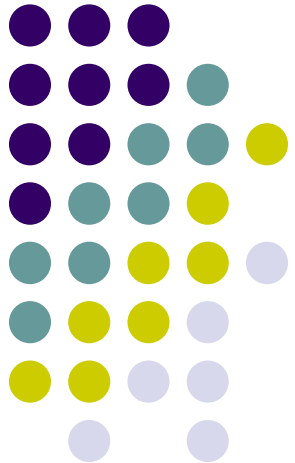
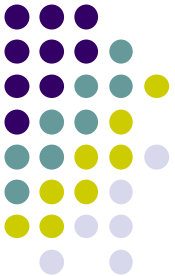
C語言補充
格式化輸出

學習 cout / printf() 的輸出方法

認識列印格式碼

學習字元的輸出函數





cout輸出指令 (1/2)

- 下面的範例說明什麼是程式區塊

```
01  // prog2_2, 程式的區塊
02  #include <iostream>              // 含括 iostream 檔案
03  #include <cstdlib>               // 含括 cstdlib 檔案
04  using namespace std;
05  int main(void)                   // main()區塊開始
06  {
07     int num=6;                    // 宣告整數 num
08     cout << "I have " << num << " apples." << endl;
09
10     system("pause");
11     return 0;
12  }                                // main()區塊結束
```

main()的區塊

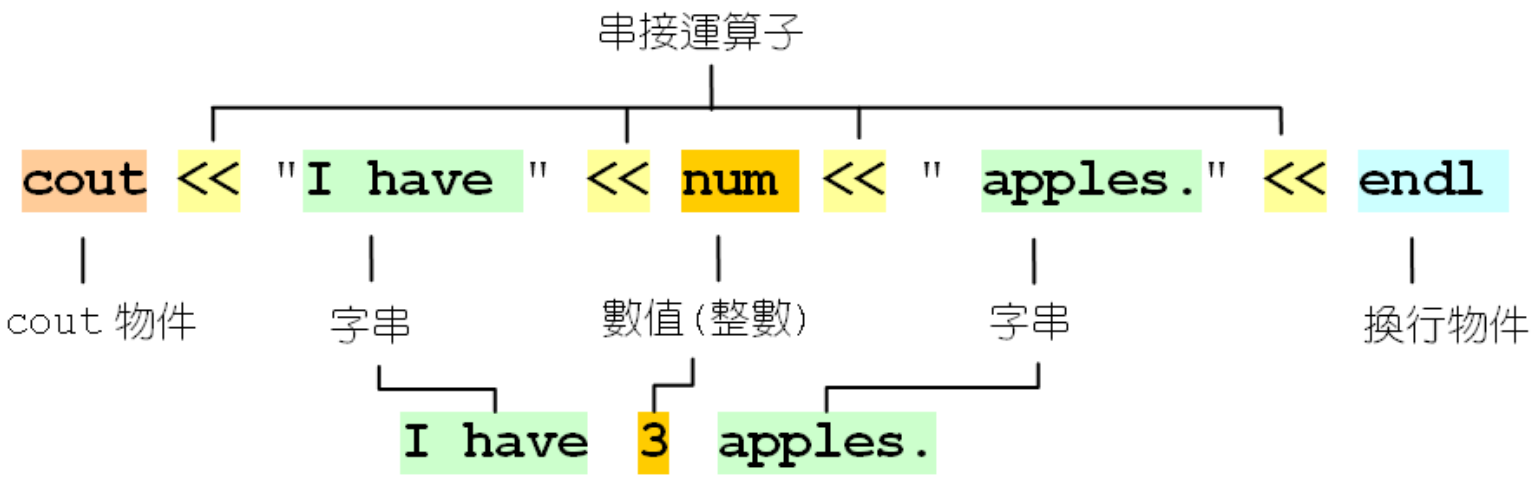
```
/* prog2_2 OUTPUT---

I have 6 apples.
--------------------*/
```

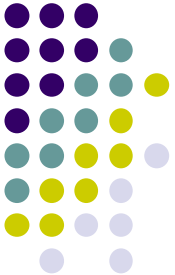
cout與串接運算子<< (2/2)

- C++是採cout與「串接運算子<<」來輸出

```
cout << "I have " << num << " apples." << endl;
cout << "You have " << num << " apples, too." << endl;
```

- 以cout顯示字串：





輸出函數 printf() (1/3)

- printf 是由 print（列印）與 format（格式）二字組成
- printf() 函數的使用格式 ：

printf() 函數的格式

```
printf("格式字串", 項目1, 項目2, ...);
```

輸出函數 printf() (2/3)

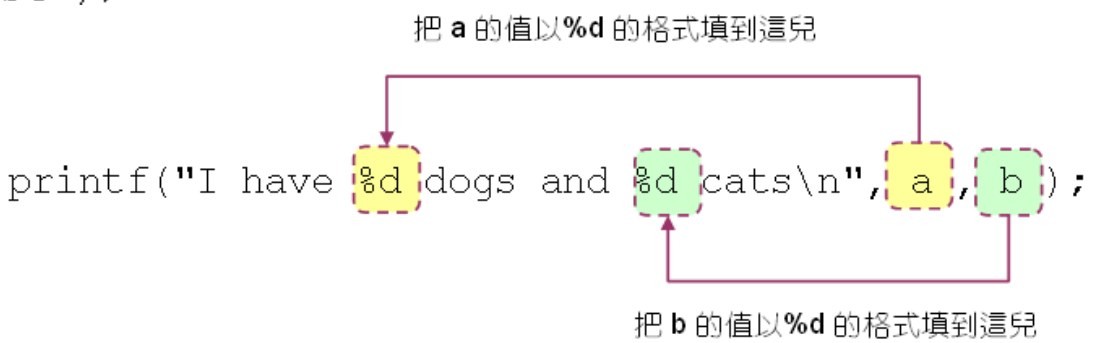
- 下面的程式為使用 printf() 函數的範例：

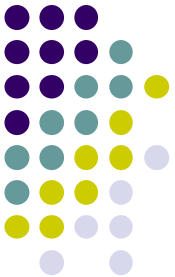
```
01  /* prog4_1, printf()函數的使用 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int a=2;
07     int b=4;
08     printf("I have %d dogs and %d cats\n",a,b);  /* 呼叫 printf()函數 */
09
10     system("pause");
11     return 0;
12  }
```

專用在C語言的標頭檔

不可用 cin.cout

```
/* prog4_1 OUTPUT------
I have 2 dogs and 4 cats
-------------------------*/
```





輸出函數 printf() (3/3)

- 下面的範例示範了如何印出字串：

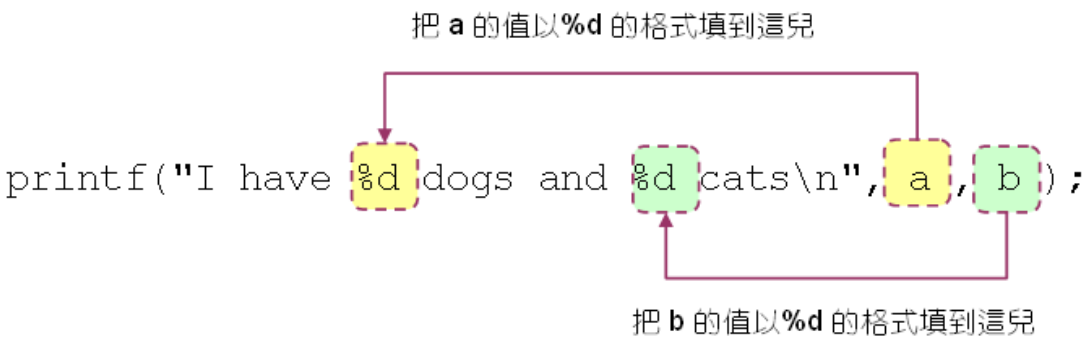
```
01  /* prog4_2, 印出字串 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     printf("Have a nice day!!\n");        /* 印出字串內容 */
07
08     system("pause");
09     return 0;
10  }

    /* prog4_2 OUTPUT--

    Have a nice day!!
    --------------------*/
```

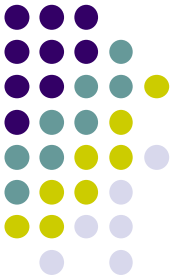
用於 printf() 的格式碼

- 下表列出了 printf() 函數常用的格式碼 (%)：

表 4.1.1 printf() 函數常用的格式碼

格式碼	說　明	格式碼	說　明
★ %c	字元	★ %%	印出百分比號特別注意
★ %d	十進位整數	%o	無號八進位整數
%ld	長整數	★ %s	字串
%e	浮點數，指數 e 型式	%u	無號十進位整數
★ %f	浮點數，小數點型式	%x	無號十六進位整數





跳脫序列

- 下表列出常用的跳脫序列 (\)：

表 4.1.2 使用於 printf() 函數的跳脫序列

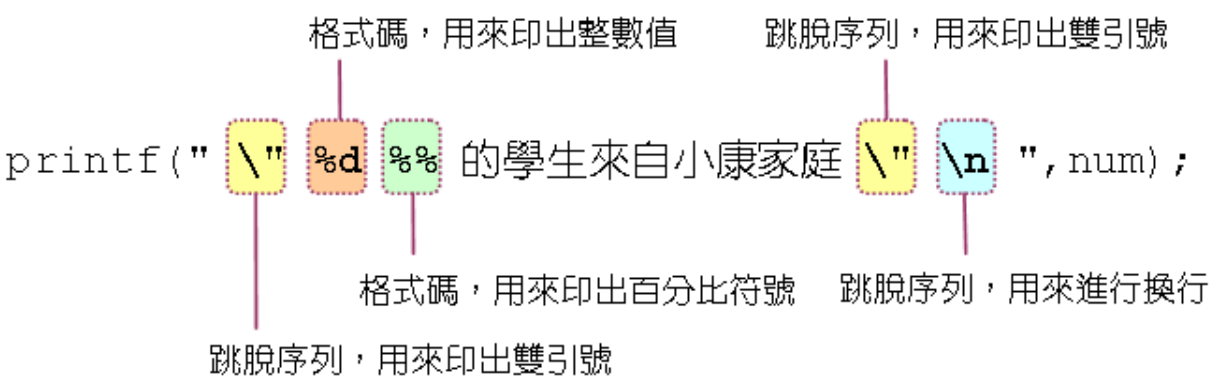
跳脫序列	功能	跳脫序列	功能
\a	警告音	☆ \"	印出雙引號
\b	倒退	☆ \\	印出反斜線
☆ \n	換行	\/	印出斜線 直接印 "/" 也可以
\r	歸位	☆ %%	印出百分比號
☆ \t	跳格		
☆ \'	印出單引號		

跳脫序列與格式碼的應用

- 下面的程式碼是利用格式碼印出字串：

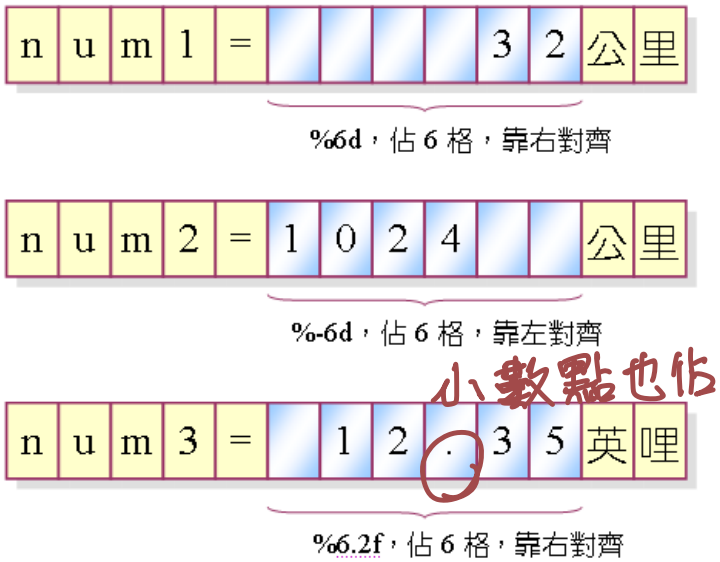
```
01  /* prog4_3, 使用 printf()函數 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num=25;
07     printf("\"%d%%的學生來自小康家庭\"\n",num);    /* 印出字串 */
08
09     system("pause");
10     return 0;
11  }
```

```
/* prog4_3 OUTPUT---
"25%的學生來自小康家庭"
----------------------*/
```



控制輸出欄位的寬度

- 設定欄位的寬度：




```
01  /* prog4_4, 印出特定格式 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num1=32, num2=1024;
07     float num3=12.3478f;
08
09     printf("num1=%6d 公里\n",num1);   /* 以「%6d」格式印出 num1 */
10     printf("num2=%-6d 公里\n",num2);  /* 以「%-6d」格式印出 num2 */
11     printf("num3=%6.2f 英哩\n",num3); /* 以「%6.2f」格式印出 num3 */
12
13     system("pause");
14     return 0;
15  }
```

```
/* prog4_4 OUTPUT---

num1=    32 公里
num2=1024   公里
num3= 12.35 英哩
--------------------*/
```

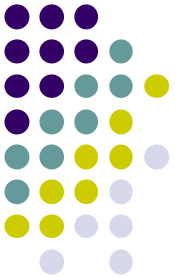
printf() 函數的修飾子(1/2)

表 4.1.3 printf() 函數的修飾子

修飾子	功能	舉例
-	靠左對齊	%-3d
+	將數值的正負號顯示出來	%+5d
空白	數值為正值時，留一格空白；為負值時，顯示負號	% 6f
0	將固定欄位長度的數值前空白處填上 0（與負號「-」同時使用時，此功能無效）	%07.2f

資料內容	格式	執行結果
12345	%10d	12345
12345	%+d	+12345
12345	%09d	000012345
12345	%-10d	12345

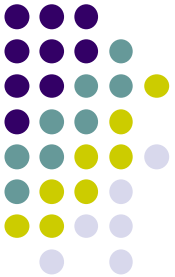
資料內容	格式	執行結果
12345	% d	12345
123.456	%7.2f	123.46
123.456	%010.3f	000123.456
123.456	%+10.4f	+123.4560



printf() 函數的修飾子(2/2)

- printf() 函數修飾子的使用範例：

```
01  /* prog4_5, 使用 printf()函數的修飾子*/
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int i=1234;
07     printf("i=%+08d\n",i);      /* 呼叫 printf()函數 */
08
09     system("pause");
10     return 0;
11  }

/* prog4_5 OUTPUT--

i=+0001234
----------------------*/
```



以不同進位的型式輸出

- 下面的程式將整數以八進位與十六進位輸出：

```
01  /* prog4_6, 將 10 進位整數以不同的進位系統做輸出 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     printf("42 的八進位是 %o\n",42);          /* 印出 42 的八進位 */
07     printf("42 的十六進位是 %x\n",42);        /* 印出 42 的十六進位 */
08
09     system("pause");
10     return 0;
11  }

/* prog4_6 OUTPUT--

42 的八進位是 52
42 的十六進位是 2a
----------------------*/
```

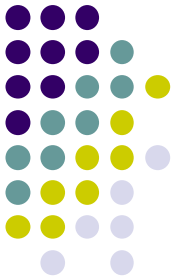
控制碼必須符合輸出的型態 (1/2)

- 錯誤的範例： 整數資料以其它型態輸出

```
01  /* prog4_7, 整數資料以其它型態輸出, 錯誤的範例 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int a=15;                    /* 宣告整數變數 a，並設值為 15 */
07
08     printf("a=%d\n",a);                          /* 印出 a 的值 */
09     printf("以浮點數型態印出: %f\n",a);          /* 以%f 格式碼印出 a 的值 */
10     printf("以指數型態印出  : %e\n",a);          /* 以%e 格式碼印出 a 的值 */
11
12     system("pause");
13     return 0;
14  }
```

```
/* prog4_7 OUTPUT-----------

a=15
以浮點數型態印出: 0.000000
以指數型態印出  : 1.910519e-297
-------------------------------*/
```

控制碼必須符合輸出的型態 (2/2)



- 修正 prog4_7 的錯誤 ：

```
01  /* prog4_8, 修正 prog4_7 的錯誤 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int a=15;                    /* 宣告整數變數 a，並設值為 15 */
07
08     printf("a=%d\n",a);                                  /* 印出 a 的值 */
09     printf("以浮點數型態印出: %f\n",(float)a);    /* 以浮點數型態印出 a */
10     printf("以指數型態印出: %e\n",(double)a);     /* 以指數型態印出 a */
11
12     system("pause");
13     return 0;
14  }
```

```
/* prog4_8 OUTPUT----------

a=15
以浮點數型態印出: 15.000000
以指數型態印出: 1.500000e+001
----------------------------*/
```

cout 格式化控制

- #include <iomanip>

- int num = 10; float pi = 3.14159;

1. setw()函數：控制後面緊接著的資料之輸出寬度

- %5d：cout << setw(5) << num << endl;

2. Setfill()函數：填充字元控制 ###10

- 無C語言寫法 cout << setw(5) << setfill('#') << num << endl;

3. dec/oct/hex：10進位/8進位/16進位之控制關鍵字

- %5X cout << setw(5) << hex << num << endl;

4. setprecision()函數：設定後面浮點數的精確度

- %.3f cout << setprecision(3) << pi << endl;

%6.2f
↳ setw(6) << setprecision(2)

%.3f
↳ setprecision(3)

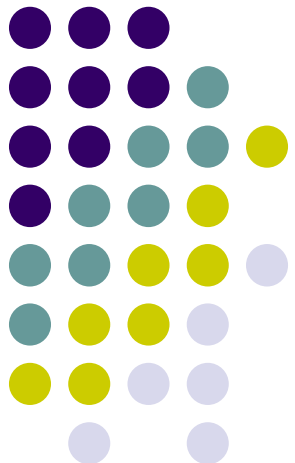
3.142

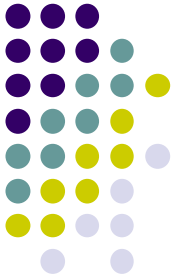
第五章
選擇性敘述與迴圈

認識程式的結構設計

學習選擇性敘述與各種迴圈的用法

學習多重選擇敘述的使用



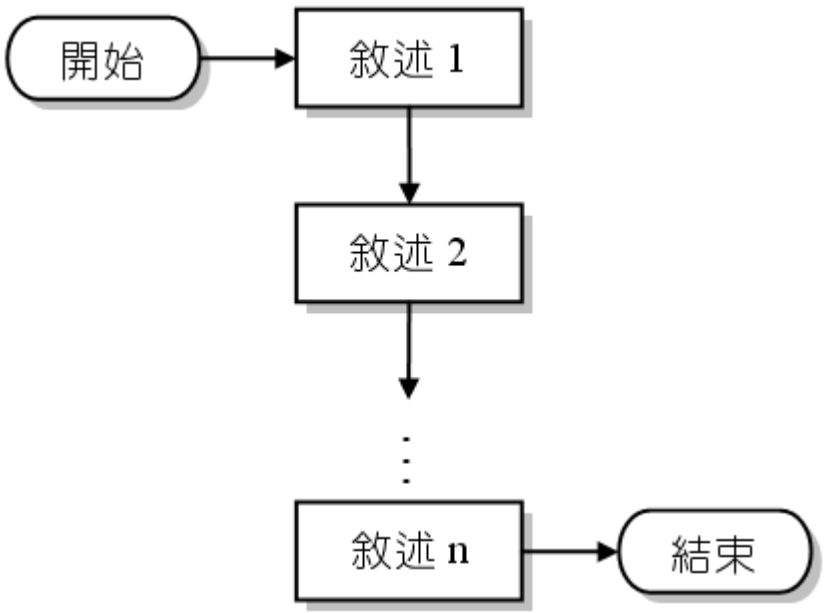


程式的結構

- 程式的結構包含有下面三種：
  - 循序性結構（sequence structure）
  - 選擇性結構（selection structure）
  - 重複性結構（iteration structure）

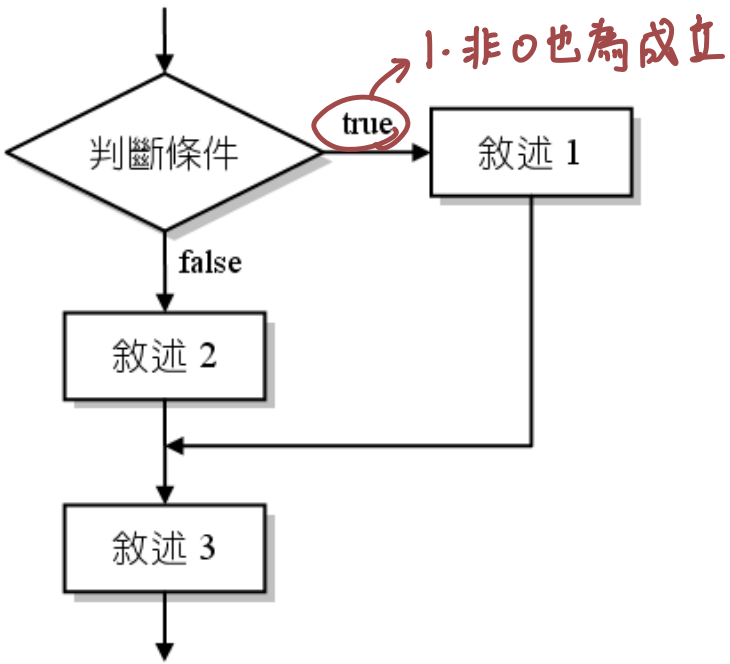
循序性結構

- 循序性結構是採上至下（top to down）的敘述方式



選擇性結構

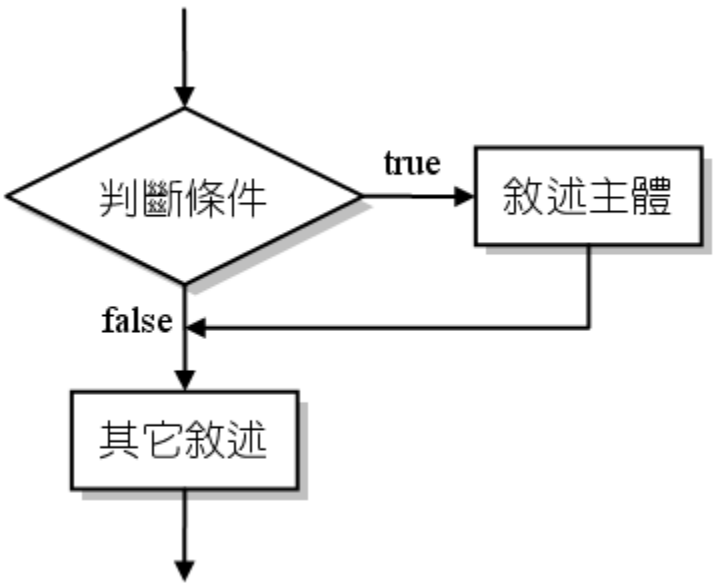
- 選擇性結構根據條件的成立與否，再決定要執行哪些敘述



選擇性結構與if敘述

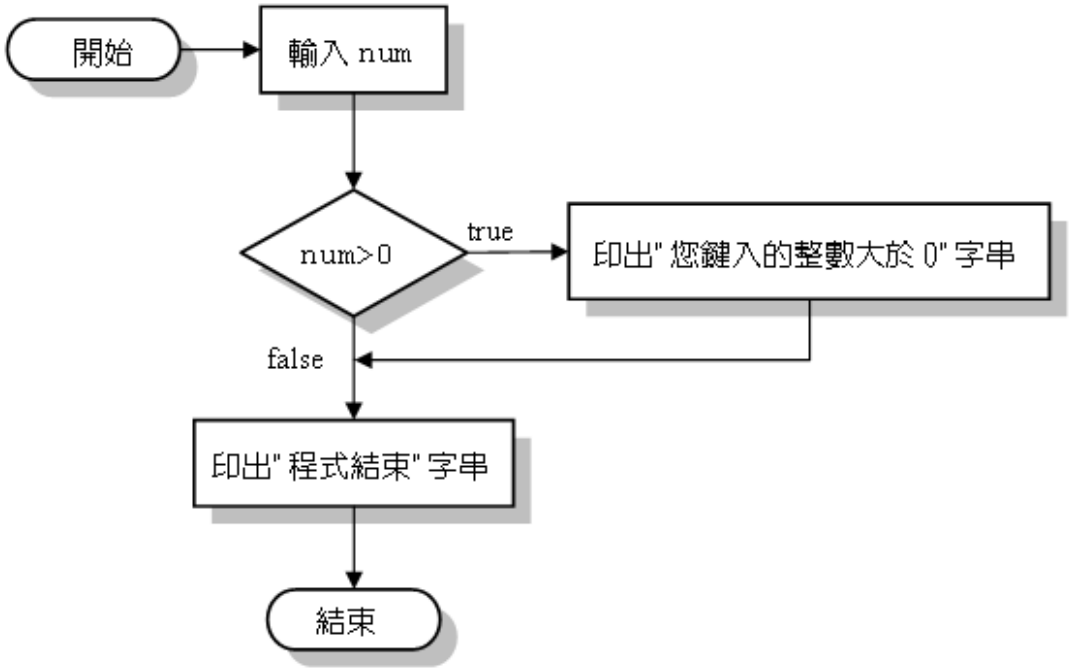
- 選擇性結構包括if、if-else及switch敘述
- if敘述的格式如下

```
if (判斷條件)
{
    敘述主體;
}
```



if 敘述的使用 (2/3)

- if 敘述的範例：

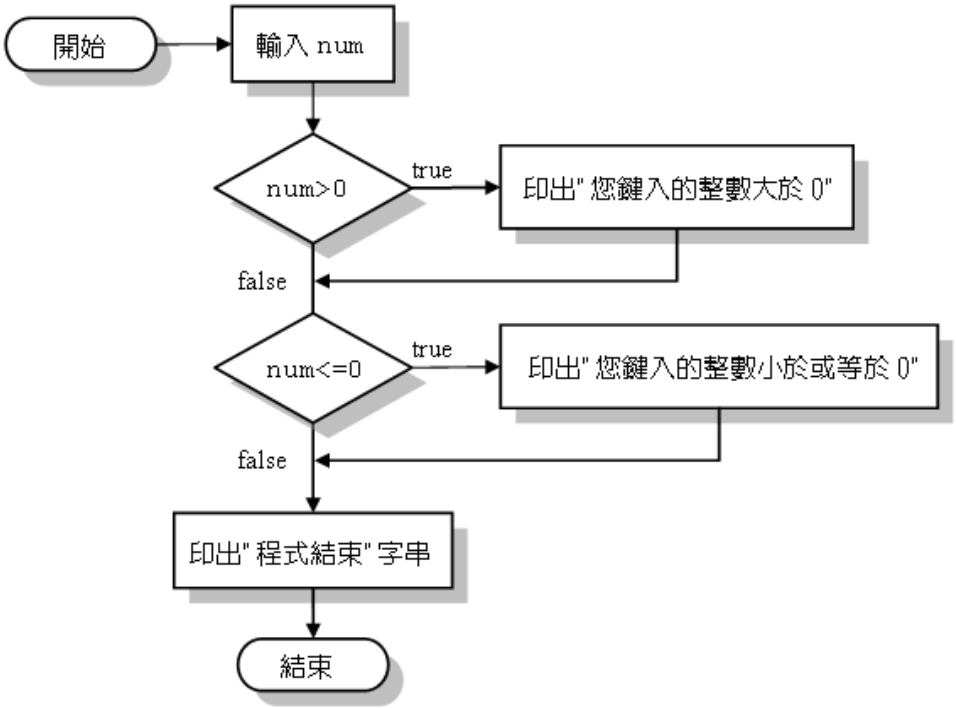



```
01  /* prog6_1, 選擇性結構 if 敘述
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num;
07     printf("請輸入一個整數:")
08     scanf("%d",&num);
09
10     if(num>0)     /* if 敘述，用來判別 num 是否大於 0 */
11        printf("您鍵入的整數大於 0\n");
12
13     printf("程式結束\n");
14
15     system("pause");
16     return 0;
17  }
```

```
/* prog6_1 OUTPUT--

請輸入一個整數:58
您鍵入的整數大於 0
程式結束
--------------------*/
```

if 敘述的使用 (3/3)

- 判別數字是否大於0：



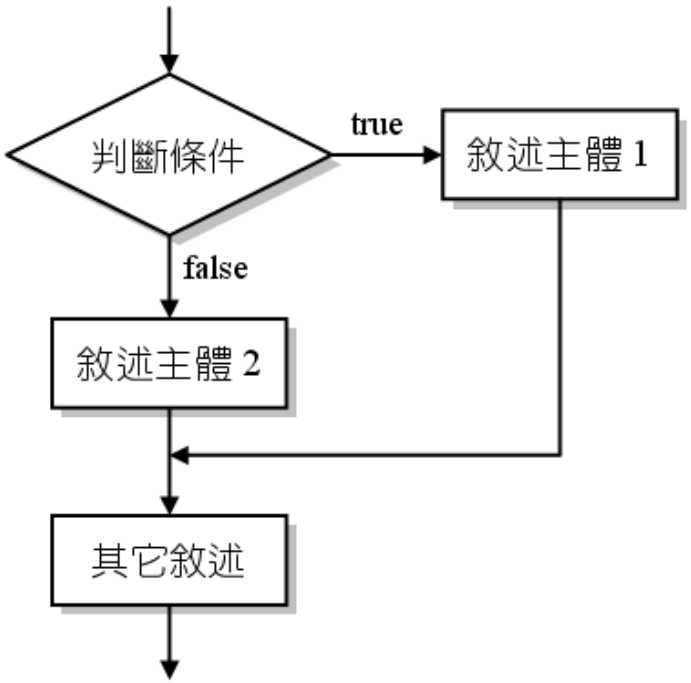

```
01  /* prog6_2, 使用兩個 if 敘述來判斷
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num;
07
08     printf("請輸入一個整數:");
09     scanf("%d",&num);
10     if(num>0)            /* if 敘述，用來判別 num 是否大於 0 */
11        printf("您鍵入的整數大於 0\n");
12     if(num<=0)           /* if 敘述，用來判別 num 是否小於等於 0 */
13        printf("您鍵入的整數小於或等於 0\n");
14     printf("程式結束\n");
15     system("pause");
16     return 0;
17  }
```

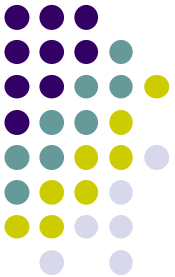
```
/* prog6_2 OUTPUT---

請輸入一個整數:-43
您鍵入的整數小於或等於 0
程式結束
----------------------*/
```

if-else敘述 (1/2)

- if-else敘述
- 當條件成立，即執行if敘述主體；如果不成立，則執行else後面的敘述主體
- if-else敘述的格式如下

```
if(判斷條件)
{
    敘述主體 1;
}
else
{
    敘述主體 2;
}
```





if-else敘述 (2/2)

- 下面是if-else的範例

```
01  // prog5_1, if-else 敘述
02  #include <iostream>
03  #include <cstdlib>
04  using namespace std;
05  int main(void)
06  {
07      int num=42;
08      if(num%3==0 && num%7==0)
09          cout << num << "可以被 3 與 7 整除" << endl;
10      else
11          cout << num << "不能被 3 與 7 整除" << endl;
12      system("pause");
13      return 0;
14  }
```

!(num%3==0 && num%7==0)

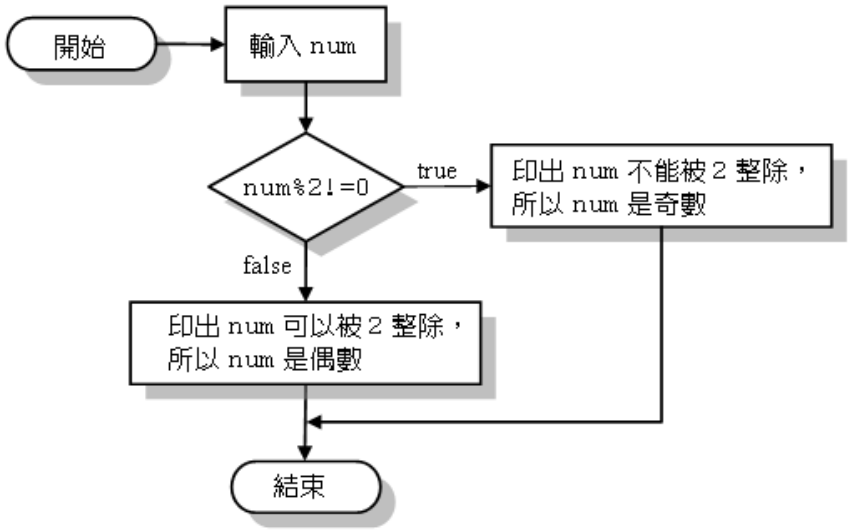
=!(num%3==0)||!(num%7==0)

=num%3!=0||num%7!=0

```
/* prog5_1 OUTPUT---
42 可以被 3 與 7 整除
------------------*/
```

if-else 敘述的範例 (2/2)

```
01  /* prog6_4,if-else 敘述的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num;
07     printf("請輸入一個整數:");
08     scanf("%d",&num);
09     if (num%2!=0) /* 如果 num 不能被 2
10     {
11        printf("%d 不能被 2 整除, ",num);
12        printf("所以%d 是奇數\n",num);        /* 印出 num 為奇數 */
13     }
14     else
15     {
16        printf("%d 可以被 2 整除, ",num);
17        printf("所以%d 是偶數\n",num);        /* 印出 num 為偶數 */
18     }
19     system("pause");
20     return 0;
21  }
```




```
/* prog6_4 OUTPUT---------

請輸入一個整數:34
34 可以被 2 整除, 所以 34 是偶數
-------------------------*/
```

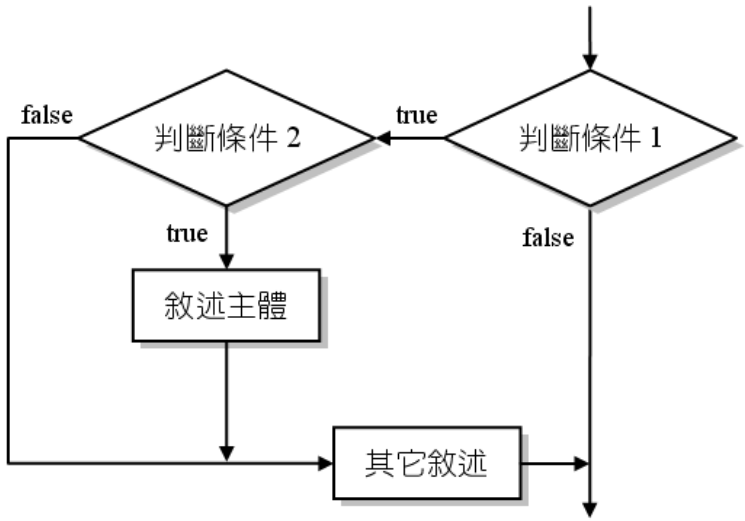
更多的選擇—巢狀if敘述

- 巢狀if：當if敘述中又包含其它if敘述

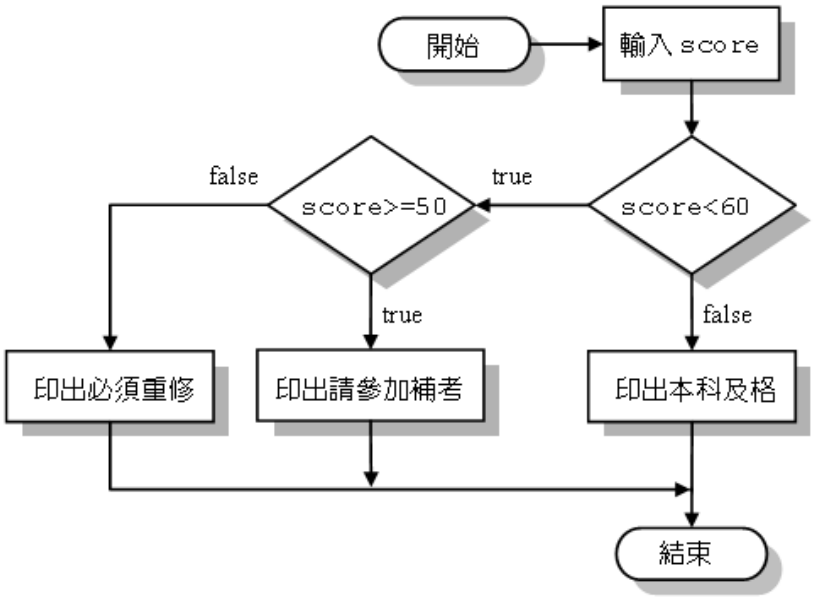
```
if(判斷條件 1)
{
   if(判斷條件 2)
   {
      敘述主體;
   }
   ...
   其它敘述;
}
```

若判斷條件 2 成立，則執行這個部份

若判斷條件 1 成立，則執行這個部份



巢狀 if 敘述的練習




```
01  /* prog6_5, 巢狀 if 敘述的練習
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int score;
07     printf("請輸入成績:");
08     scanf("%d",&score);
09     if (score<60)    /* 如果
10     {
11        if(score>=50) /* 如果 score>=50 */
12           printf("請參加補考\n");
13        else
14           printf("必須重修\n");
15     }
16     else
17        printf("本科及格\n");
18     system("pause");
19     return 0;
20  }
```

一個敘述可省略

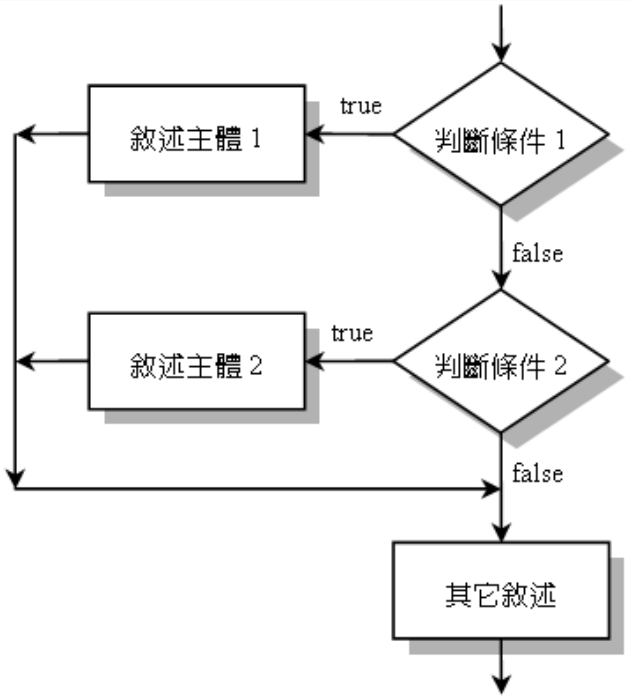
```
/* prog6_5 OUTPUT---

請輸入成績:52
請參加補考
--------------------*/
```

使用 if-else-if 敘述

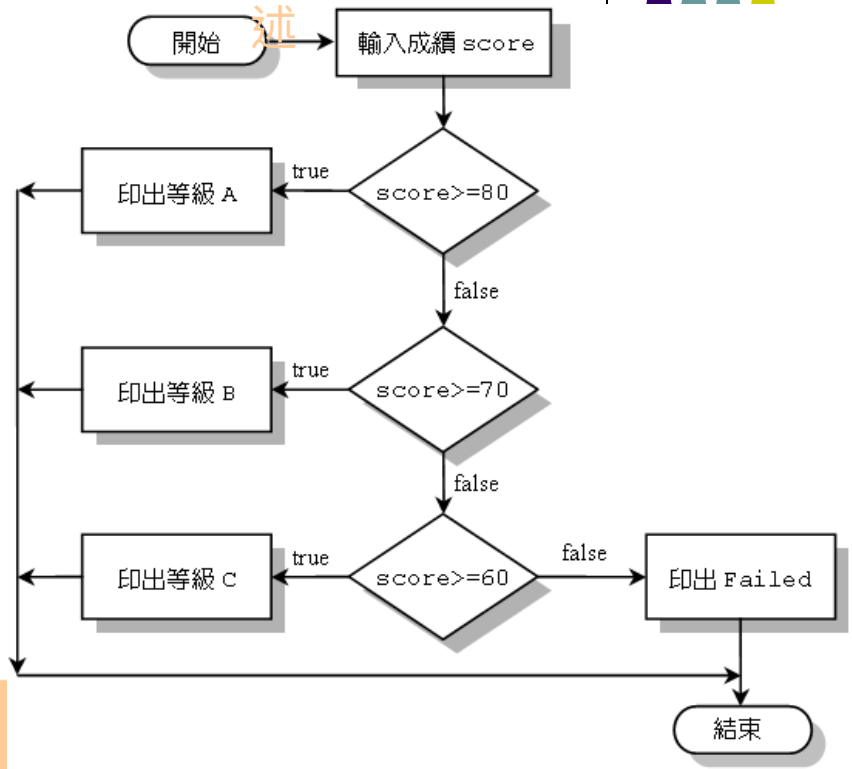
- if-else-if：當 if 判斷不成立，必須進行其它判斷時

if-else-if 敘述的語法

```
if(判斷條件1)
{
   敘述主體1;
}
else if(判斷條件2)
{
   敘述主體2;
}
```

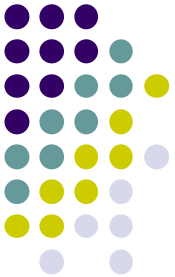


if-else-if 敘述的應用




```
01  /* prog6_6, if-else-if 敘述的應用 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int score;
07     printf("Your score:");
08     scanf("%d",&score);
09     if (score>=80)
10        printf("%d is A\n",score);
11     else if (score>=70)
12           printf("%d is B\n",score);        /* 印出 B */
13        else if (score>=60)
14              printf("%d is C\n",score);    /* 印出 C */
15           else
16              printf("Failed!!\n");
17     system("pause");
18     return 0;
19  }
```

```
/* prog6_6 OUTPUT---

Your score:58
Failed!!
--------------------*/
```



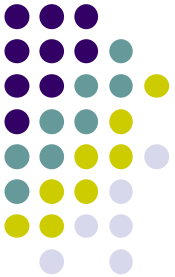
if 與 else 的配對問題 (1/2)

- else會與離它最近的 if 配對：

```
01  /* prog6_7, if-else 配對問題(一) */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num;
07     printf("請輸入一個整數:");
08     scanf("%d",&num);
09
10     if (num>=0)
11       if(num<=10)
12          printf("數字介於 0 到 10 之間\n");
13       else
14          printf("數字大於 10\n");
15
16     system("pause");
17     return 0;
18  }
```

```
/* prog6_7 OUTPUT---

請輸入一個整數: 7
數字介於 0 到 10 之間
--------------------*/
```

第 13 行的 else 與第 11 行的 if 配對



if 與 else 的配對問題 (2/2)

- else 如果要與離它較遠的 if 配對，需加大括號：

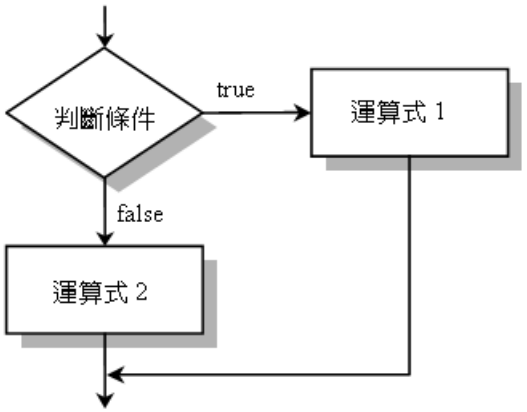
```
01  /* prog6_8, if-else 配對問題(二) */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int num;
07     printf("請輸入一個整數:");
08     scanf("%d",&num);
09     if (num>=0)
10     {
11        if(num<=10)
12           printf("數字介於 0 到 10 之間\n");
13     }
14     else     /* 如果第 10 行的 if 敘述不成立 */
15        printf("數字小於 0\n");
16     system("pause");
17     return 0;
18  }
```

```
/* prog6_8 OUTPUT---

請輸入一個整數:-26
數字小於 0
--------------------*/
```

條件運算子 (1/2)

- 條件運算子（conditional operator）可代替if-else敘述

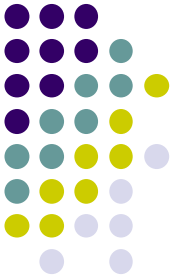
條件運算子	意義
?:	根據條件的成立與否，來決定是哪一個運算式會被執行



- 條件運算子的格式如下

```
傳回值 = 判斷條件 ? 運算式 1 : 運算式 2 ;
```

第一個運算元：判斷條件
第二個運算元：運算式 1
第三個運算元：運算式 2



條件運算子 (2/2)

- 想把運算的結果設給某個變數，可用下面的語法：

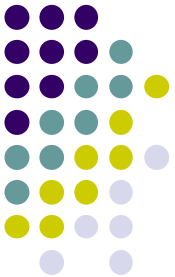
把運算結果設給變數

```
變數名稱 = 條件判斷 ? 運算式1 : 運算式2
```

- 上面的語法只需一行，但相當於下面的 if-else 敘述：

對等的 if-else 敘述

```
if (條件判斷)
   變數名稱 = 運算式1;
else
   變數名稱 = 運算式2;
```



條件運算子 (2/2)

- 條件運算子的使用範例

```
01  // prog5_2, 條件運算子?:的使用
02  #include <iostream>
03  #include <cstdlib>
04  using namespace std;
05  int main(void)
06  {
07      int a=5,b=12,min;
08      min=(a<b)?a:b;          // 利用條件運算子判斷 a 與 b 何者為小
09      cout << "a=" << a << ", b=" << b << endl;
10      cout << min << "是較小的數" << endl;
11
12      system("pause");
13      return 0;
14  }
```

min=(a<b && a<c)? a:(b<c)? :b

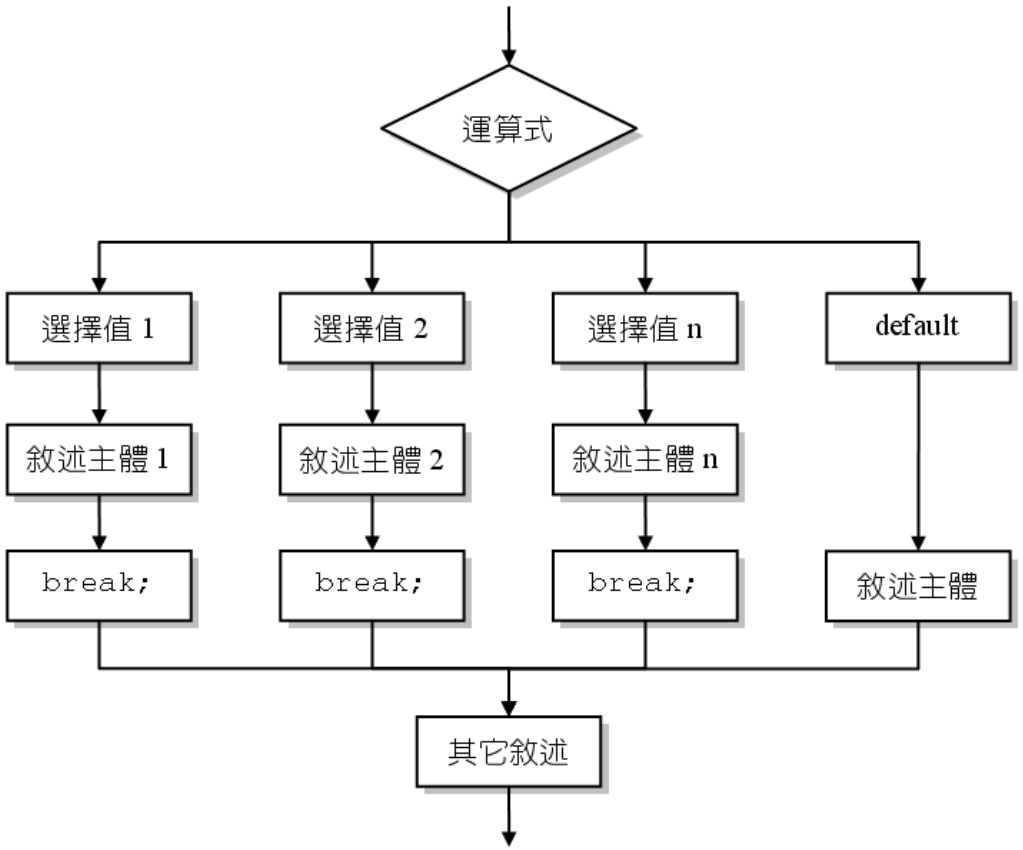
```
/* prog5_2 OUTPUT----

a=5, b=12
5 是較小的數
--------------------*/
```

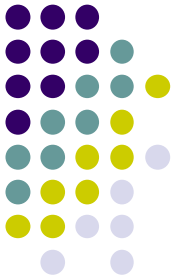
switch敘述 (1/3)

- switch敘述的格式如下

```
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;
              ...
    case 選擇值 n:
            敘述主體 n;
            break;
    default:
            敘述主體;
}
```

只能是整數or字



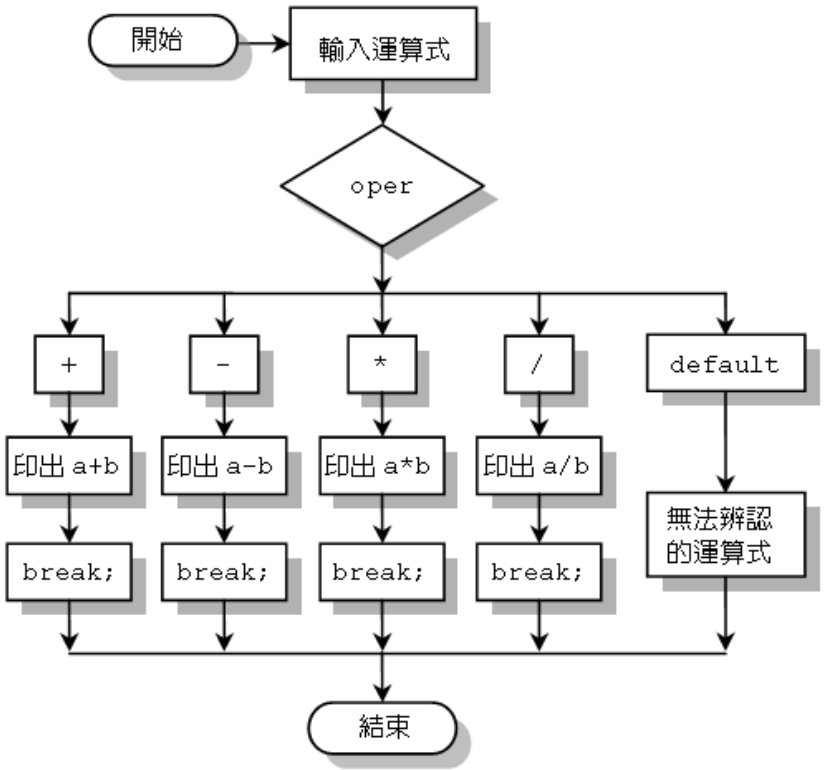


switch敘述 (2/3)

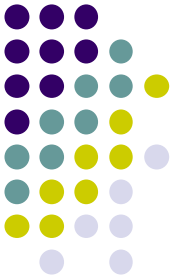
- witch敘述執行的流程
  - switch敘述先計算括號中運算式的運算結果
  - 如果某個case的選擇值符合運算式的結果，就會執行該case所包含的敘述，直到執行至break敘述後才跳離整個switch敘述
  - 若是所有case的選擇值皆不適合，則執行default之後所包含的敘述，執行完畢即離開switch敘述
  - 如果沒有定義default的敘述，則直接跳離switch敘述

switch 敘述的範例 (1/2)

- 依據選擇值進行四則運算：



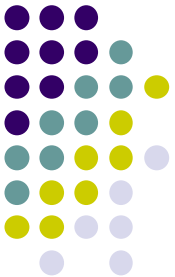

```
01  /* prog6_10, switch 敘述的使用範例 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06     int a,b;
07     char oper;
08     printf("請輸入運算式(例如:3+2): ");
09     scanf("%d %c %d",&a,&oper,&b);
10
11     switch(oper)
12     {
13        case '+':
14           printf("%d+%d=%d\n",a,b,a+b);          /* 印出 a+b */
15           break;
```



switch 敘述的範例 (1/2)

```
16          case '-':
17             printf("%d-%d=%d\n",a,b,a-b);          /* 印出 a-b */
18             break;
19          case '*':
20             printf("%d*%d=%d\n",a,b,a*b);          /* 印出 a*b */
21             break;
22          case '/':
23             printf("%d/%d=%.3f\n",a,b,(float)a/b); /* 印出 a%b */
24             break;
25          default:
26             printf("無法辨認的運算式!!\n");          /* 印出字串 */
27       }
28       system("pause");
29       return 0;
30    }
```

```
/* prog6_10 OUTPUT--------

請輸入運算式(例如:3+2): 100/7
100/7=14.286
-----------------------------*/
```

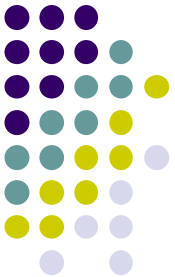


將不同的選擇值並列 (1/2)

```
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;
```

將不同的選擇值並列 (2/2)

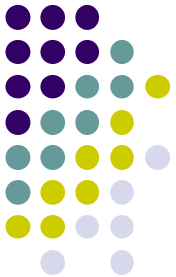
```
20          case 'c':   /* 輸入 c 或 C 時印出 Be study hard! */
21          case 'C':
22             printf("Be study hard!\n");
23             break;
24          default:    /* 輸入其他字元時印出 Failed! */
25             printf("Failed!\n");
26       }
27       system("pause");
28       return 0;
29    }

/* prog6_11 OUTPUT---

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



不加 break 的 switch 敘述 (1/2)

- 如果沒有加break，可能會造成switch執行錯誤：

```
01  /* prog6_12, 忘了加上 break 的 switch 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");
```



不加 break 的 switch 敘述 (2/2)

```
15          case 'b':   /* 輸入 b 或 B 時印出 Good! */
16          case 'B':
17             printf("Good!\n");  //fall through
18          case 'c':   /* 輸入 c 或 C 時印出 Be study hard! */
19          case 'C':
20             printf("Be study hard!\n");
21          default:    /* 輸入其他字元時印出 Failed! */
22             printf("Failed!\n");
23       }
24       system("pause");
25       return 0;
26    }

/* prog6_12 OUTPUT--

Input grade:b
Good!
Be study hard!
Failed!
----------------------*/
```

問題1：if 與 switch 可以互相轉換嗎？

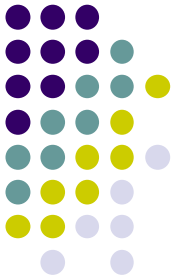
問題2：if 與 switch 的使用時機？

補充: switch 敘述比對數值範圍

- 使用...來設定一個範圍的數值，而不用連續的撰寫case來比對，但數值範圍不允許重疊

```
switch(i)
{
    case 10 ... 30:
        // statements
    case 31 ... 1000:
        // statements
    default:
        // statements
}
```

要有空格



-The End-