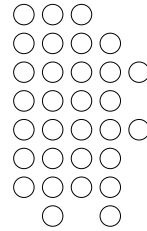


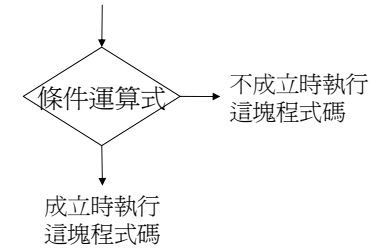
第六章 選擇性敘述

if 敘述
條件運算子的應用
多重選擇—switch 敘述
無條件跳離的 goto 敘述



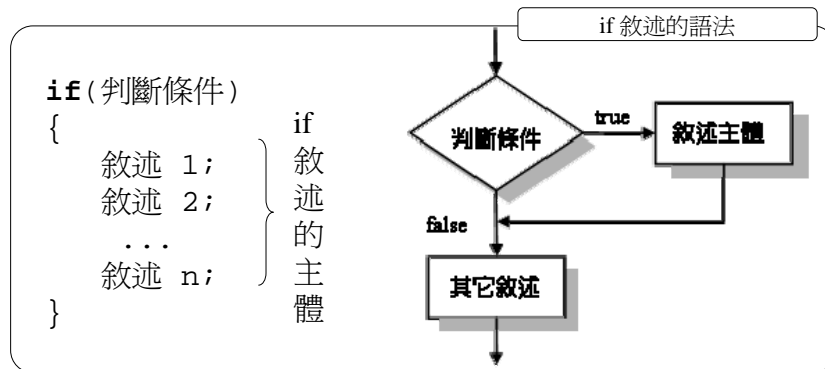
選擇性敘述

- 選擇性敘述可依據一個變數或一個條件運算式，讓 CPU 決定執行的流程
- 常用的選擇性敘述：
 - if
 - if-else
 - switch



if 敘述的使用 (1/3)

- 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"); /* prog6_1 OUTPUT--
14
15     system("pause");
16     return 0;
17 }
```

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

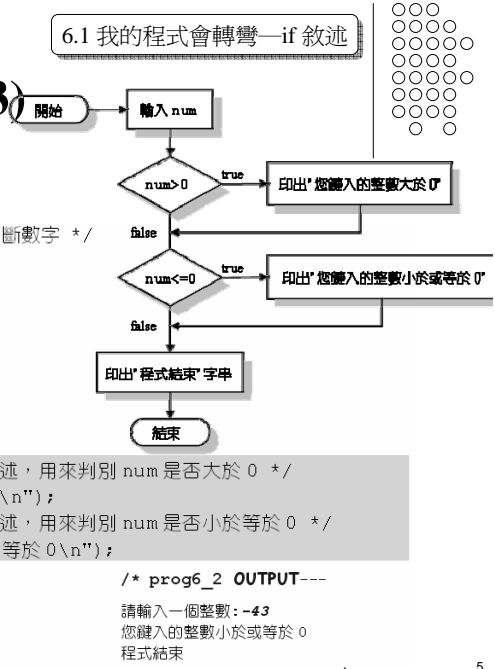
6.1 我的程式會轉彎—if 敘述

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 敘述來判斷數字 */
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 }
```

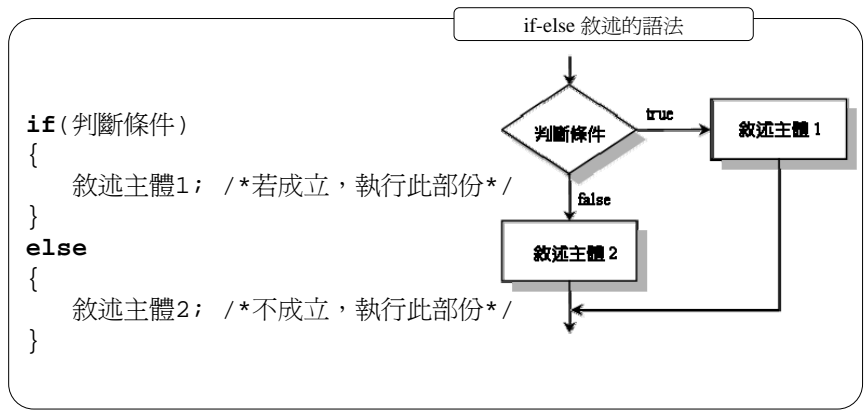
```
/* prog6_2 OUTPUT---
請輸入一個整數: -43
您鍵入的整數小於或等於 0
程式結束
-----*/
```



6.2 另外的選擇—if-else 敘述

if-else 敘述

- if-else 敘述可用於：成立時則執行 1，否則執行 2



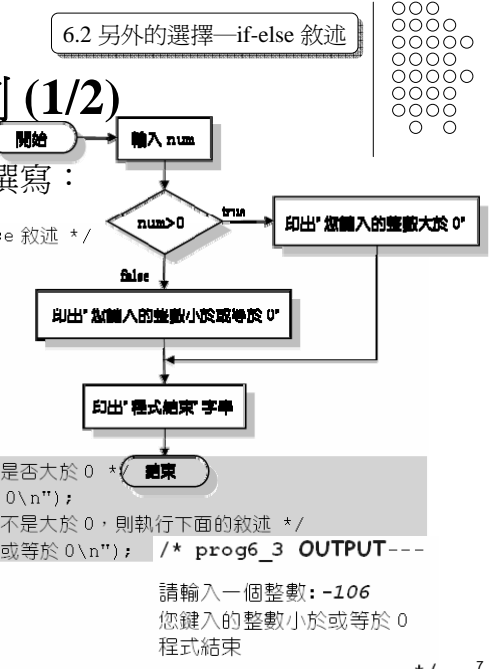
6.2 另外的選擇—if-else 敘述

if-else 敘述的範例 (1/2)

- 將 prog6_2 改以 if-else 撰寫：

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

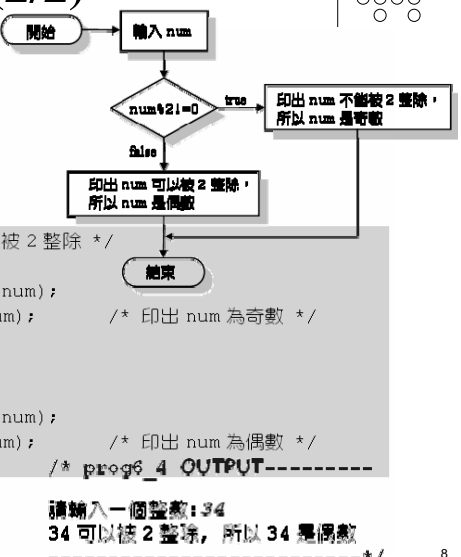


6.2 另外的選擇—if-else 敘述

if-else 敘述的範例 (2/2)

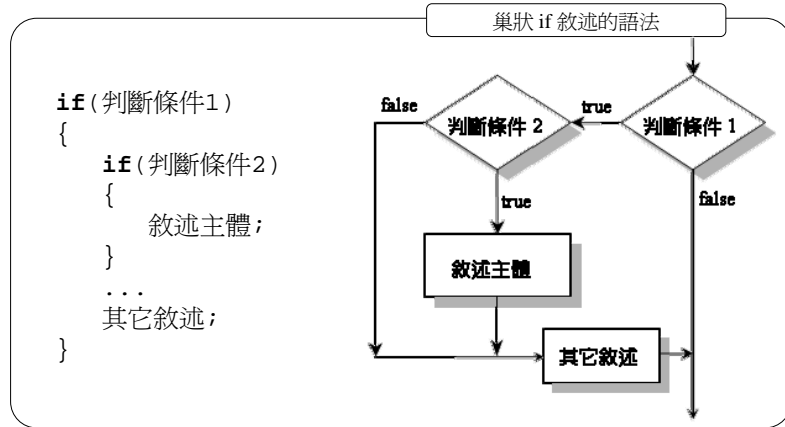
```
01 /* prog6_4, if-else 敘述的練習 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num;
07     printf("請輸入一個整數:");
08     scanf("%d", &num);
09     if (num%2!=0) /* 如果 num 不能被 2 整除 */
10     {
11         printf("%d 不能被 2 整除, ", num);
12         printf("所以 %d 是奇數\n", num); /* 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
-----*/
```



巢狀 if 敘述

- if 裡面還有其它的 if 敘述，則稱為巢狀 if 敘述

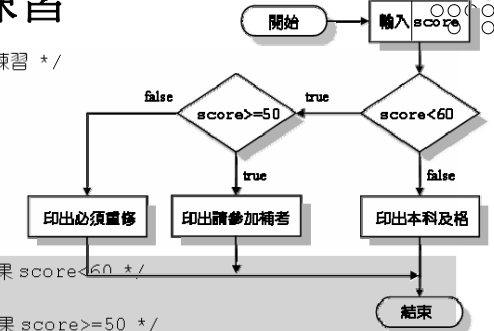


9

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 score<60 */
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 }
  
```



```

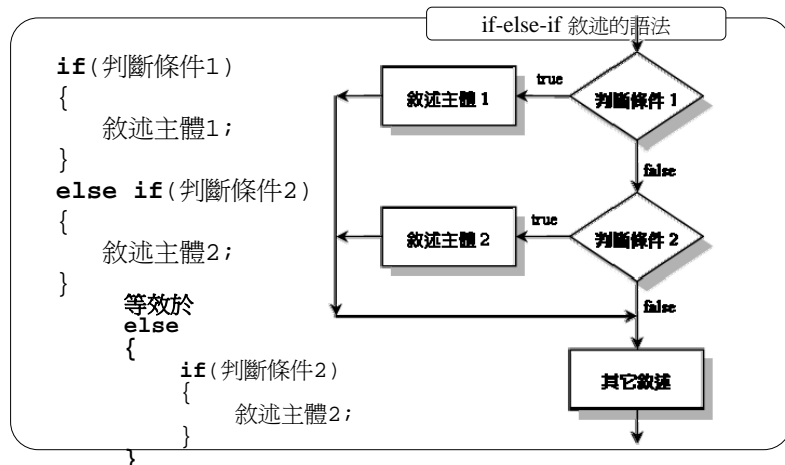
/* prog6_5 OUTPUT---
請輸入成績:52
請參加補考
-----*/
  
```

10

使用 if-else-if 敘述

多重選擇

- if-else-if：當 if 判斷不成立，必須進行其它判斷時

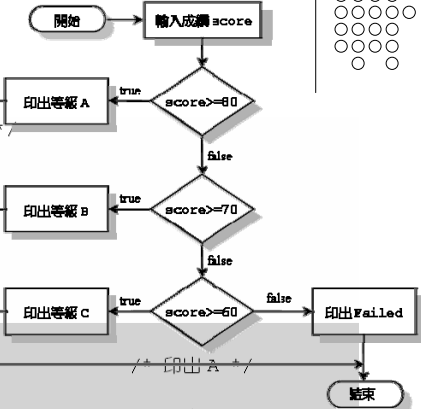


11

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); /* 印出 A */
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"); /* prog6_6 OUTPUT---
17     system("pause");
18     return 0;
19 }
  
```



```

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

12

if 與 else 的配對問題 (1/2)

- else 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     {
12         if (num<=10)
13             printf("數字介於 0 到 10 之間\n");
14         else
15             printf("數字大於 10\n");
16     }
17     system("pause");
18     return 0;
19 }
```

/* 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

13

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
-----*/

14

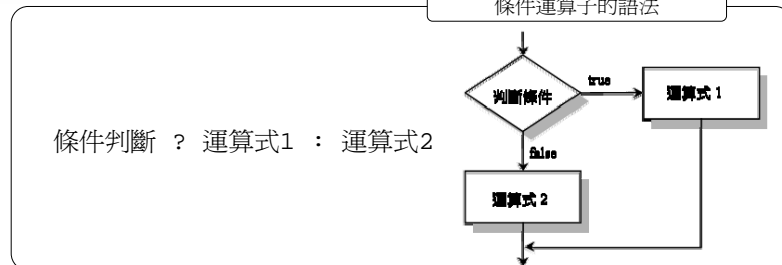
條件運算子 (1/2)

- 條件運算子可代替簡單的 if-else 敘述

表 6.3.1 條件運算子

條件運算子	意義
?:	根據條件的成立與否，來決定結果為?:後的運算式

條件運算子的語法



15

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;
```

16

條件運算子的範例

- 利用條件運算子判斷兩個數中較大的數：

```

01 /* prog6_9, 條件運算子的練習 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int num1,num2,larger;
07     printf("請輸入兩個整數:");
08     scanf("%d %d",&num1,&num2);
09
10     num1>num2 ? (larger=num1) : (larger=num2); /* 條件運算子 */
11     printf("%d 數值較大\n",larger);
12
13     system("pause");
14     return 0;
15 }

```

larger = num1>num2 ? num1 : num2;

```

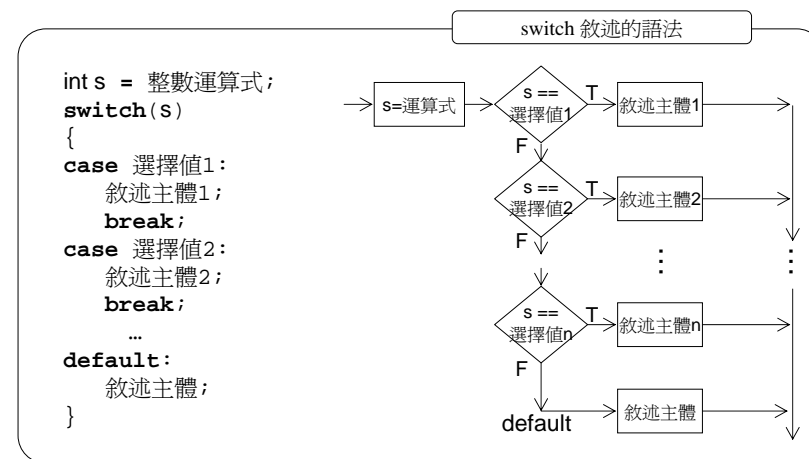
/* prog6_9 OUTPUT---
請輸入兩個整數:33 76
76 數值較大
-----*/

```

17

switch 敘述

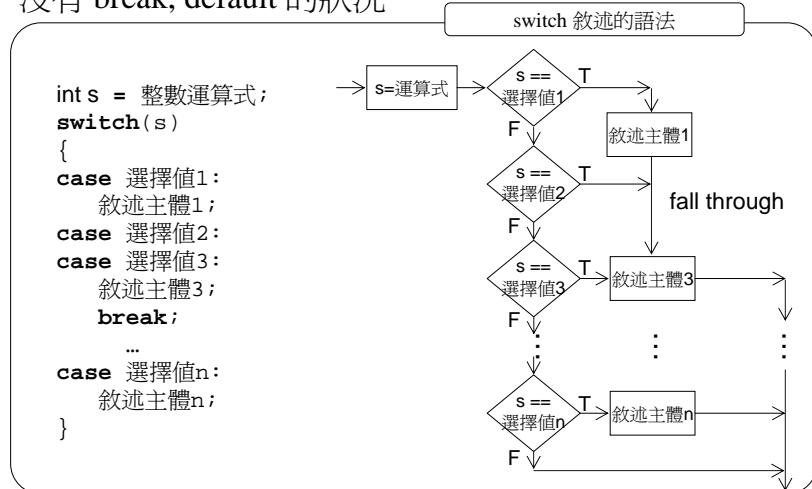
- switch 敘述可依據整數運算式的結果進行多重選擇



18

switch 敘述 (cont'd)

- 沒有 break, default 的狀況



19

switch 敘述的範例 (1/2)

- 依據選擇值進行四則運算：



20



switch 敘述的範例 (1/2)

```

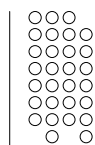
16     case '-':
17         printf("%d-%d=%d\n", a, b, a-b);      /* 印出 a-b */
18         break;
19     case '*':
20         printf("%d*%d=%d\n", a, b, a*b);      /* 印出 a*b */
21         break;
22     case '/':
23         printf("%d/%d=%.3f\n", a, b, (float)a/b); /* 印出 a/b */
24         break;
25     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
-----*/

21



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;

```

22



將不同的選擇值並列 (2/2)

```

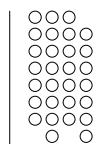
20     case 'c': /* 輸入 c 或 C 時印出 Be study hard! */
21     case 'C':
22         printf("Be study hard!\n");
23         break;
24     default: /* 輸入其他字元時印出 Failed! */
25         printf("Failed!\n");
26     }
27     system("pause");
28     return 0;
29 }

```

/* prog6_11 OUTPUT---

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

23



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");

```

24

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");
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--

```

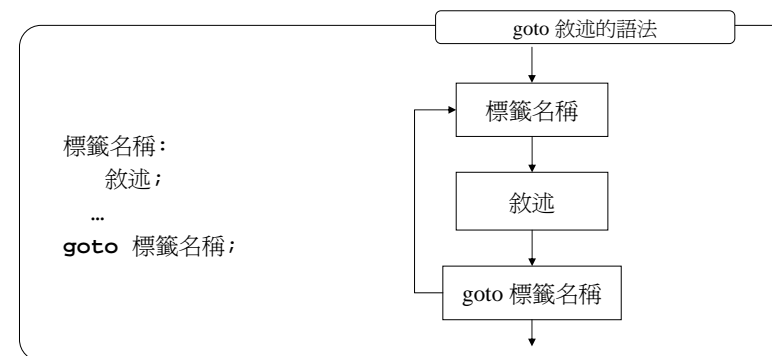
Input grade:b
Good!
Be study hard!
Failed!
-----*/

```

25

使用 goto 敘述

- goto 敘述可強制執行流程跳到指定的地方執行，常常配合條件敘述使用，由於 goto 的使用太過自由，使用時很容易讓程式的複雜度快速增加，非萬不得已請不要使用



26

goto 敘述的範例

```

01 /* prog6_13, 使用 goto 敘述 */
02 #include <stdio.h>
03 #include <stdlib.h>
04 int main(void)
05 {
06     int i=0,sum=0;
07     start: /* start 標籤 */
08         i++;
09         sum+=i;
10         printf("%d",i);
11         if (i<10)
12         {
13             printf("+"); /* 印出+, 並回到 start 標籤內的敘述 */
14             goto start;
15         }
16         printf("%d\n",sum); /* 印出 sum 的值 */
17     system("pause");
18     return 0;
19 }

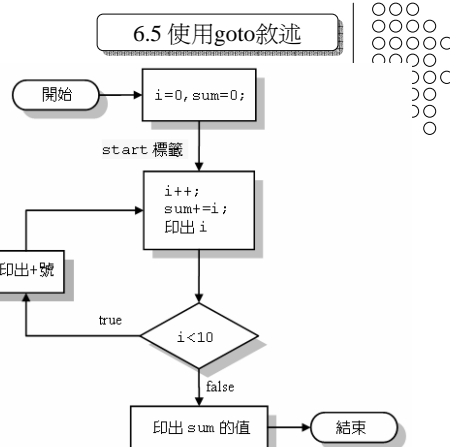
```

/* prog6_13 OUTPUT----

```

1+2+3+4+5+6+7+8+9+10=55
-----*/

```



27

注意事項

- goto 敘述直接換成低階的 CPU 指令 jmp
- goto 敘述使得流程太有彈性，破壞程式的結構性，使得程式不容易閱讀，不容易修改與維護
- 每一個 goto 敘述皆可以使用其它的語法替代
- 避免使用 goto 敘述

28