一個出口，好維護，也可以減少除錯的時間

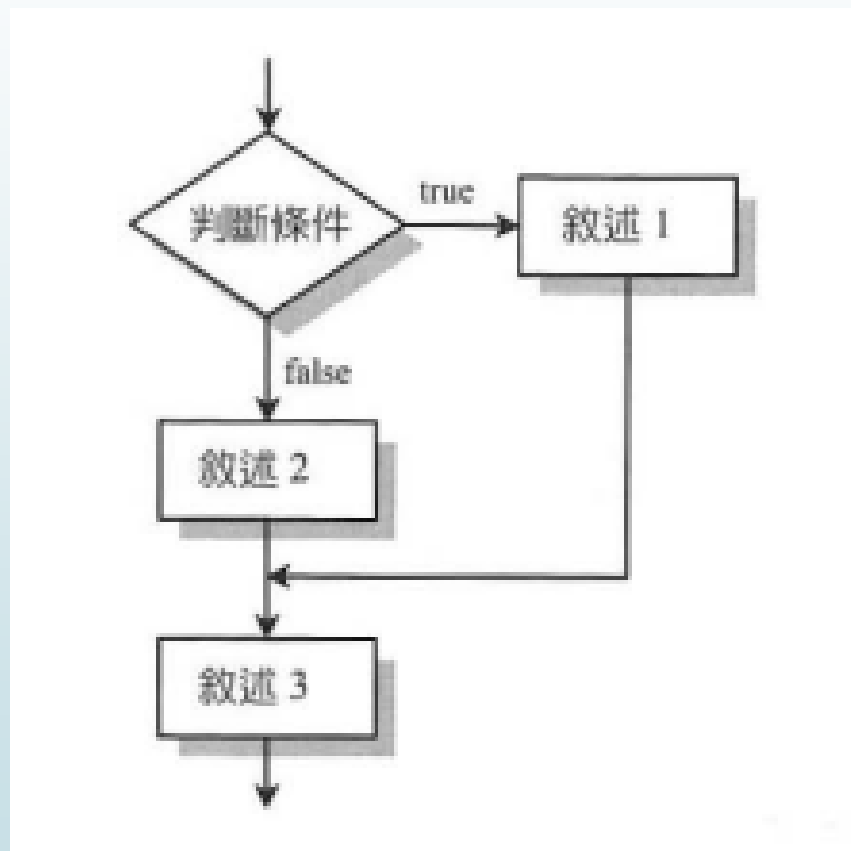
循序性結構

- 由上至下的敘述方式，一行執行完畢後，再接著執行下一行敘述



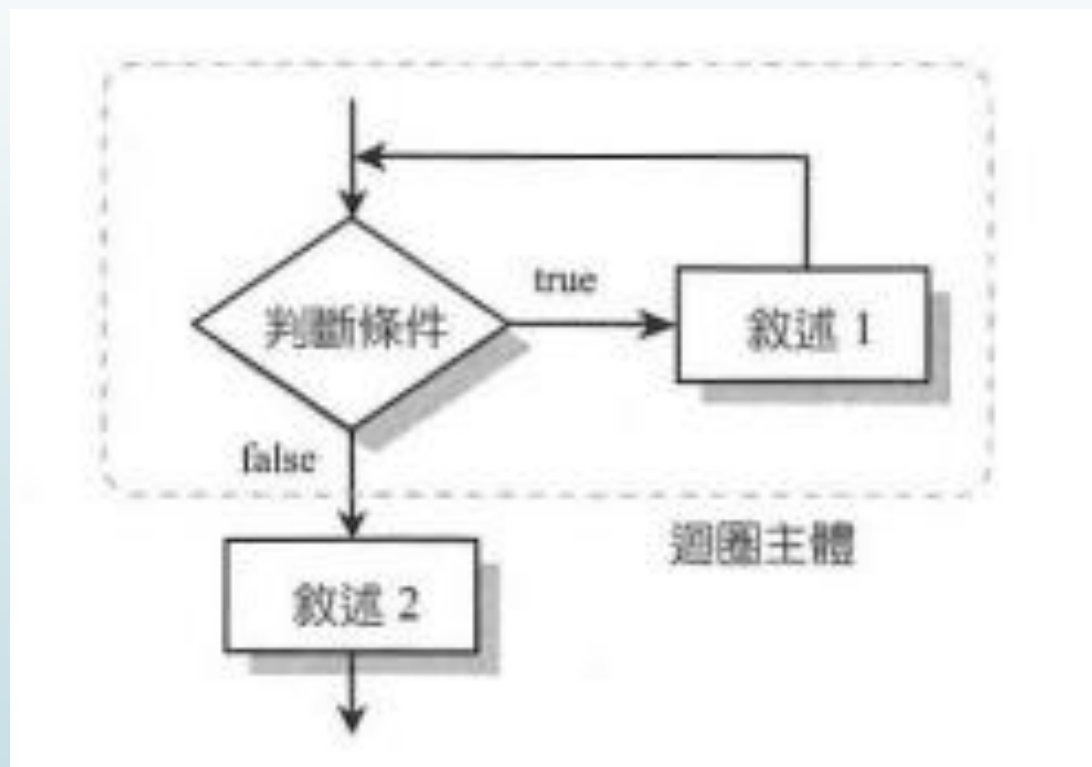
選擇性結構

- 根據判斷條件的成立與否，再決定要執行那些敘述結構



重複性結構

- 根據判斷條件的成立與否，決定程式段落的執行次數，這個程式段落就稱為迴圈主體



迴圈

5

- 迴圈有for、while和do while三種
- 迴圈適合運用在重複性的計算
 - EX
 - $\text{num} = 1+2+3+4+5+\dots+1000$

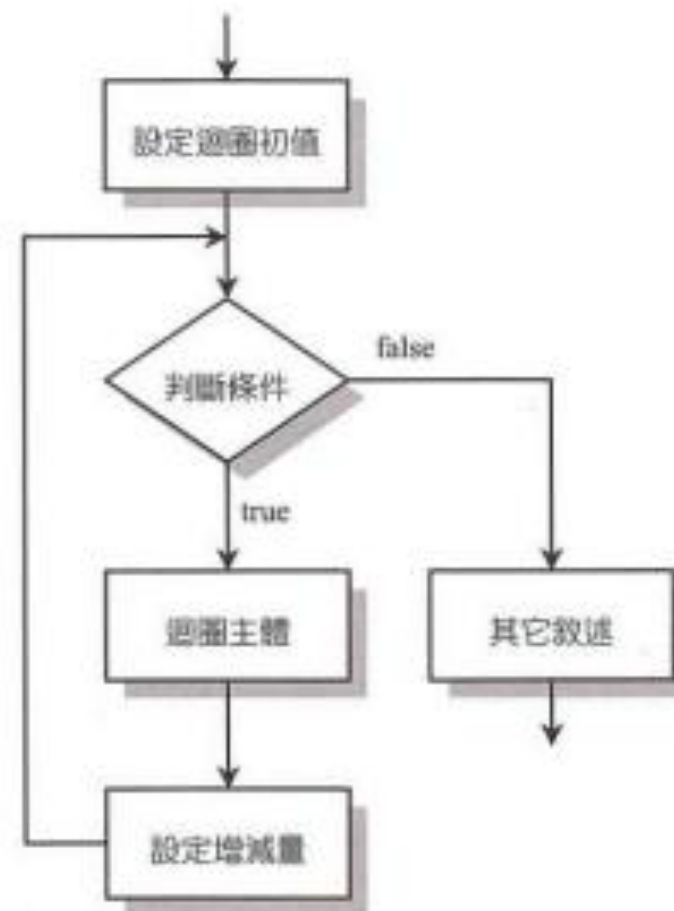
for迴圈

語法

```
for(設定迴圈初值 ; 判斷條件 ; 設定增減值)  
{  
    ...  
}
```

- 若迴圈主體只有一行時，可省略左右大括號

流程圖

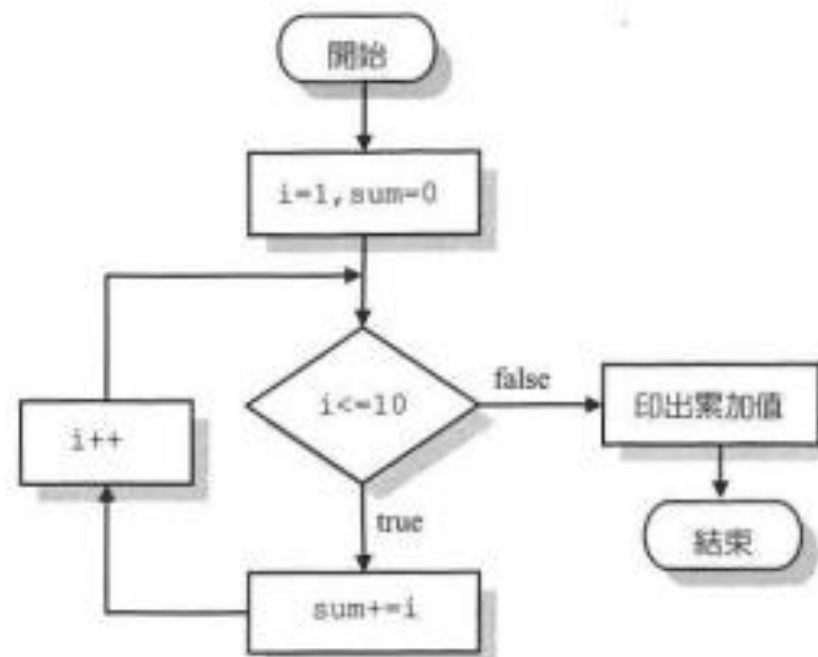


for 範例

```
#include <stdio.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;
}
```



for 範例變數變化狀態

表 7.2.1 for 迴圈內，i 與 sum 值變化的情形

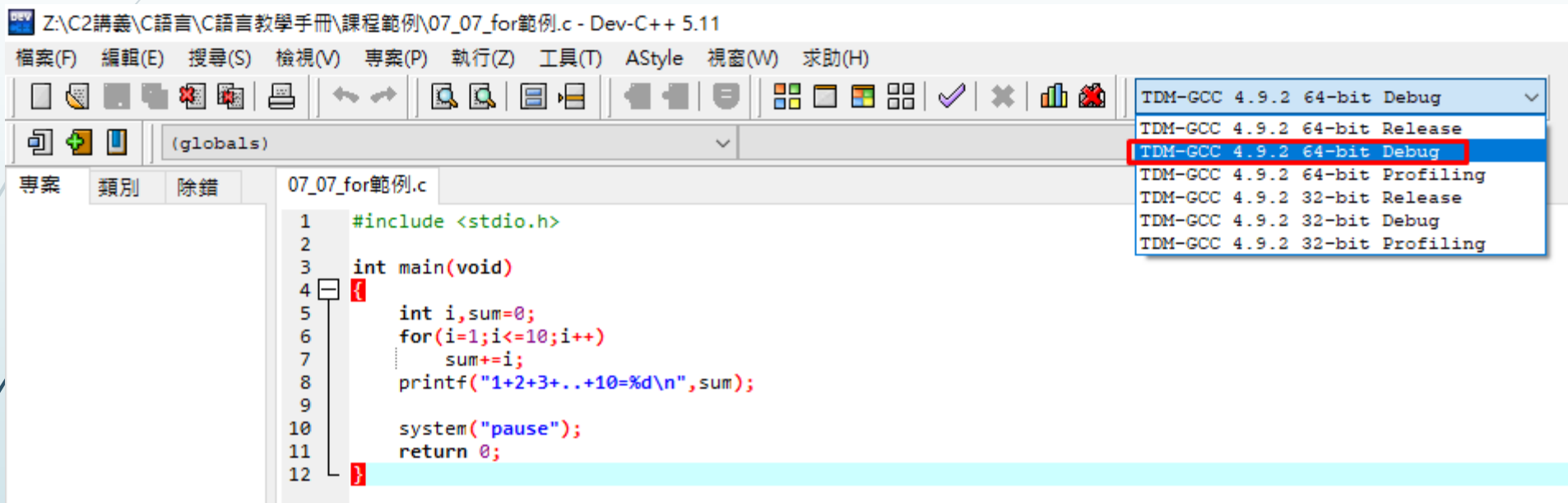
i 的值	sum 的值	計算 $\text{sum}+=i$ 之後，sum 的值
1	0	1
2	1	3
3	3	6
4	6	10
5	10	15
6	15	21
7	21	28
8	28	36
9	36	45
10	45	55

執行完 for 迴圈之後，
sum 的值

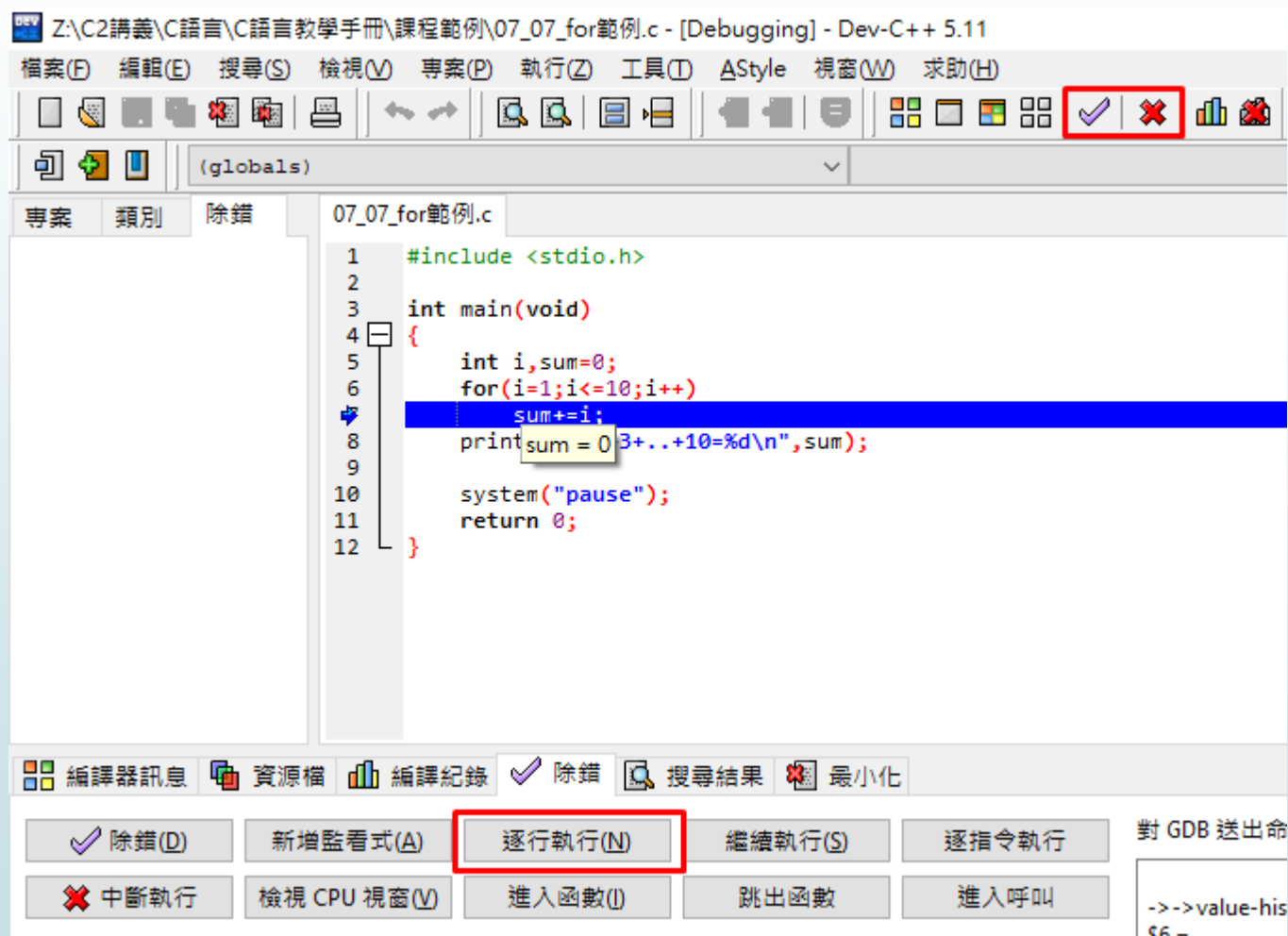
Dev C++設定除錯模式

- 設定編譯模式
 - 改為debug模式
- 選擇中斷點該行高亮紅底
- 點選除錯(前提必須先編譯過程式)
- 除錯中選項
 - 逐行執行
 - 需使用中斷執行才能結束除錯模式
 - 滑鼠移到變數上可觀察變數狀態
 - 可將多個變數加到監看式中更容易觀察

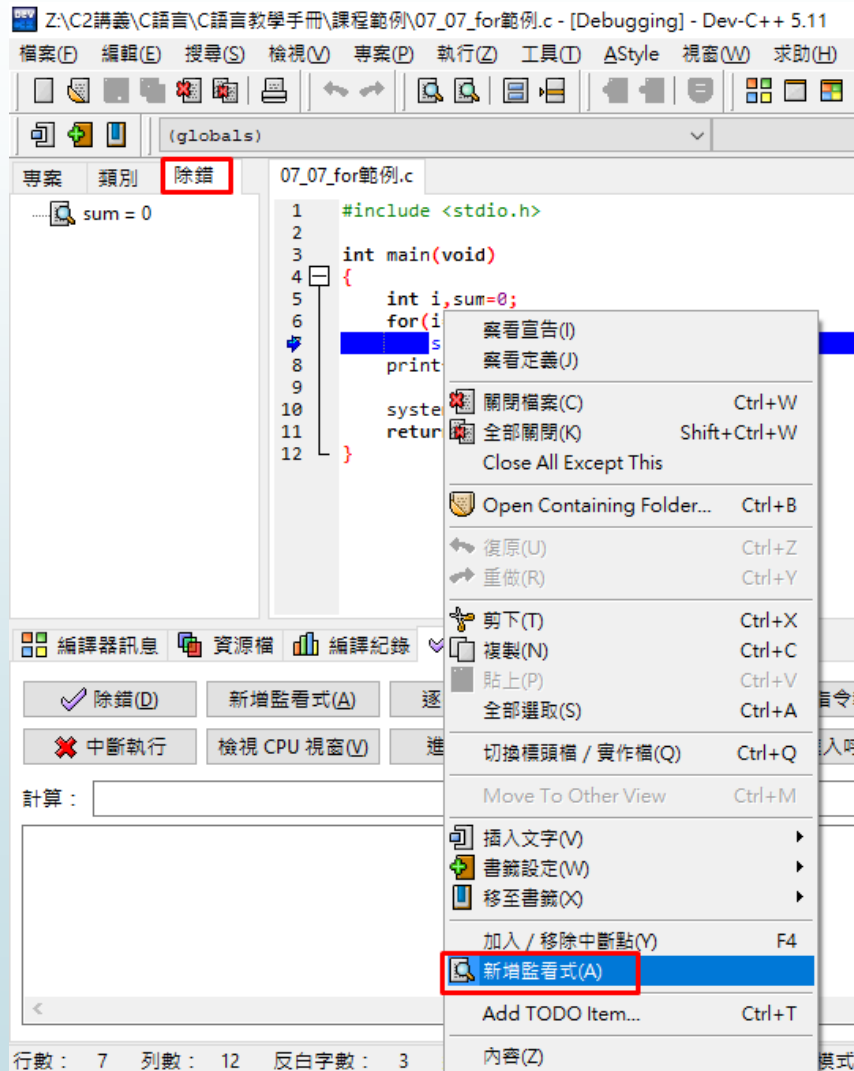
設定編譯模式



除錯模式



將變數加入監聽

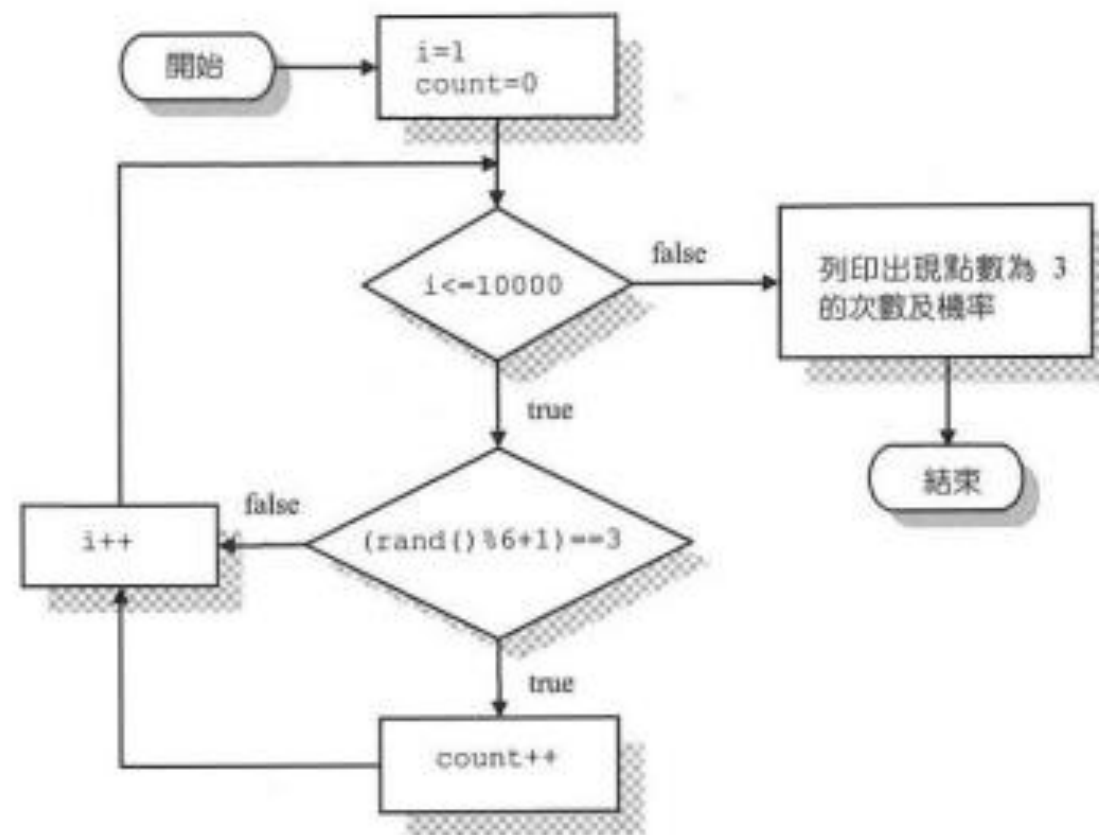


for範例-計算機率

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i, count=0;

    for(i=1; i<=10000; i++)
        if((rand()%6+1)==3)
            count++;

    printf("擲10000次骰子時，出現3點的次數為%d次\n", count);
    printf("機率為%.3f\n", (float)count/10000);
    system("pause");
    return 0;
}
```

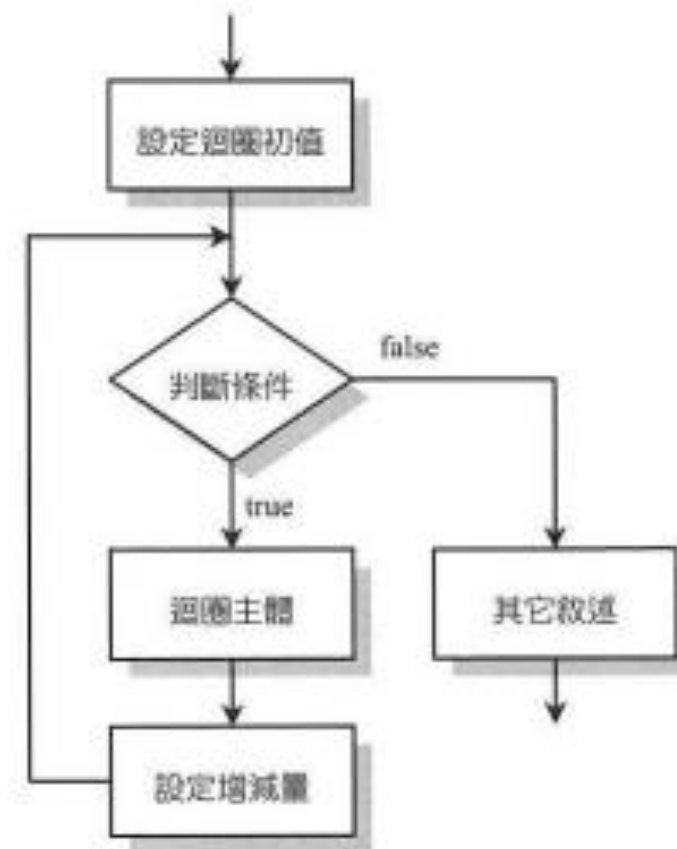


while迴圈

語法

```
while(判斷條件)
{
    迴圈主體;
    設定增減值;
}
```

流程圖



while範例

```
#include <stdio.h>

int main(void)
{
    int i=1,sum=0;
    while(sum<=100)
    {
        sum+=i;
        printf("從1累加到%2d=%2d\n",i,sum);
        i++;
    }
    printf("必須累加到%d\n",i-1);
    system("pause");
    return 0;
}
```


for迴圈與while迴圈比較

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

無窮迴圈

- ➡ 無窮迴圈(endless loop)在迴圈執行的過程中，找不到可以離開迴圈的出口，

```
#include <stdio.h>

int main(void)
{
    int i=1;
    while(i>0)
    {
        printf("i=%d\n",i++);
    }
    system("pause");
    return 0;
}
```

無窮迴圈範例

```
#include <stdio.h>

int main(void)
{
    char ch;
    while(ch!=17)
    {
        ch = getch();
        printf("ASCII of ch = %d\n",ch);
    }
    printf("您已經按了Ctrl+q...\n");
    system("pause");
    return 0;
}
```

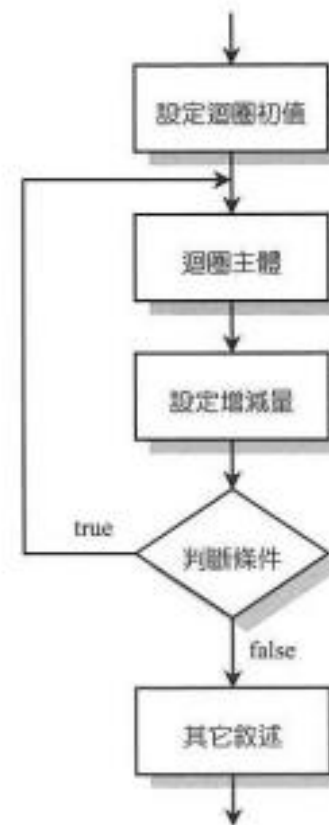
do while

語法

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

- ➡ 迴圈主體會先執行一次再根據判斷條件是否執行
- ➡ 注意結尾要加「;」當作結尾

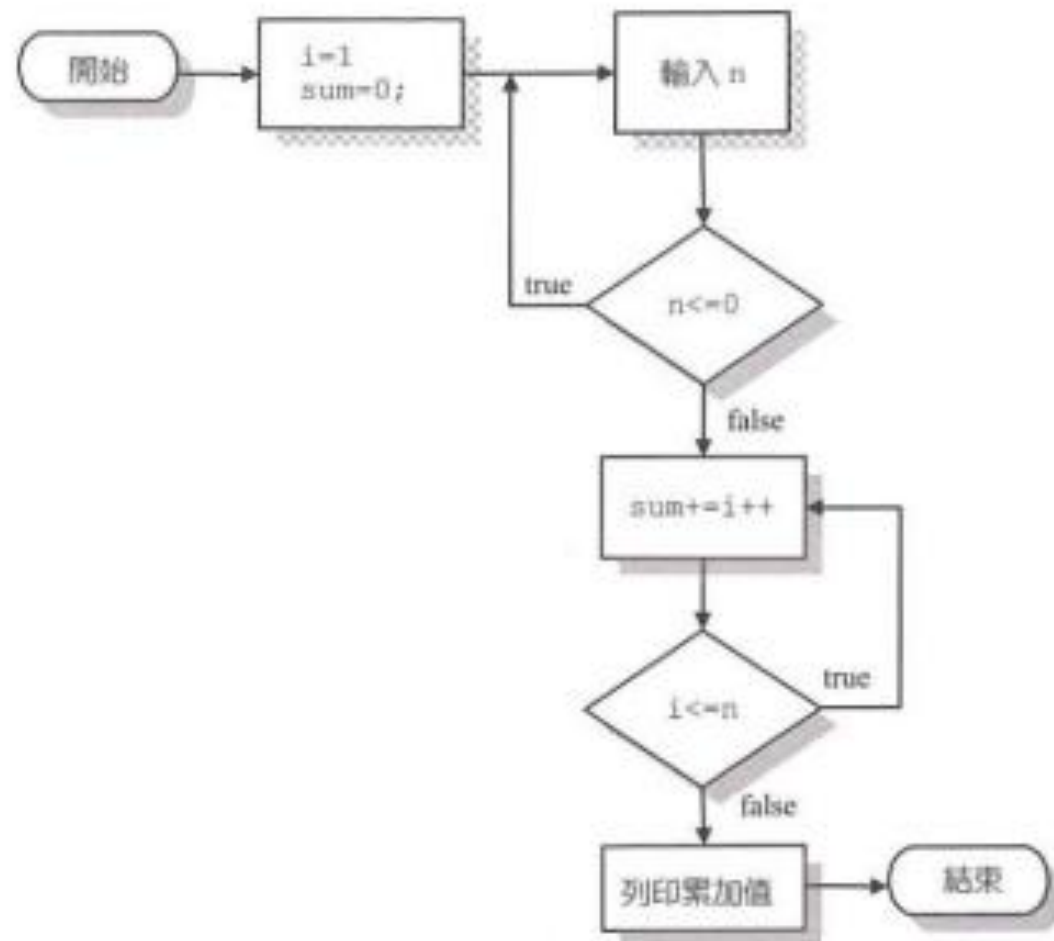
流程圖



do while範例

```
#include <stdio.h>

int main(void)
{
    int n,i=1,sum=0;
    do
    {
        printf("請輸入n值 (n>0):");
        scanf("%d",&n);
    }while(n<=0);
    do
    {
        sum+=i++;
    }while(i<=n);
    printf("1+2+...+%d=%d\n",n,sum);
    system("pause");
    return 0;
}
```



do while範例-階乘

$$n! = \begin{cases} 1 \times 2 \times 3 \times \cdots \times n; & n \geq 1 \\ 1; & n = 0 \end{cases}$$

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

```
int main(void)
```

```
{
```

```
    int n,i=1,fact=1;
```

```
    do
```

```
    {
```

```
        printf("請輸入n值 (n>0):");
```

```
        scanf("%d",&n);
```

```
    }while(n<=0);
```

```
    do
```

```
        fact*=i++;
```

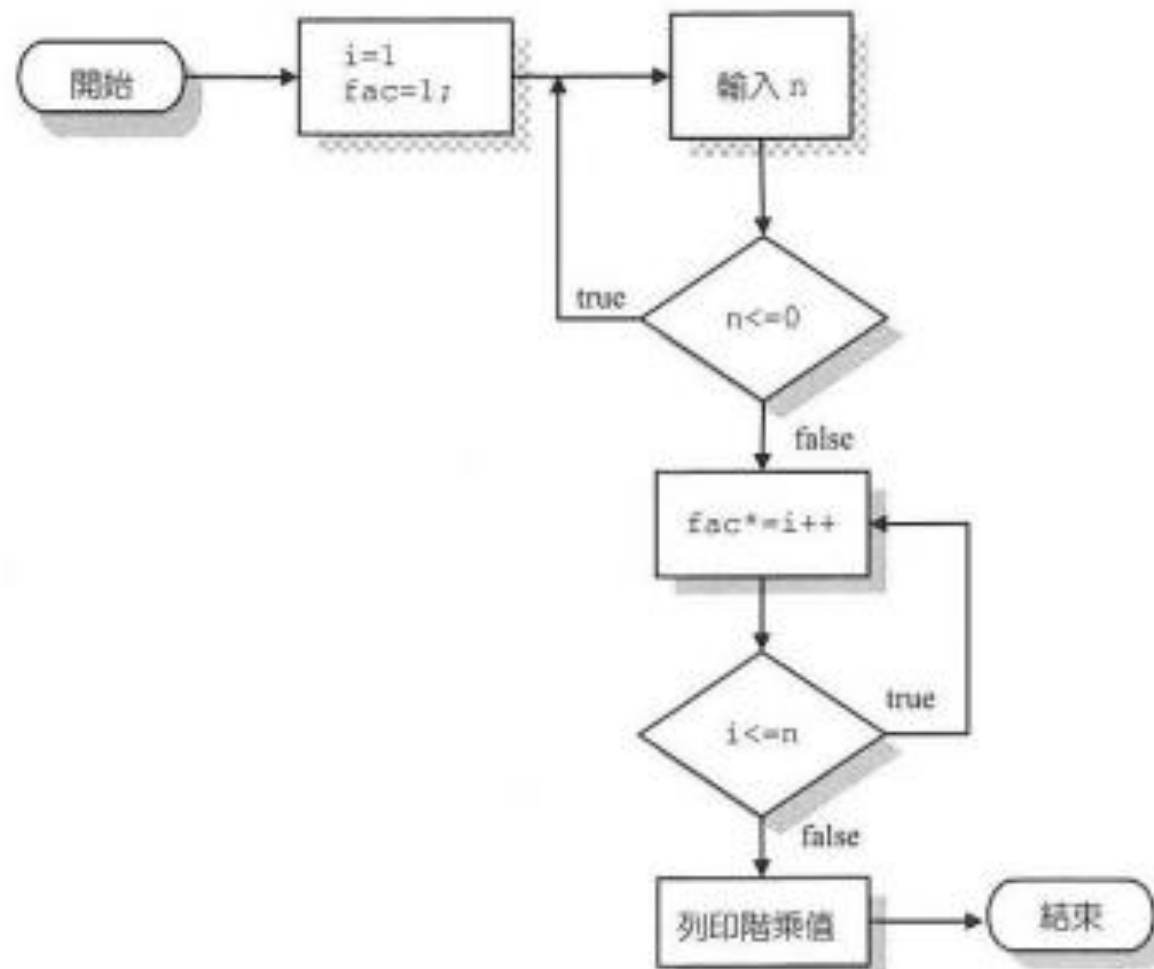
```
    while(i<=n);
```

```
    printf("%d!=%d\n",n,fact);
```

```
    system("pause");
```

```
    return 0;
```

```
}
```



空迴圈

- 迴圈主體沒有任何敘述，但還是會消耗CPU處理時間
- 觀察某部分的執行結果，而故意將執行速度加以延遲

```
for (設定初值; 判斷條件; 設定增減量)  
{  
}
```

或是

```
for (設定初值; 判斷條件; 設定增減量) ; → 要加分號
```

空迴圈的誤用

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

```
int main(void)
{
    int i;
    for(i=1;i<=10000;i++);
    ...    printf("i=%d\n",i);

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


各迴圈比較

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

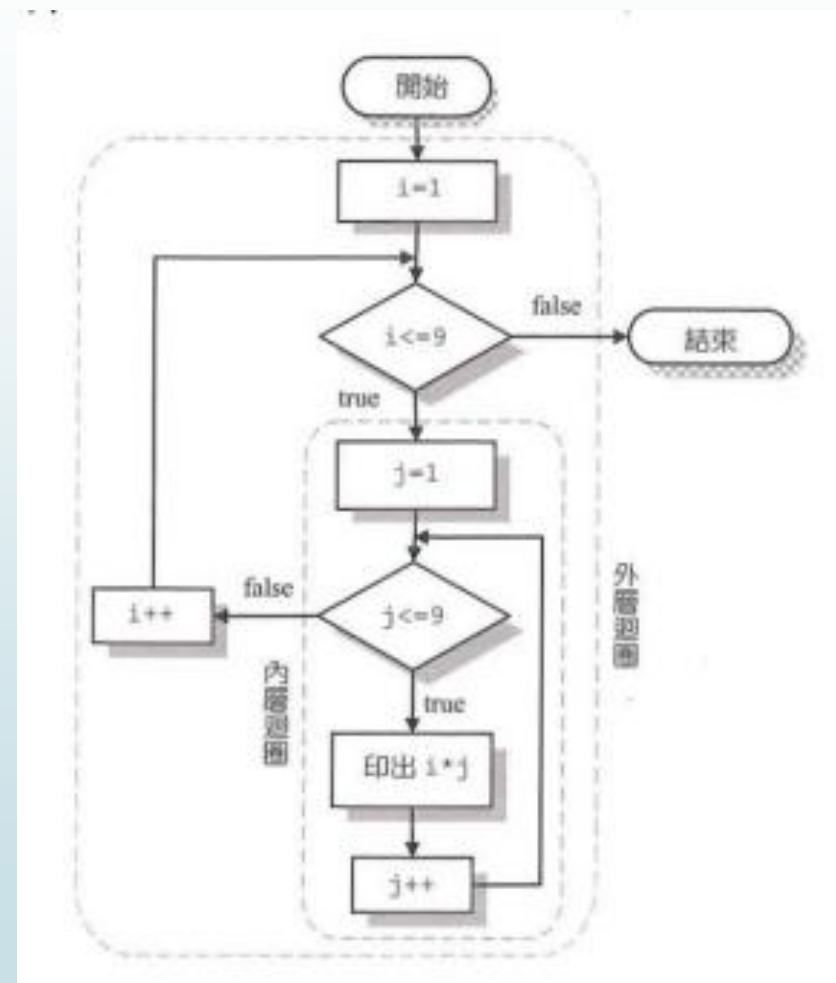
for 巢狀迴圈練習

Z:\C2講義\C語言\C語言教學手冊\課程範例\07_25_巢狀迴圈-九九乘法表.exe

```

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

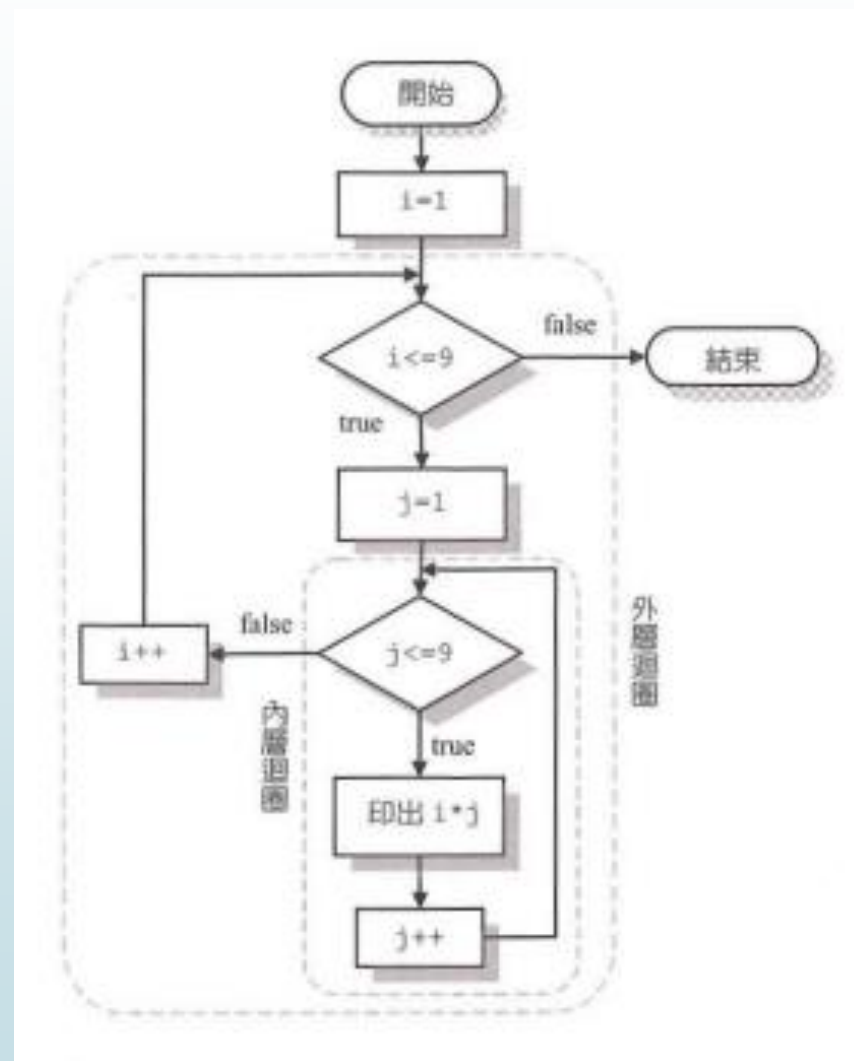
請按任意鍵繼續 . . .



while 巢狀迴圈改寫

```
#include <stdio.h>

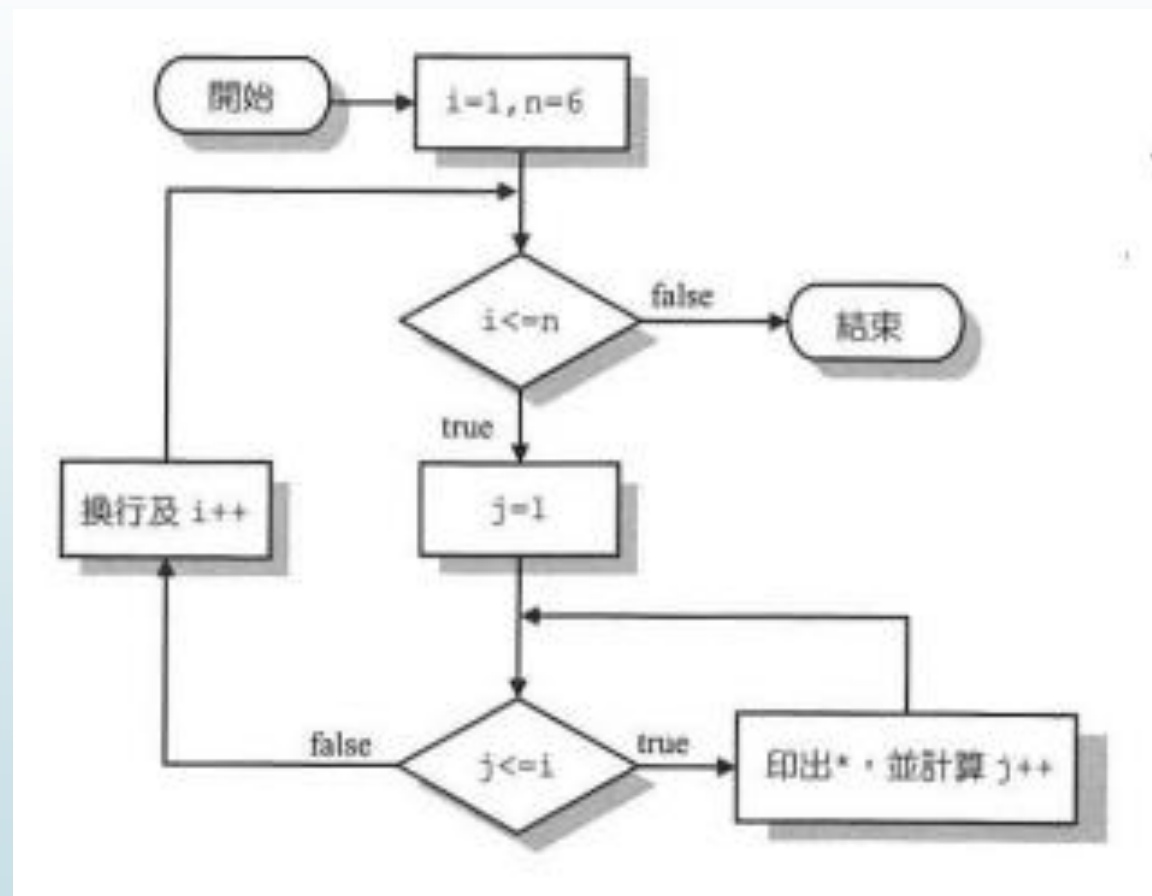
int main(void)
{
    int i=1;
    while(i<=9)
    {
        int j=1;
        while(j<=9)
        {
            printf("%d*%d=%2d ",i,j,i*j);
            j++;
        }
        printf("\n");
        i++;
    }
    system("pause");
    return 0;
}
```



迴圈範例-三角形

```
#include <stdio.h>

int main(void)
{
    int i,j,n=6;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("*");
        }
        printf("\n");
    }
    system("pause");
    return 0;
}
```



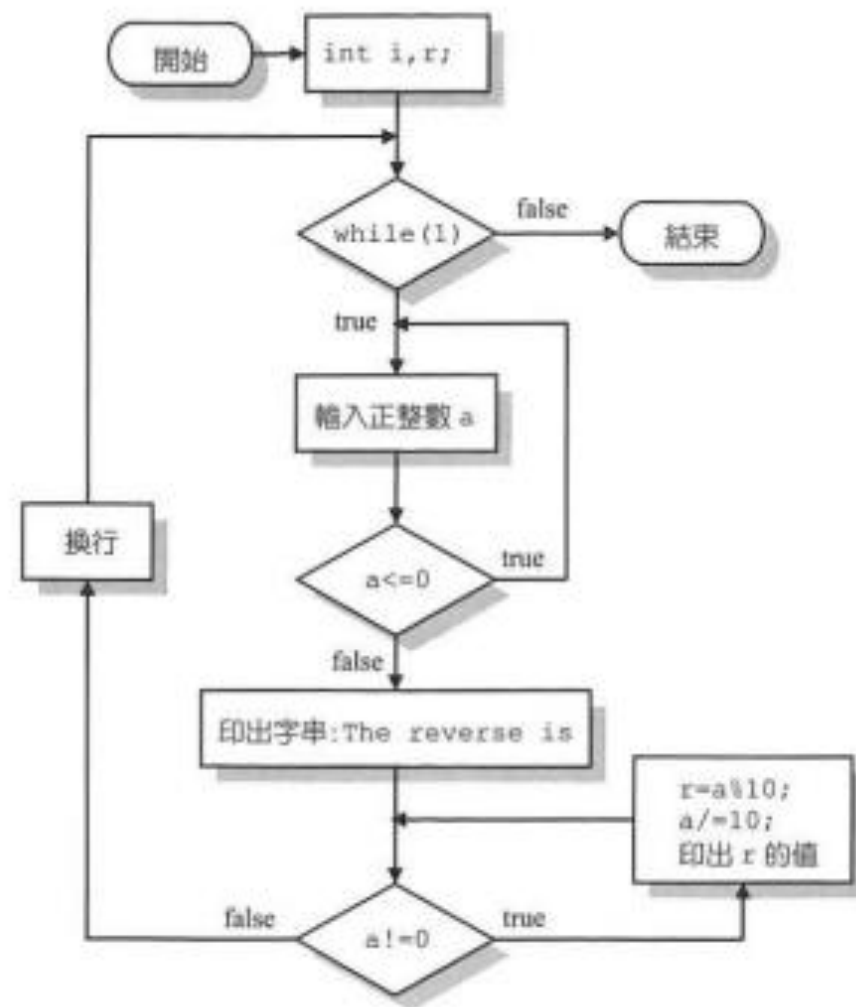
迴圈練習-將整數反過來列印

- 輸入一個正整數將其倒印出來
- 輸入整數為負號時重新輸入

```
input an integer:-50
input an integer:798654321
The reverse is 123456897

input an integer:123456123
The reverse is 321654321

input an integer:
```



迴圈-break

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

break 範例

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

```
int main(void)
```

```
{
```

```
    int i;
```

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

```
    {
```

```
        if(i%3==0)
```

```
            break;
```

```
        printf("i=%d\n",i);
```

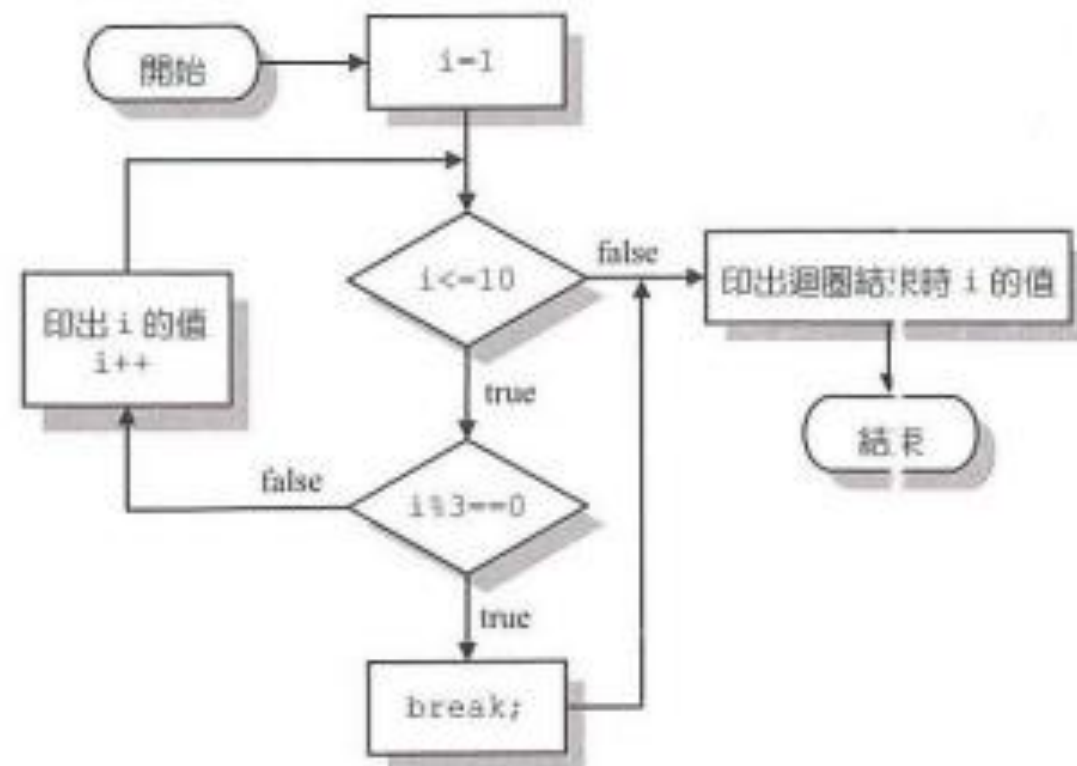
```
    }
```

```
    printf("跳離迴圈時, i=%d\n",i);
```

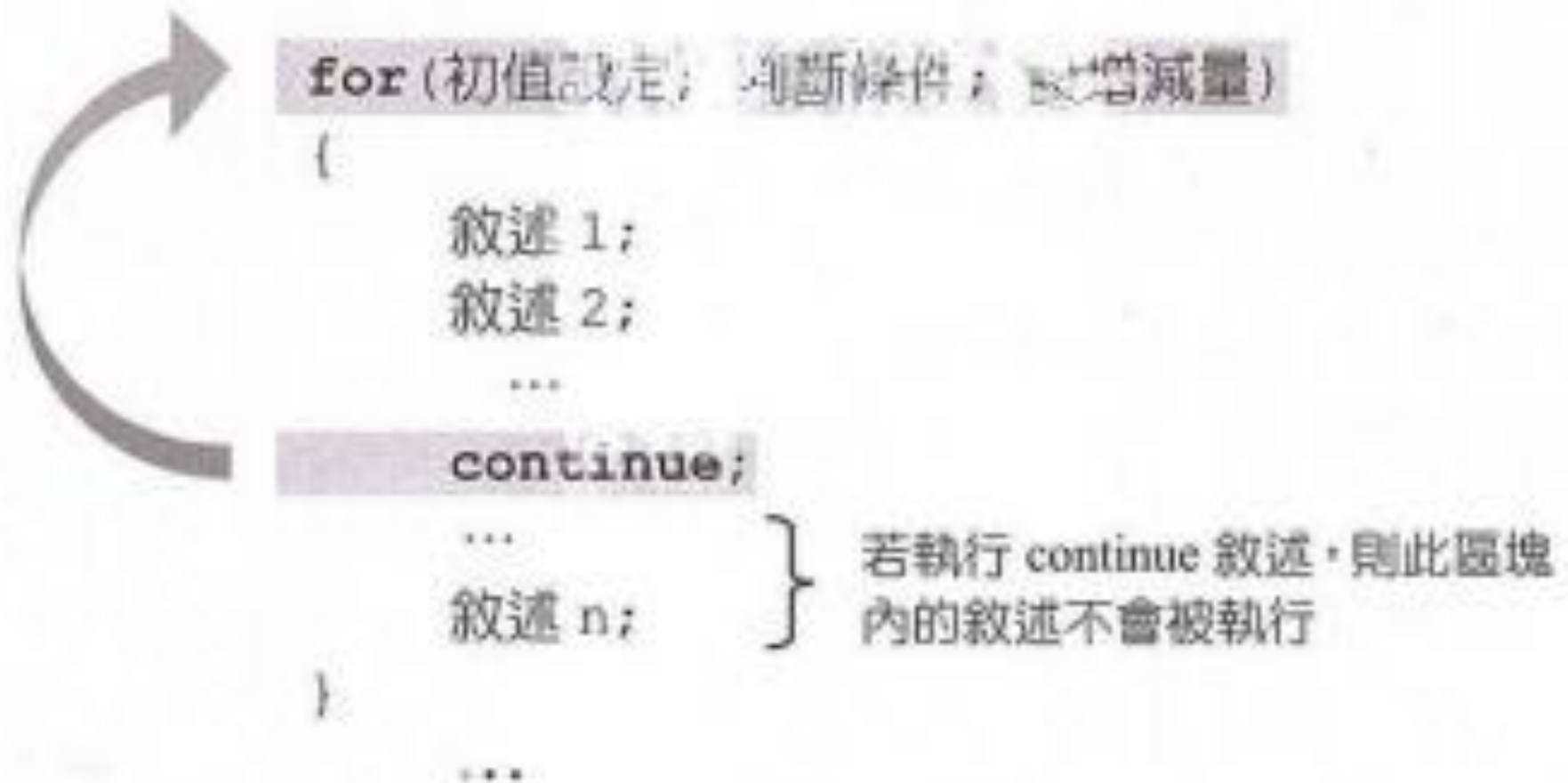
```
    system("pause");
```

```
    return 0;
```

```
}
```



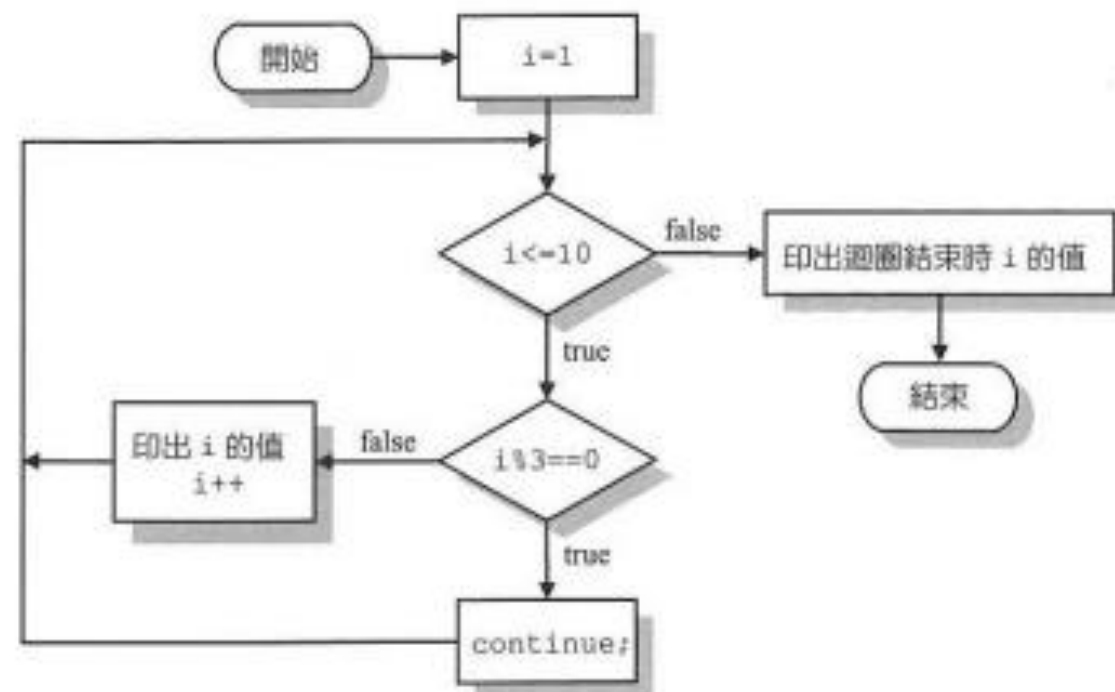
迴圈-continue



continue 範例

```
#include <stdio.h>

int main(void)
{
    int i;
    for(i=1; i<=10; i++)
    {
        if(i%3==0)
            continue;
        printf("i=%d\n", i);
    }
    printf("跳離迴圈時, i=%d\n", i);
    system("pause");
    return 0;
}
```



習題-九九乘法表

► 請用兩層for迴圈完成

```
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
請按任意鍵繼續 . . .
```

習題-列印三角形

- 試利用巢狀迴圈完成以下如圖結果的程式

```
1
12
123
1234
12345
請按任意鍵繼續 . . .
```