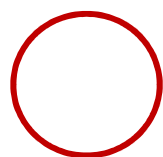


Scanf, operation, if

I-Fen Chao

提高程式的可讀性

```
01  /* prog2 6,有縮排的程式碼  */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int i;
08      for(i=1;i<=2;i++)
09      {
10          printf("Cats are running, ");
11          printf("dogs are chasing.\n");
12      }
13      system("pause");
14      return 0;
15  }
```



```
01  /* prog2 7, 沒有縮排的程式碼  */
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int i;
08      for(i=1;i<=2;i++)
09      {
10          printf("Cats are running, ");
11          printf("dogs are chasing.\n");
12      }
13      system("pause");
14      return 0;
15  }
```



提高程式的可讀性

```
/* prog2_7, examples */  
/* created by Wien Hong */
```

} 以註解符號對每一行做註解

```
/*  
    This paragraph demonstrates the  
    capability of comments used by C  
    November 06 2003  
*/
```

} 於「/*」和「*/」符號之間的
文字均是註解

Printf 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
\r	歸位		
\t	跳格		
\'	印出單引號		

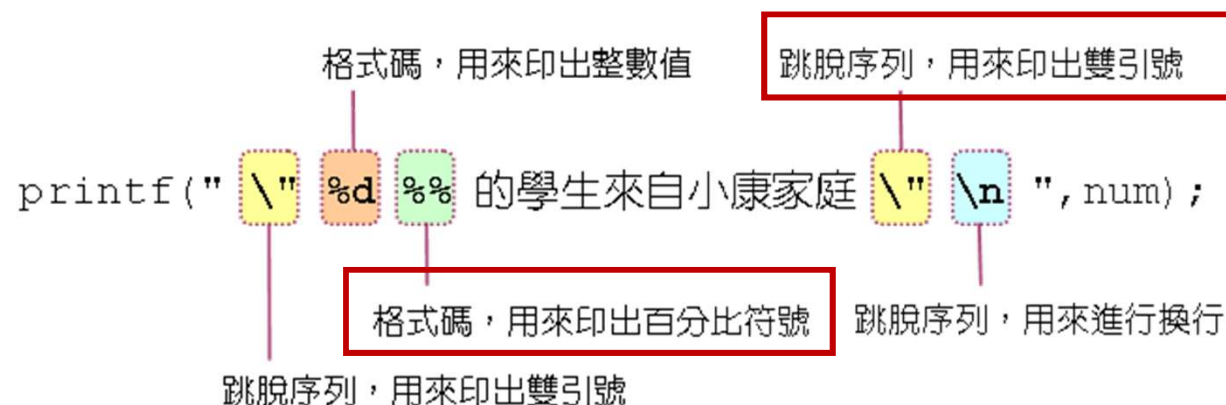
Printf

跳脫序列與格式碼的應用

```

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  }

```



Printf

控制輸出欄位的寬度

```

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  }

```

n	u	m	1	=						3	2	公	里
---	---	---	---	---	--	--	--	--	--	---	---	---	---

%6d, 佔 6 格, 靠右對齊

n	u	m	2	=	1	0	2	4				公	里
---	---	---	---	---	---	---	---	---	--	--	--	---	---

%-6d, 佔 6 格, 靠左對齊

n	u	m	3	=		1	2	.	3	5	英	哩
---	---	---	---	---	--	---	---	---	---	---	---	---

%6.2f, 佔 6 格, 靠右對齊

printf() 函數的修飾子

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

					1	2	3	4	5
--	--	--	--	--	---	---	---	---	---

12345 %+d

+	1	2	3	4	5
---	---	---	---	---	---

12345 %09d

0	0	0	0	1	2	3	4	5
---	---	---	---	---	---	---	---	---

12345 %-10d

1	2	3	4	5					
---	---	---	---	---	--	--	--	--	--

資料內容 格式 執行結果

12345 % d

	1	2	3	4	5
--	---	---	---	---	---

123.456 %7.2f

	1	2	3	.	4	6
--	---	---	---	---	---	---

123.456 %010.3f

0	0	0	1	2	3	.	4	5	6
---	---	---	---	---	---	---	---	---	---

123.456 %+10.4f

	+	1	2	3	.	4	5	6	0
--	---	---	---	---	---	---	---	---	---

Reading Data from the Keyboard

scanf_s() function

- 語法(syntax)

`scanf_s (format_string, argument_list) ;`

```
int income;  
float expense;  
  
scanf_s ("%d", &income);  
scanf_s ("%f", &expense);
```

- The ‘address of’ operator **&**
 - to get the address of variables

錯誤的寫法

```
int i;
```

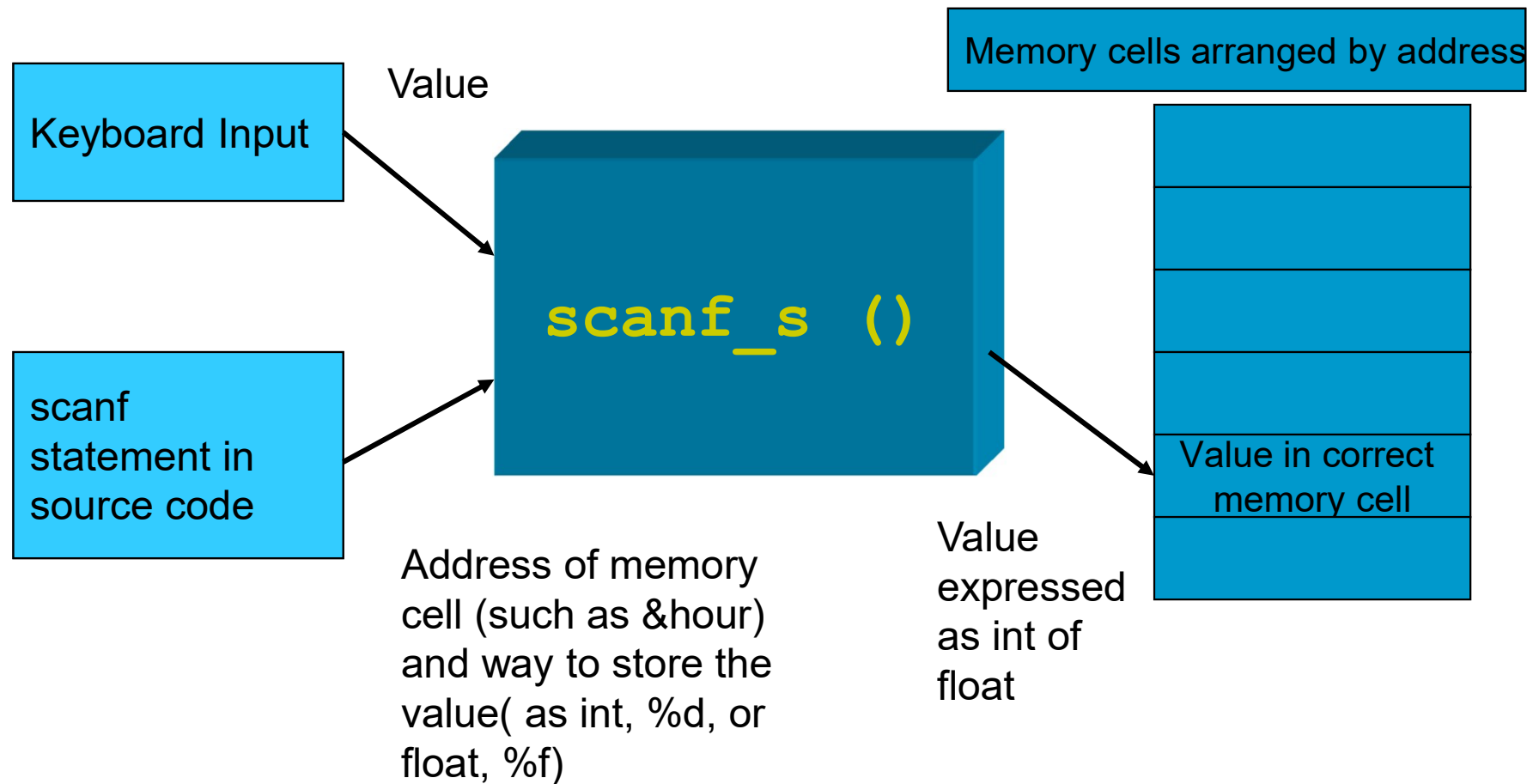
```
scanf_s("%d", i);
```

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

會產生哪樣的錯誤？

發生錯誤時，我看得懂嗎？

Operation of *scanf*() function



輸入函數scanf_s() (以下請自行改成scanf_s)

/* prog4_9 OUTPUT---

請輸入一個整數: 78

num=78

-----*/

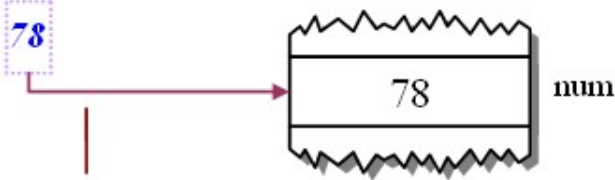
```

01  /* prog4_9, 使用 scanf() 函數 */
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);          /* 由鍵盤輸入整數，並指定給 num 存放 */
10      printf("num=%d\n",num);    /* 印出 num 的內容 */
11
12      system("pause");
13      return 0;
14  }

```

scanf("%d",&num);

請輸入一個整數: 78



將數值 78 寫到變數 num 裡

輸入字元應注意的事項

/* prog4_13 OUTPUT----

Input a character: R → 先輸入一個空白鍵再輸入 R
ch= , ascii code is 32
-----*/

```
01  /* prog4_13, 輸入字元 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      char ch;
07
08      printf("Input a character:");
09      scanf("%c",&ch);          /* 由鍵盤輸入字元並指定給變數 ch */
10      printf("ch=%c, ascii code is %d\n",ch,ch);
11      system("pause");
12      return 0;
13  }
```

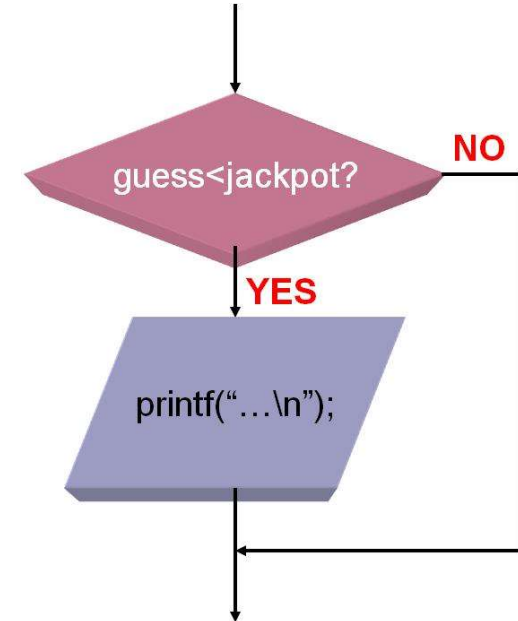
IF control structure - Relational Expression

- Relational expression

left_operand **relational_operator** *right_operand*

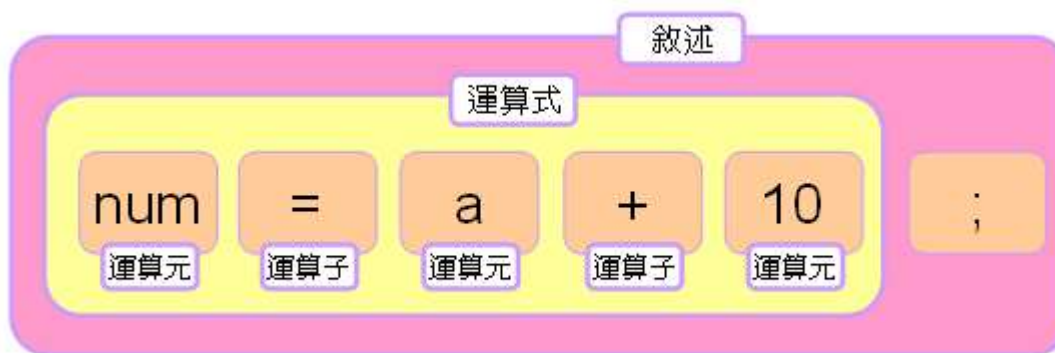
- Example of relational expression

`guess < jackpot`



運算式、運算元與運算子

- 運算式由運算元與運算子組成
 - 運算式：**expression**
 - 運算元：**operand**，如變數**sum**，或常數**10**等
 - 運算子：**operator**，如「+」、「-」、「*」與「/」等符號



運算子：設定

- 「設定」運算子可將變數設值
- 等號(=)是「設定」的意思，如下面的範例：

age = 14 ;

變數名稱 設定值

運算子：設定

/* prog5_1 OUTPUT---

age=14

將 age 加 1 之後, age=15

-----*/

```
01  /* prog5_1, 設定運算子「=」 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int age=14;
07
08      printf("age=%d\n", age);
09      age=age+1;    /* 將 age 加 1 後，再設回給 age 存放 */
10      printf("將 age 加 1 之後, age=%d\n", age);
11
12      system("pause");
13      return 0;
14  }
```


一元運算子

```
/* prog5_2 OUTPUT--
```

```
a=0, !a=1  
b=6, !b=0
```

```
-----*/
```

- 一元運算子（**unary operator**）只需要一個運算元
 - `+3;` `/* 表示正3，3 為運算元 */`
 - `-a;` `/* 表示負a，a 為運算元 */`
 - `!a;` `/* NOT運算，若a為0，則!a為1，若a不為0，則!a為0 */`

```
01  /* prog5_2, 「!」運算的用法 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      int a=0;
07      int b=6;
08      printf("a=%d, !a=%d\n",a,!a);    /* 印出 a 及!a 的值 */
09      printf("b=%d, !b=%d\n",b,!b);    /* 印出 b 及!b 的值 */
10
11      system("pause");
12      return 0;
13  }
```

餘數運算子

```
/* prog5_3 OUTPUT---
```

```
12%4=0
```

```
12%5=2
```

```
12%16=12
```

```
-----*/
```

- 下面的範例是餘數運算子「%」的練習：
 - 要印出「%」符號，可用格式碼「%%」

```
01  /* prog5_3, 餘數運算子的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      printf("12%%4=%d\n",12%4);      /* 求出 12/4 的餘數 */
07      printf("12%%5=%d\n",12%5);      /* 求出 12/5 的餘數 */
08      printf("12%%16=%d\n",12%16);    /* 求出 12/16 的餘數 */
09
10      system("pause");
11      return 0;
12  }
```

關係運算子與 if 敘述

if 敘述的格式

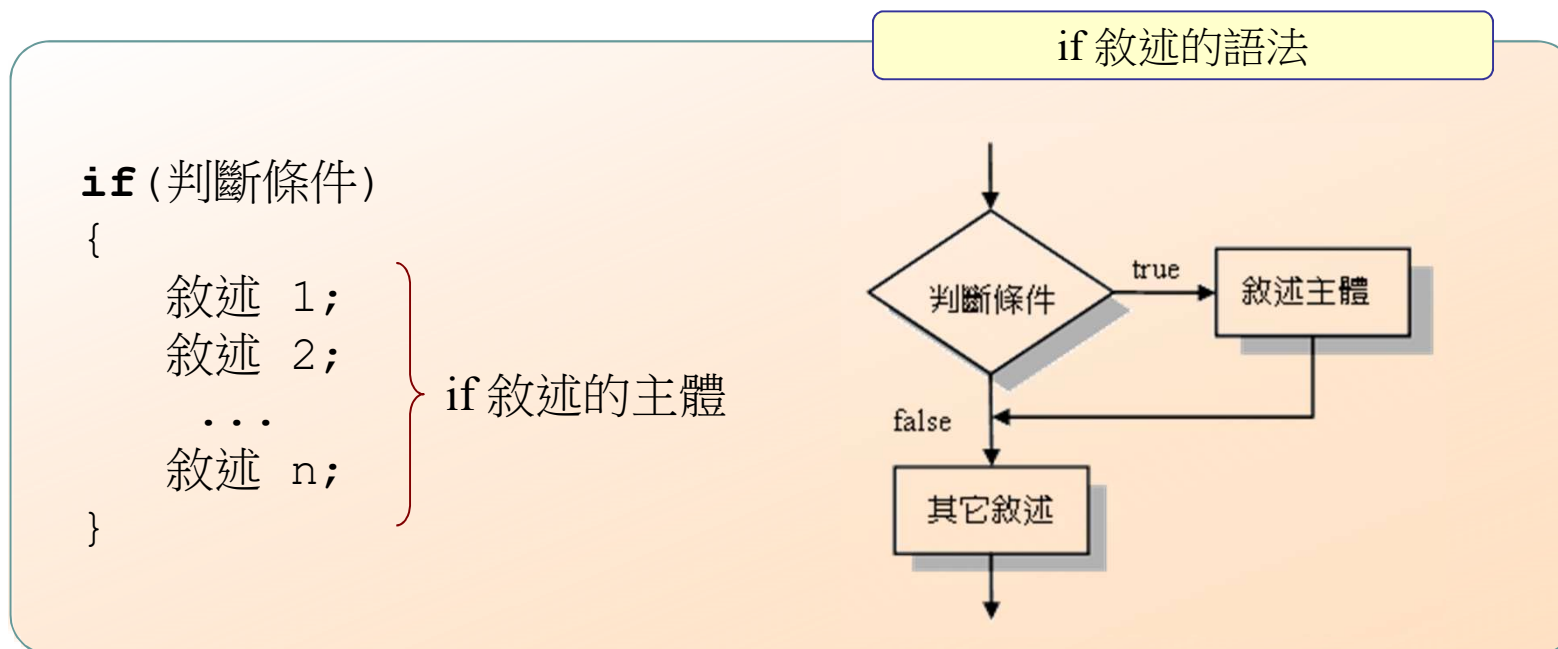
if (判斷條件)
敘述主體;

表 5.1.4 關係運算子的說明

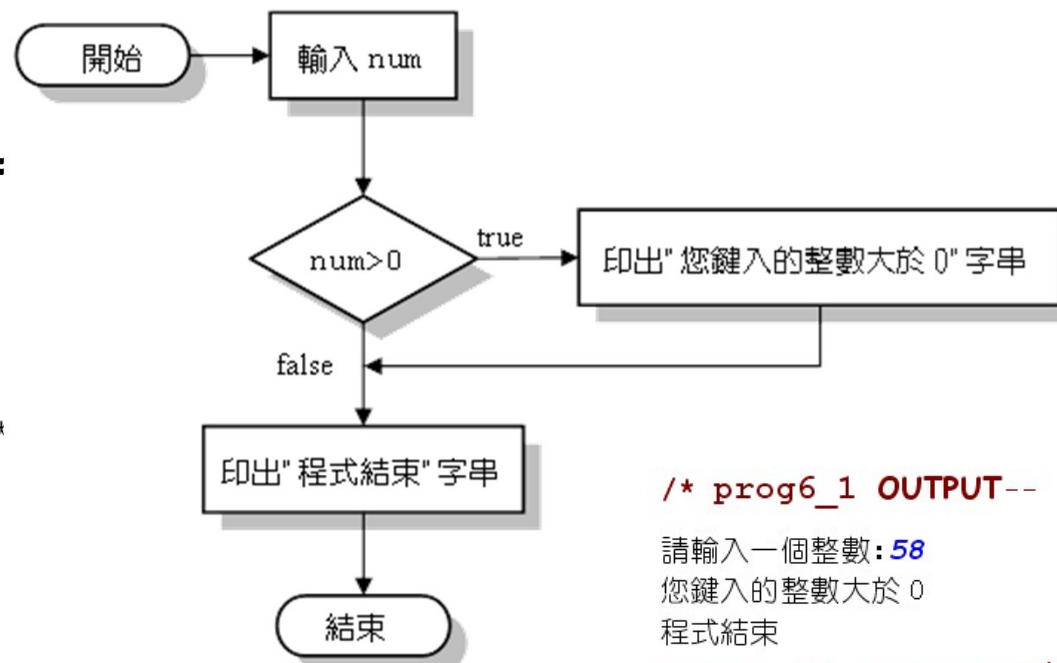
關係運算子	意義	範例	說明
>	大於	$a > b$	判別 a 是否大於 b
<	小於	$a < b$	判別 a 是否小於 b
>=	大於等於	$a \geq b$	判別 a 是否大於等於 b
<=	小於等於	$a \leq b$	判別 a 是否小於等於 b
==	等於	$a == b$	判別 a 是否等於 b
!=	不等於	$a != b$	判別 a 是否不等於 b

A block *if* statement

- if 敘述可依據條件式是否成立，來決定程式的流程



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

-----*/

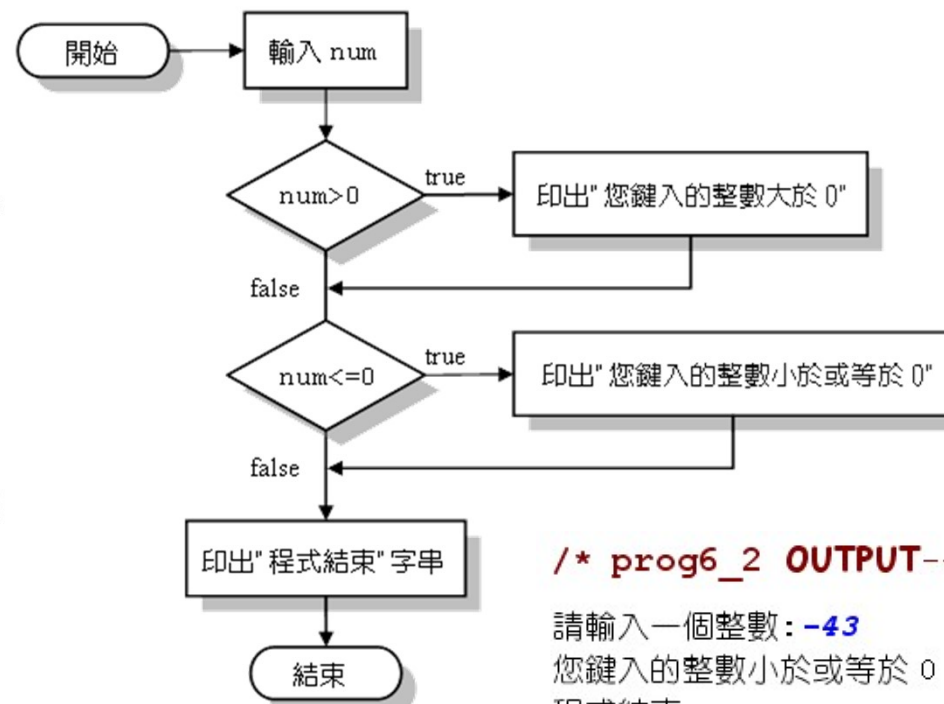
If

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

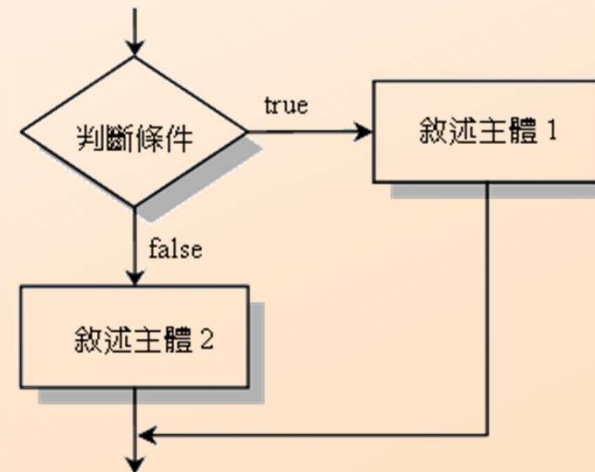
-----*

if-else 敘述

- **if-else** 敘述可用於：成立時則執行**A**，否則執行**B**

if-else 敘述的語法

```
if (判斷條件)
{
    敘述主體1; /*若成立，執行此部份*/
}
else
{
    敘述主體2; /*不成立，執行此部份*/
}
```



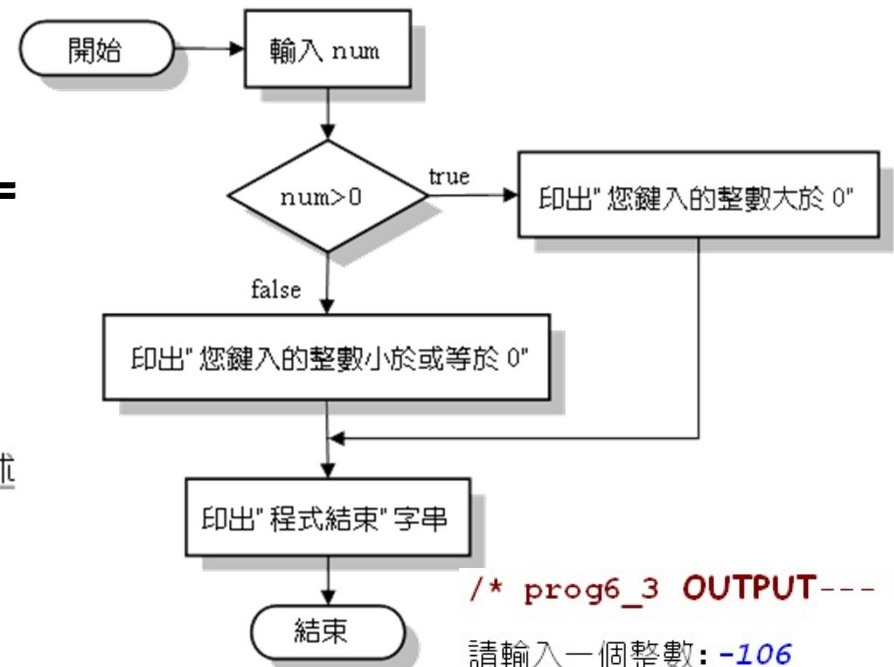
if-else

判別數字是否大於0

```

01  /* prog6_3, 選擇性結構 if-else 敘述
02  #include <stdio.h>
03  #include <stdlib.h>
04
05  int main(void)
06  {
07      int num;
08      printf("請輸入一個整數:");
09      scanf("%d",&num);
10      if(num>0)      /* 判別 num 是否大於 0 */
11          printf("您鍵入的整數大於 0\n");
12      else            /* 如果 num 不是大於 0，則執行下面的敘述 */
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_3 OUTPUT---

請輸入一個整數: -106

您鍵入的整數小於或等於 0

程式結束

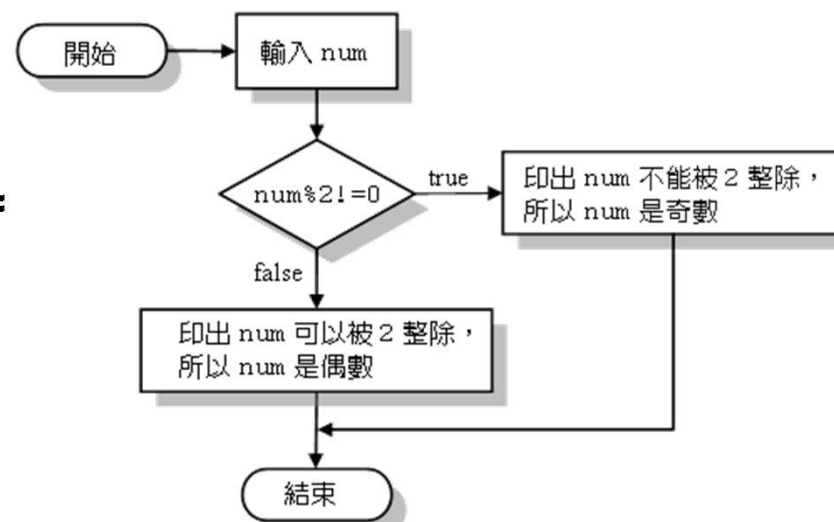
-----*/

if-else 敘述的範例

```

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 整除 */
10      {
11          printf("%d 不能被 2 整除, ", num);
12          printf("所以%d 是奇數\n", num);
13      }
14      else
15      {
16          printf("%d 可以被 2 整除, ", num);
17          printf("所以%d 是偶數\n", num);
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

/* 印出 num 為奇數 */

/* 印出 num 為偶數 */

Source code

```
#include <stdio.h>
void main(void)
{
    int i, guess, jackpot=8;

    printf("Try to guess the jackpot number \nbetween 1 and 10!\n");
    printf("Please type a number.\n");
    scanf ("%d",&guess);

    if (guess<jackpot)
        printf("Try a bigger number\n");

    if (guess>jackpot)
        printf("Try a smaller number\n");

    if (guess==jackpot)
        printf("Verify your guess by typing it one more time\n");

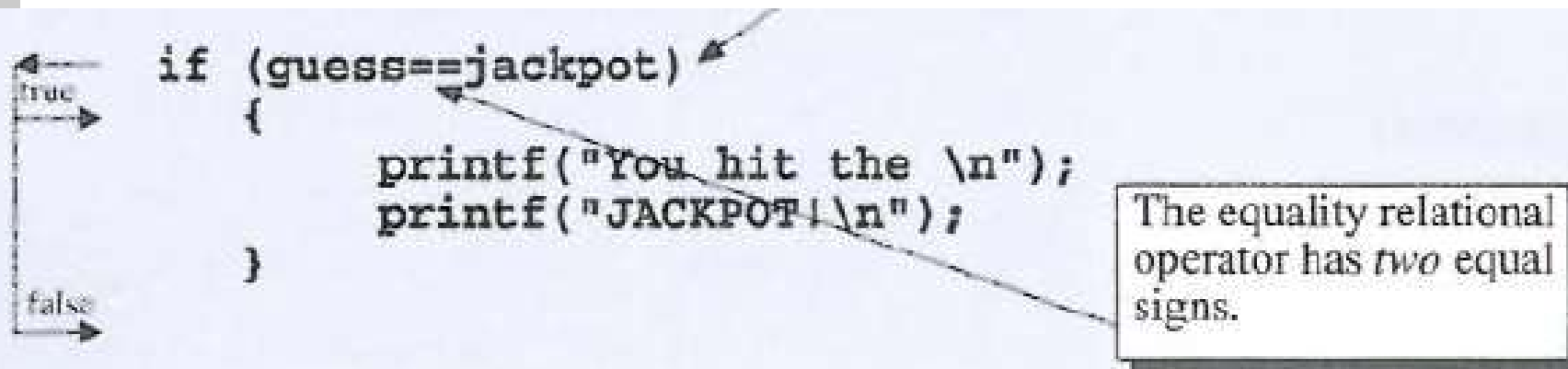
    scanf ("%d", &guess);
```

Relational expressions. The result of a relational expression can be regarded as only true or false.

true
false

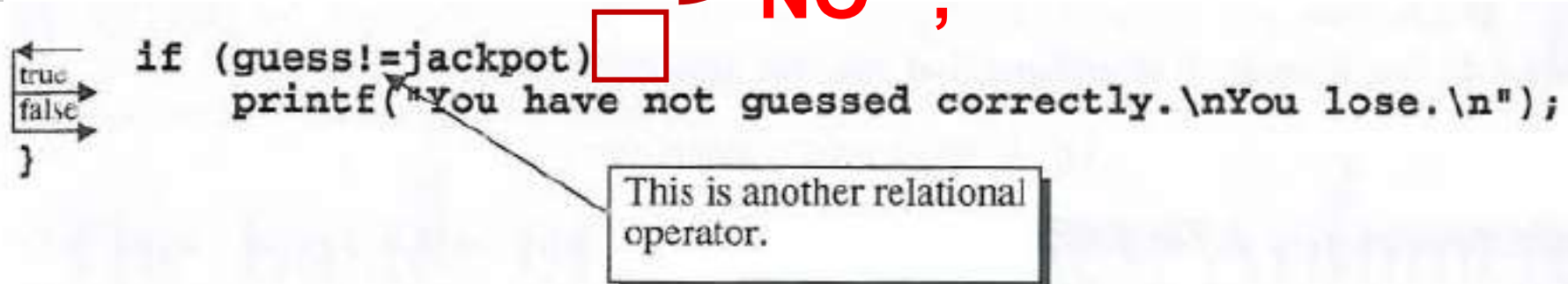
true
false

true
false



不要加分號“;”

NO “;”



An example

```
01  /* prog5_4, 關係運算子的練習 */
02  #include <stdio.h>
03  #include <stdlib.h>
04  int main(void)
05  {
06      if(5>2)    /* 判斷 5>2 是否成立 */
07          printf("5>2 成立\n");
08
09      if(1)      /* 1 代表 true, 所以 if 的判斷結果會成立 */
10          printf("此行一定會被執行\n");
11
12      if(3==8)   /* 判斷 3 是否等於 8 */
13          printf("3==8 成立\n");
14
15      system("pause");
16      return 0;
17  }
```

Logical Expressions: Logical operator in C

Operator	Name	Operation	Operator Type
!	Logical NOT	Negation	Unary
&&	Logical AND	Conjunction	Binary
	Logical OR	Inclusive disjunction	Binary

if(x > 5 || < -7) 這樣寫不對!

if(x > 5 || x < -7) 這樣寫才對!

Source code

```
#include <stdio.h>
void main(void)
{
    int x=5,y=0;

    printf("x=%2d, y=%2d\n\n",x,y);

    if (x>0 && y>=0)
        printf("x is greater than 0 and"
               "y is greater than or equal to 0\n\n");

    if (x==0 || y==0)
        printf("x equals 0 or y equals 0\n\n");

    if (! (x==y))
        printf("x is not equal to y\n");
}
```

Logical operators are commonly used with relational expressions.

Output

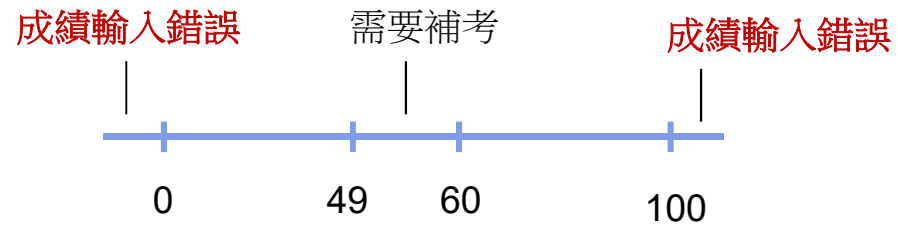
Result of logical expression

A	B	A&&B	A B	!A	!B
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

The logical value of a single variable

- false
 - zero
- true
 - nonzero, include negative values

An example



```

01  /* prog5_6, 邏輯運算子的應用 */
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
10      /* 若成績超出 0 到 100 之間 */
11      printf("成績輸入錯誤!!\n");
12
13      /* 若成績介於 50 到 59 之間 */
14      printf("需要補考!!\n");
15      system("pause");
16      return 0;
17  }

```

Confusing Equality (==) and Assignment (=) Operators

- For example, suppose we intend to write

```
if ( payCode == 4 )  
    printf( "You get a bonus!" );
```

but we accidentally write

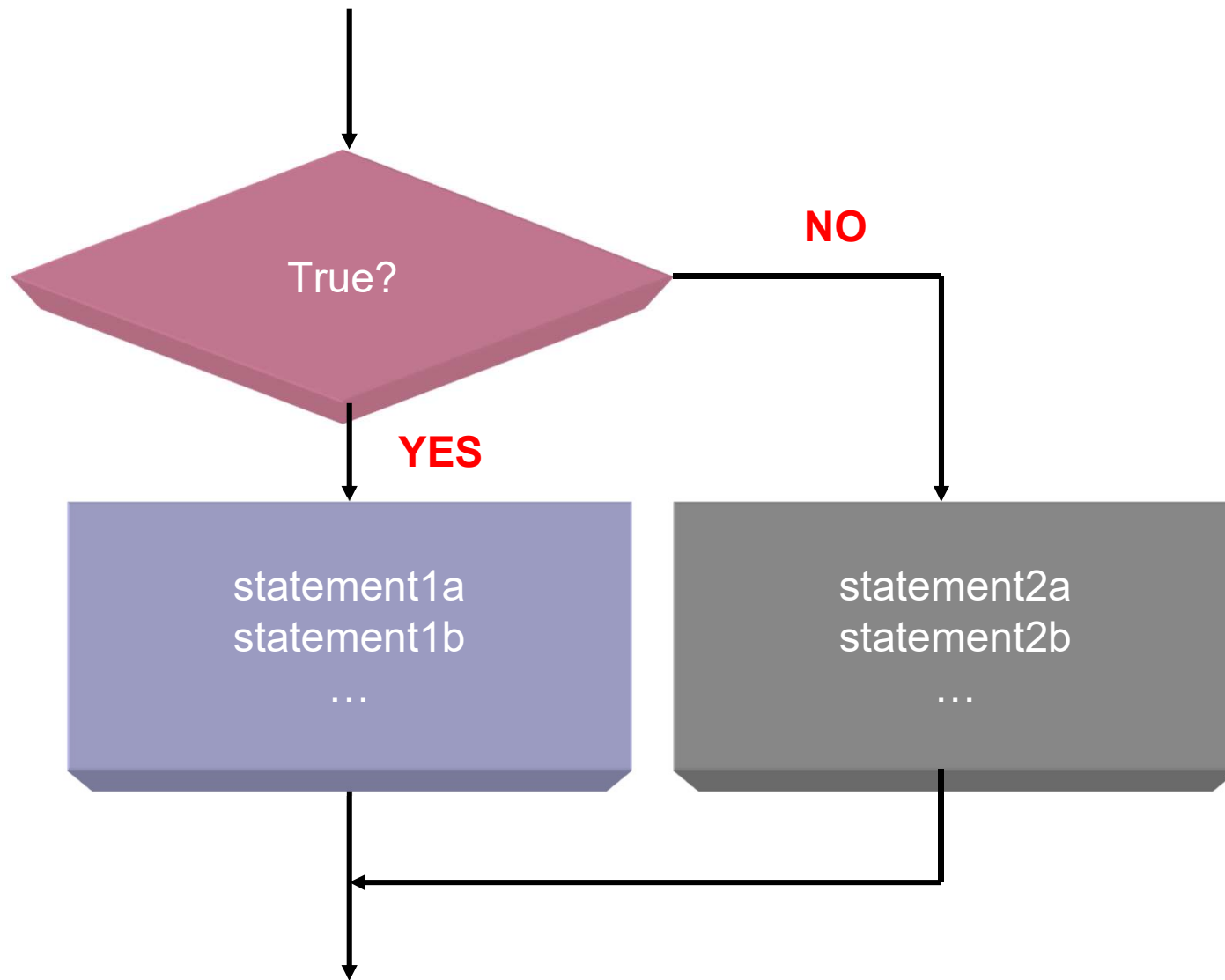
```
if ( payCode = 4 )  
    printf( "You get a bonus!" );
```

Simple *if-else* control structure

- Syntax

```
if (expression)
{
    statement1a;
    statement1b;
    ...
}
else
{
    statement2a;
    statement2b;
    ...
}
```

NO expression!!

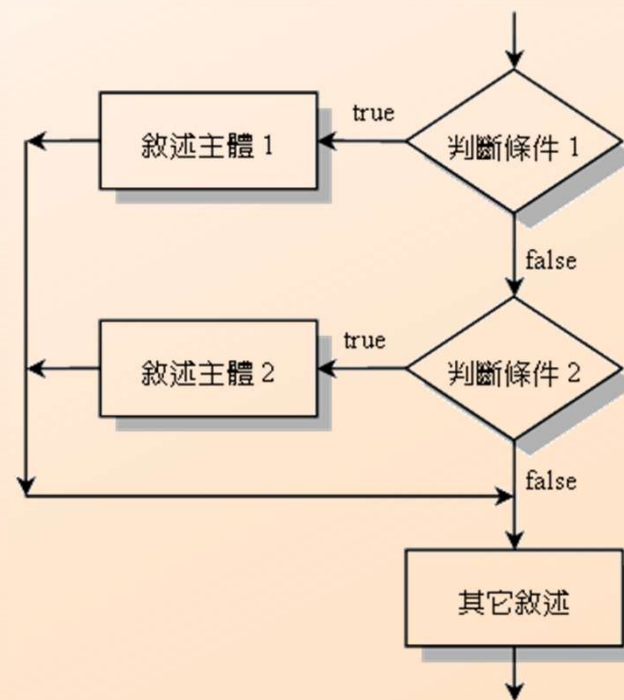


if-else-if 敘述

- **if-else-if**：當 if 判斷不成立，必須進行其它判斷時

```
if (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 敘述的語法



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

```
if ( grade >= 90 )  
    printf( "A\n" );  
else if ( grade >= 80 )  
    printf( "B\n" );  
else if ( grade >= 70 )  
    printf( "C\n" );  
else if ( grade >= 60 )  
    printf( "D\n" );  
else  
    printf( "F\n" );
```

如果要執行的
statement只有一
行，可以不加{ }

如果grade=95

→測試一次

→print 'A'

後面程式不再執行

```
if ( grade >= 90 )  
    printf( "A\n" );
```

如果grade=73

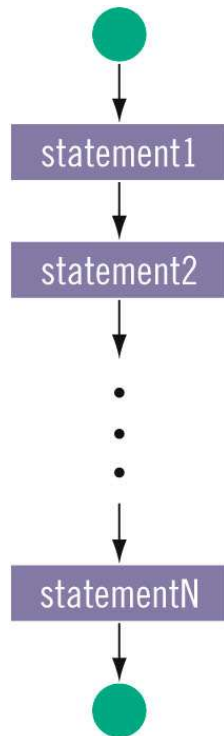
→測試三次

→print 'C'

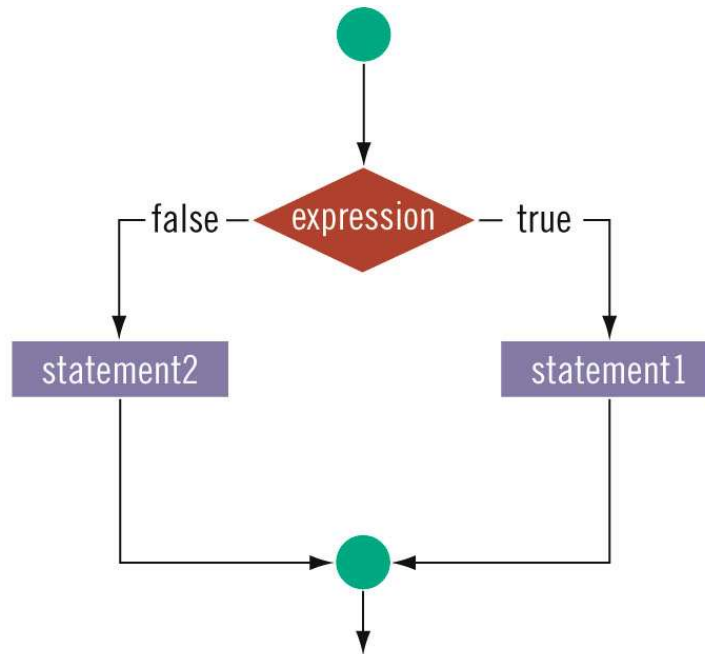
後面程式不再執行

```
if ( grade >= 90 )  
    printf( "A\n" );  
else if ( grade >= 80 )  
    printf( "B\n" );  
else if ( grade >= 70 )  
    printf( "C\n" );
```

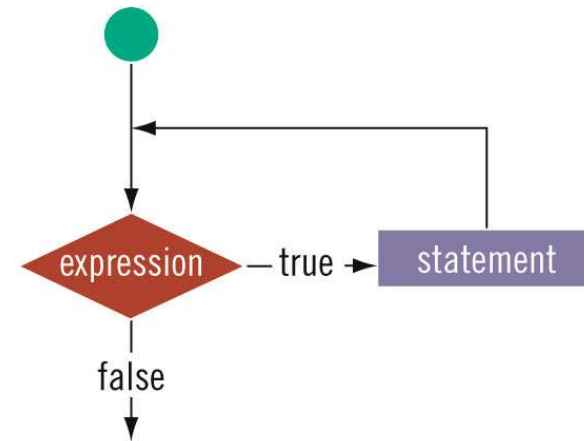
Flow Chart



a. Sequence



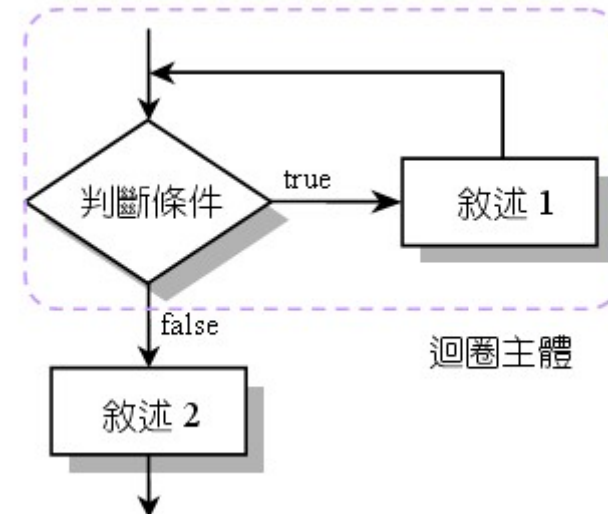
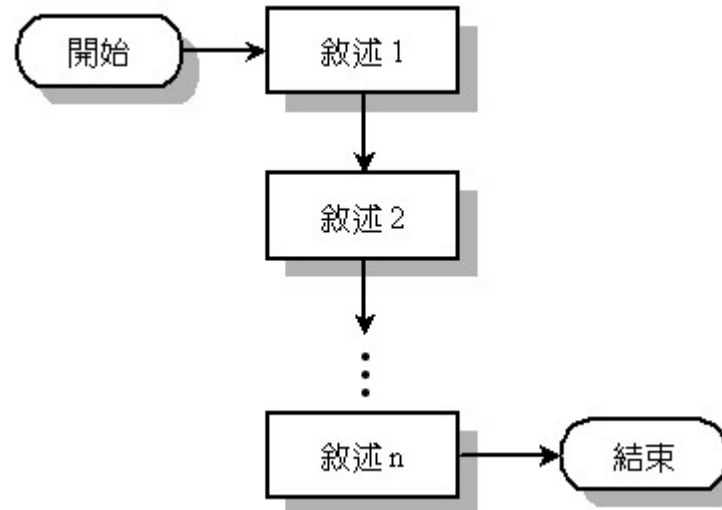
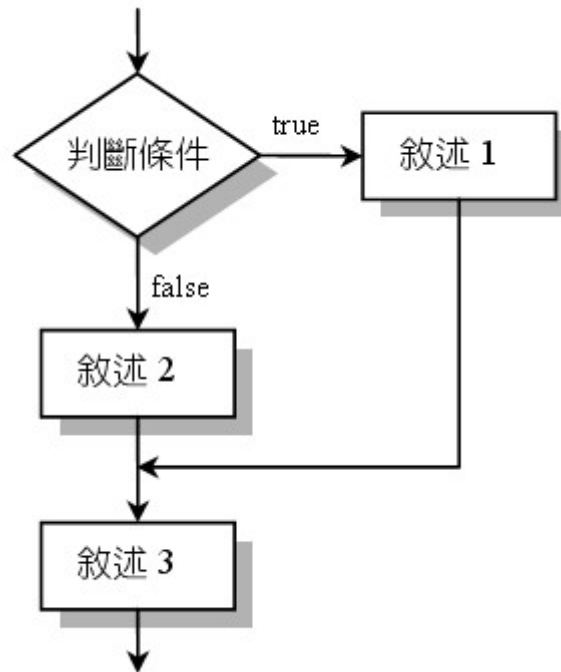
b. Selection



c. Repetition

Flow Chart

- Sequence structure
- Selection structure
- Iteration structure



Flow Chart

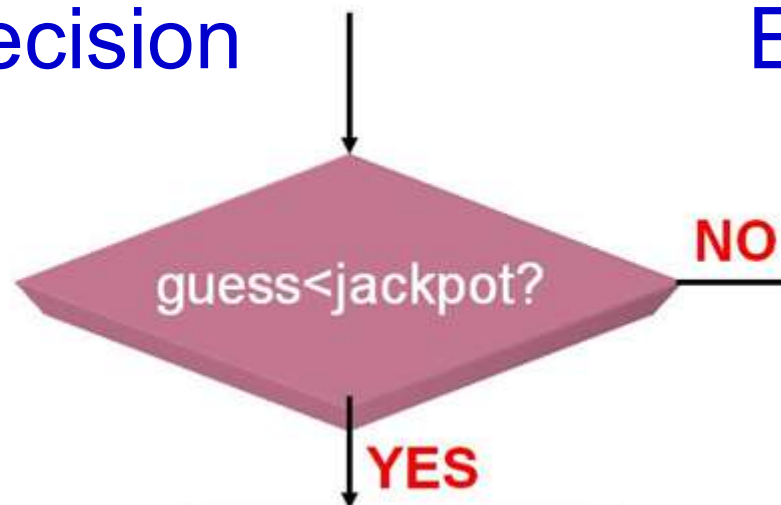
Begin



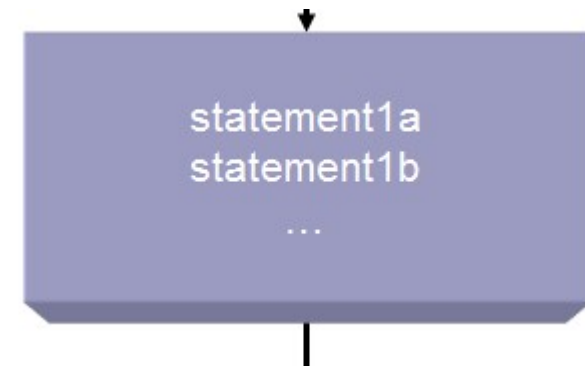
End



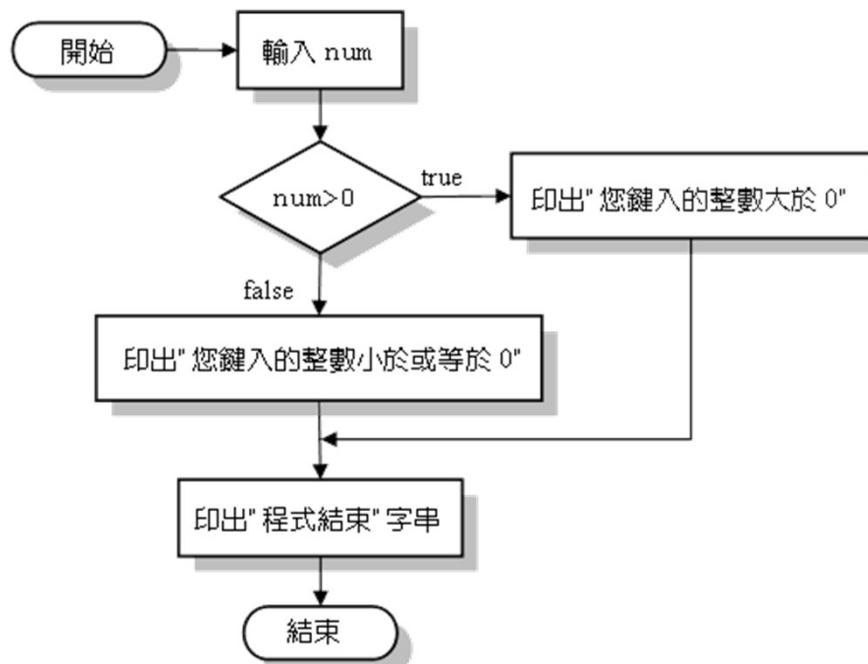
Decision



Events, codes...



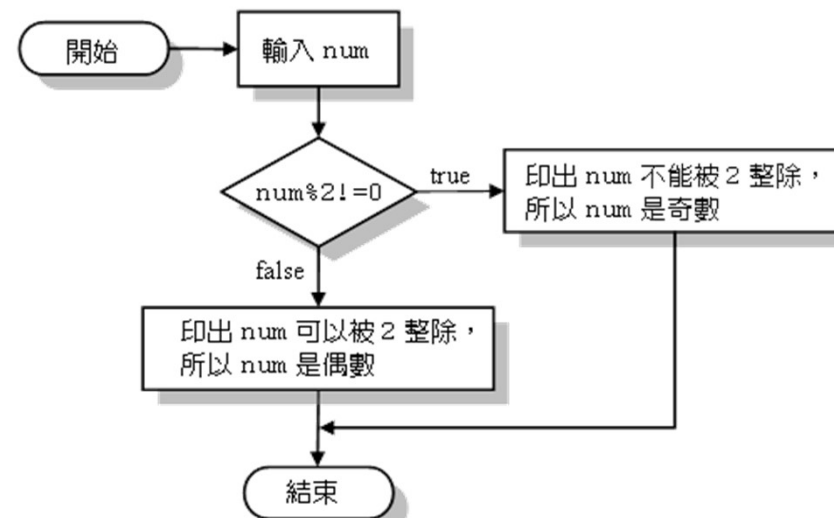
A Practice - if..else..



請輸入一個整數: **-106**

您鍵入的整數小於或等於 0
程式結束

-----*/



請輸入一個整數: **34**

34 可以被 2 整除，所以 34 是偶數

-----*/