第五章講授重點

講授重點

1. 函式之結構
2. 現有數學函式庫
3. 傳值或傳參考呼叫
4. 亂數產生

1. 函式之結構

- 發展和維護大型程式的最好方法，便是以一些較小的單元或模組 (module) 來建構整個程式，這些小單元要比整個大程式好管理多了。
- C裡面的模組稱爲函式 (function)。
- C程式的撰寫，通常是將程式設計師所寫的新函式，與位於C標準函式庫 (C standard library) 中事先寫好的函式結合起來構成一個程式。

1. 函式之結構(續)

- C標準函式庫提供了包羅萬象的函式，包括了常用的數學運算、字串處理、字元處理、輸入／輸出，以及許多其他有用的功能。
- printf、scanf和pow函式都是標準函式庫的函式。

1. 函式之結構(續)

```
1  /* Fig. 5.3: fig05_03.c
2     Creating and using a programmer-defined function */
3  #include <stdio.h>
4
5  int square( int y ); /* function prototype */
6
7  /* function main begins program execution */
8  int main( void )
9  {
10    int x; /* counter */
11
12    /* loop 10 times and calculate and output square of x each time */
13    for ( x = 1; x <= 10; x++ ) {
14       printf( "%d  ", square( x ) ); /* function call */
15    } /* end for */
16
17    printf( "\n" );
18    return 0; /* indicates successful termination */
19 } /* end main */
20
```

圖 5.3　使用程式設計師自訂函式

1. 函式之結構(續)

```
21  /* square function definition returns square of parameter */
22  int square( int y ) /* y is a copy of argument to function */
23  {
24     return y * y; /* returns square of y as an int */
25  } /* end function square */
```

```
1  4  9  16  25  36  49  64  81  100
```

圖 5.3　使用程式設計師自訂函式

```
# include <……>
函式宣告 (例如  int square ( int y) ;  )

main()
{
    int x, y=5;
    x = square(y);
}

函式(例如  int square ( int  a)  )
{
   int b;
   b = a*a;
   return b ;
}
```

例題

wk1_1.c

```c
/* Example 1 in Week 1*/
#include <stdio.h>

int main()
{
    int A, B, S;

    printf("Enter 1st integer\n");
    scanf("%d", &A); /*鍵入第一個數字*/
    printf("Enter 2nd integer\n");
    scanf("%d", &B); /*鍵入第二個數字*/
    S = A+B;
    printf("Sum is %d\n", S);
    return 0;

}
```

例 題(續)

[*] wk1_2.c

```c
/* Example 1_2 for integer*/
#include <stdio.h>

int sum(int a, int b);

main()
{
    int A, B, S;

    printf("Enter 1st integer\n");
    scanf("%d", &A); /*鍵入第一個數字*/
    printf("Enter 2nd integer\n");
    scanf("%d", &B); /*鍵入第二個數字*/
    S = sum(A, B);
    printf("Sum is %d\n", S);

}

int sum(int a, int b)
{
    int c;
    c = a+b;
    return c;
}

```

2. 現有數學函式庫

- # include <stdlib.h>
- # include <math.h>
- 圖5.2整理一些C的常用數學函式庫函式。
- 在此圖中，變數x和y的型別都是double。.

2. 現有數學函式庫

函式	說明	範例
`sqrt( x )`	x的平方根	`sqrt( 900.0 )` is `30.0` `sqrt( 9.0 )` is `3.0`
`exp( x )`	指數函式 e^x	`exp( 1.0 )` is `2.718282` `exp( 2.0 )` is `7.389056`
`log( x )`	x的自然對數 (底爲 e)	`log( 2.718282 )` is `1.0` `log( 7.389056 )` is `2.0`

`log10( x )`	x的對數 (底爲 10)	`log10( 1.0 )` is `0.0`
		`log10( 10.0 )` is `1.0`
		`log10( 100.0 )` is `2.0`

`fabs( x )`	x的絕對值	`fabs( 13.5 )` is `13.5`
		`fabs( 0.0 )` is `0.0`
		`fabs( -13.5 )` is `13.5`

`ceil( x )`	不小於 x的最小整數	`ceil( 9.2 )` is `10.0` `ceil( -9.8 )` is `-9.0`
`floor( x )`	不大於 x的最大整數	`floor( 9.2 )` is `9.0` `floor( -9.8 )` is `-10.0`

圖 5.2　常用的數學函式庫函式

pow(x, y)	x的y次方 (x^y)	pow(2, 7) is 128.0 pow(9, .5) is 3.0
fmod(x, y)	x/y的浮點餘數	fmod(13.657, 2.333) is 1.992
sin(x)	x的正弦值 (x的單位爲弧度)	sin(0.0) is 0.0
cos(x)	x的餘弦值 (x的單位爲弧度)	cos(0.0) is 1.0
tan(x)	x的正切值 (x的單位爲弧度)	tan(0.0) is 0.0

圖 5.2　常用的數學函式庫函式

5.6 函式原型

- 函式原型的另一項功能是**強制的引數型別轉換(coercion of arguments)**，亦即強迫引數變成恰當的型別。
- 例如，雖然數學函式sqrt其函式原型（位於math.h檔）規定了一個double引數，但我們可以用一個整數引數來呼叫它，而且此函式也能正確地運作。
- 敘述式

```
printf( "%.3f\n", sqrt( 4 ) );
```

可以正確地執行sqrt(4)，並印出值2.000。

5.6 函式原型

- 編譯器會在引數值傳給sqrt之前，將整數值4轉換成double值4.0。
- 總而言之，當引數的值沒有準確地對應到函式原型中的參數型別時，這些引數值將會在函式呼叫之前，先轉換成正確的型別。
- 這種轉換如果沒有遵守C的提升規則(promotion rules)的話，可能導致不正確的結果產生。
- 提升規則規定了在不遺漏資料的前提下，某一型別如何才能轉換成另一型別。

5.6 函式原型

- 在sqrt的例子中，int可以自動轉換成double而不會遺漏資料。
- 不過，若是double轉換成int的話，double值的小數部分將會捨去。
- 將大的整數型別轉換成較小的整數型別（如long變short），也可能造成數值改變。
- 提升規則會自動應用到含有兩種（或更多）資料型別之數值的運算式，也稱爲混合型別運算式 (mixed-type expressions)。

5.6 函式原型

- 混合型別運算式中，每一個值的型別會自動提升爲此運算式中的最高型別（事實上是爲每個值製造一個暫時的值，用在運算式的計算上－－原來的值並沒有跟著改變）。
- 圖5.5由最高型別至最低型別列出了所有的資料型別，並列出每一個型別的printf和scanf轉換指定。

資料型別	printf 的轉換指定詞	scanf 的轉換指定詞
long double	%Lf	%Lf
double	%f	%lf
float	%f	%f
unsigned long int	%lu	%lu
long int	%ld	%ld
unsigned int	%u	%u
int	%d	%d
unsigned short	%hu	%hu
short	%hd	%hd
char	%c	%c

圖 5.5　資料型別的提升階層

作業

- 請將第3章(ch03.rar)，t2_1.c 改成函式型式執行
- 函式定名爲階乘 factorial(int y)

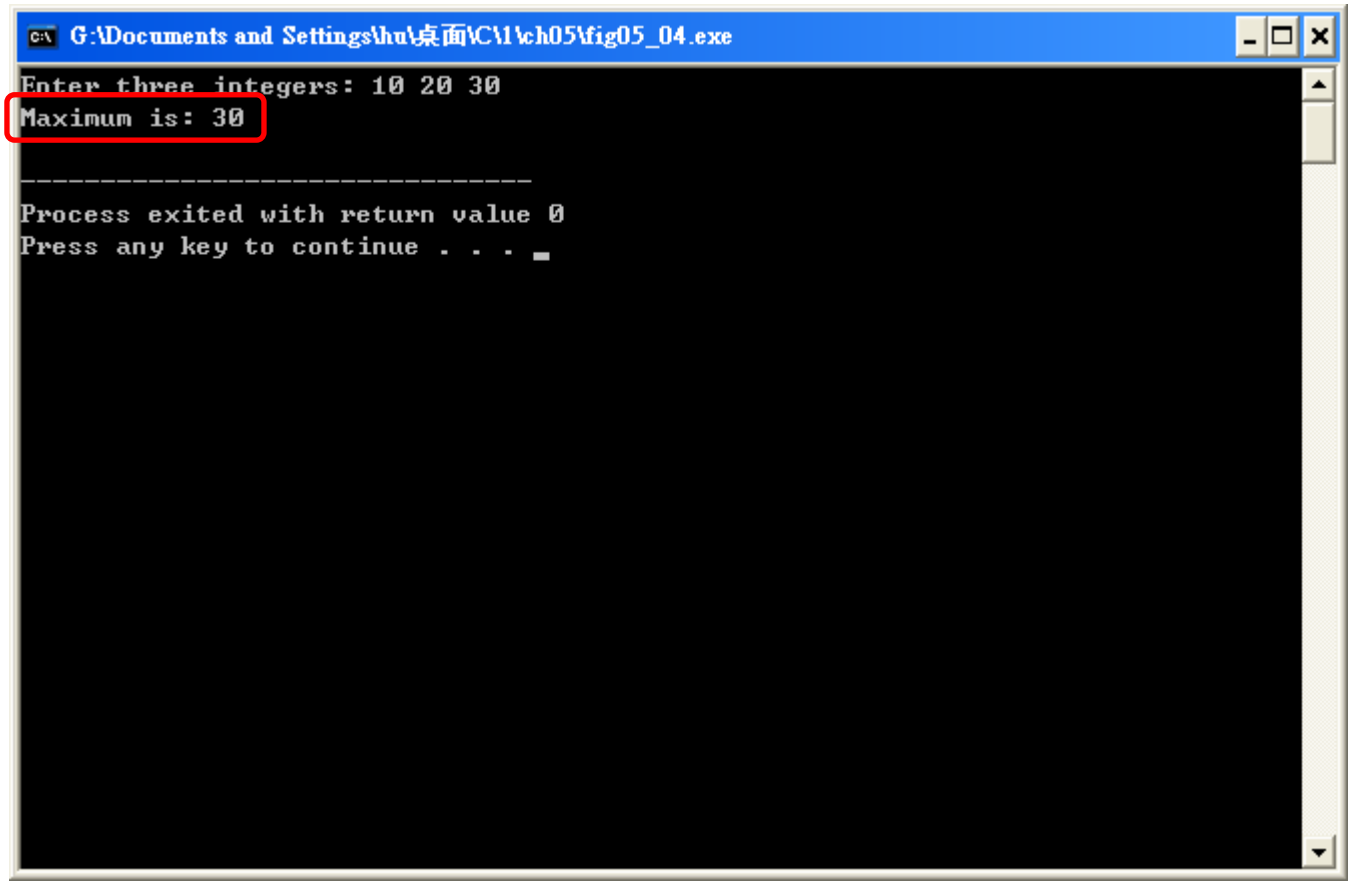
3. 傳值或傳參考呼叫

- 傳值呼叫 (call by value) 和傳參考呼叫 (call by reference) 是大多數的程式語言用來引用函式的兩種方式。
- 當以傳值呼叫來傳遞引數時，此引數值的一份*複製*將會傳給受呼叫的函式。
- 對此複製所做的修改並不會影響到呼叫者原來變數的值。

fig05_04.c

```c
/* fig05_04.c
   Finding the maximum of three integers */
#include <stdio.h>

int maximum( int x, int y, int z ); /* function prototype */

int main( void )
{
   int number1; /* first integer */
   int number2; /* second integer */
   int number3; /* third integer */

   printf( "Enter three integers: " );
   scanf( "%d%d%d", &number1, &number2, &number3 );

   /* number1, number2 and number3 are arguments
      to the maximum function call */
   printf( "Maximum is: %d\n", maximum( number1, number2, number3 ) );

   return 0;
}

int maximum( int x, int y, int z )
{
   int max = x; /* assume x is largest */

   if ( y > max ) { /* if y is larger than max, assign y to max */
      max = y;
   } /* end if */

   if ( z > max ) { /* if z is larger than max, assign z to max */
      max = z;
   } /* end if */

   return max;
}
```

傳值呼叫

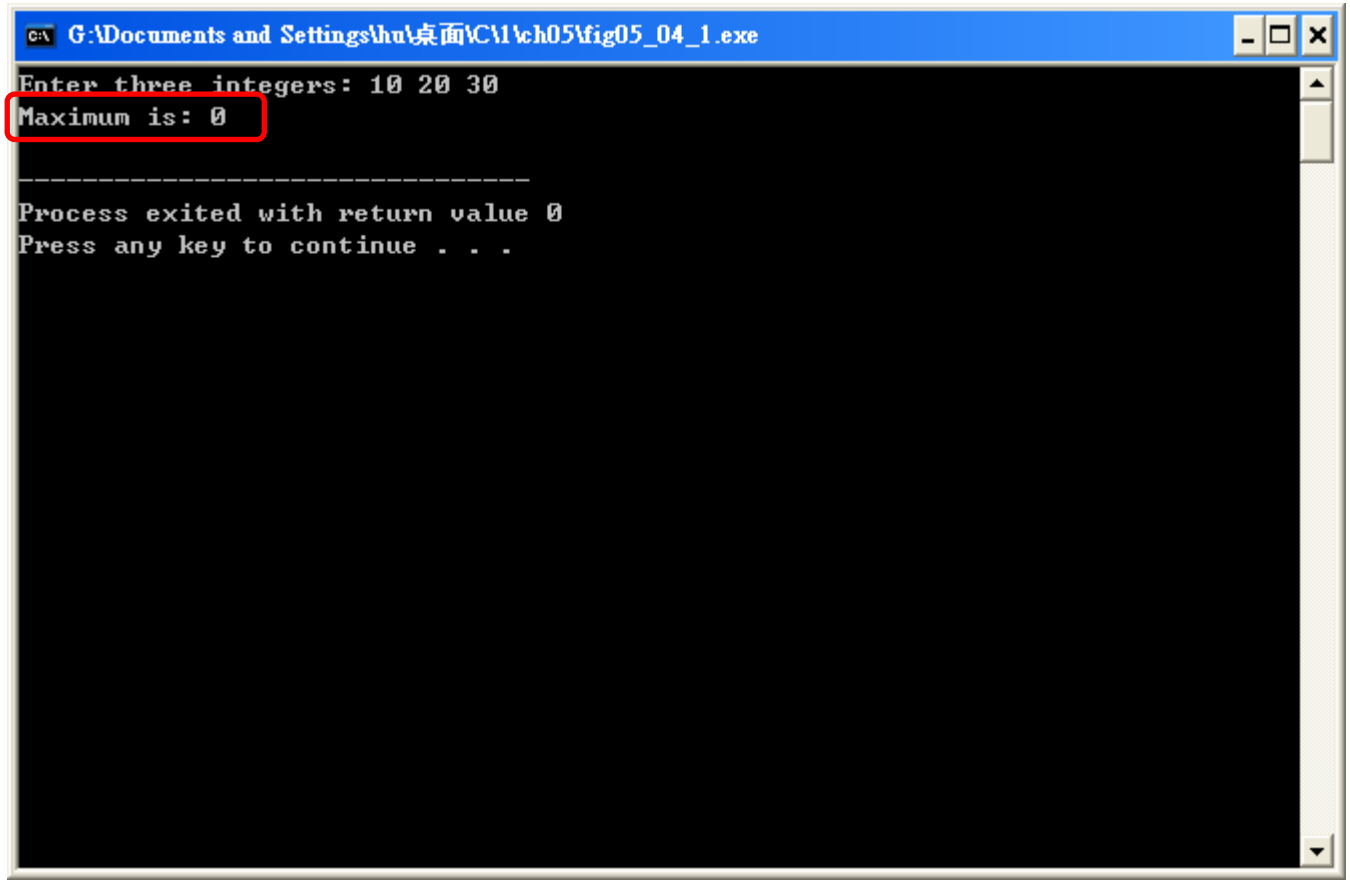

G:\Documents and Settings\hu\桌面\C\1\ch05\fig05_04.exe
Enter three integers: 10 20 30
Maximum is: 30

Process exited with return value 0
Press any key to continue . . .

fig05_04_1.cpp

```
1  /* fig05_04_1.c
2     Finding the maximum of three integers */
3  #include <stdio.h>
4
5  int maximum( int x, int y, int z, int maxx ); /* function prototype */
6
7  int main( void )
8  {
9     int number1; /* first integer */
10    int number2; /* second integer */
11    int number3; /* third integer */
12    int max = 0;
13
14    printf( "Enter three integers: " );
15    scanf( "%d%d%d", &number1, &number2, &number3 );
16
17    /* number1, number2 and number3 are arguments
18       to the maximum function call */
19    maximum( number1, number2, number3, max );
20    printf( "Maximum is: %d\n", max );
21
22    return 0;
23 }
24
25
26 int maximum( int x, int y, int z, int maxx )
27 {
28    maxx = x; /* assume x is largest */
29
30    if ( y > maxx ) { /* if y is larger than max, assign y to max */
31       maxx = y;
32    } /* end if */
33
34    if ( z > maxx ) { /* if z is larger than max, assign z to max */
35       maxx = z;
36    } /* end if */
37
38    return maxx;
39 }
```

傳值呼叫


G:\Documents and Settings\hu\桌面\C\1\ch05\fig05_04_1.exe
Enter three integers: 10 20 30
Maximum is: 0

Process exited with return value 0
Press any key to continue . . .

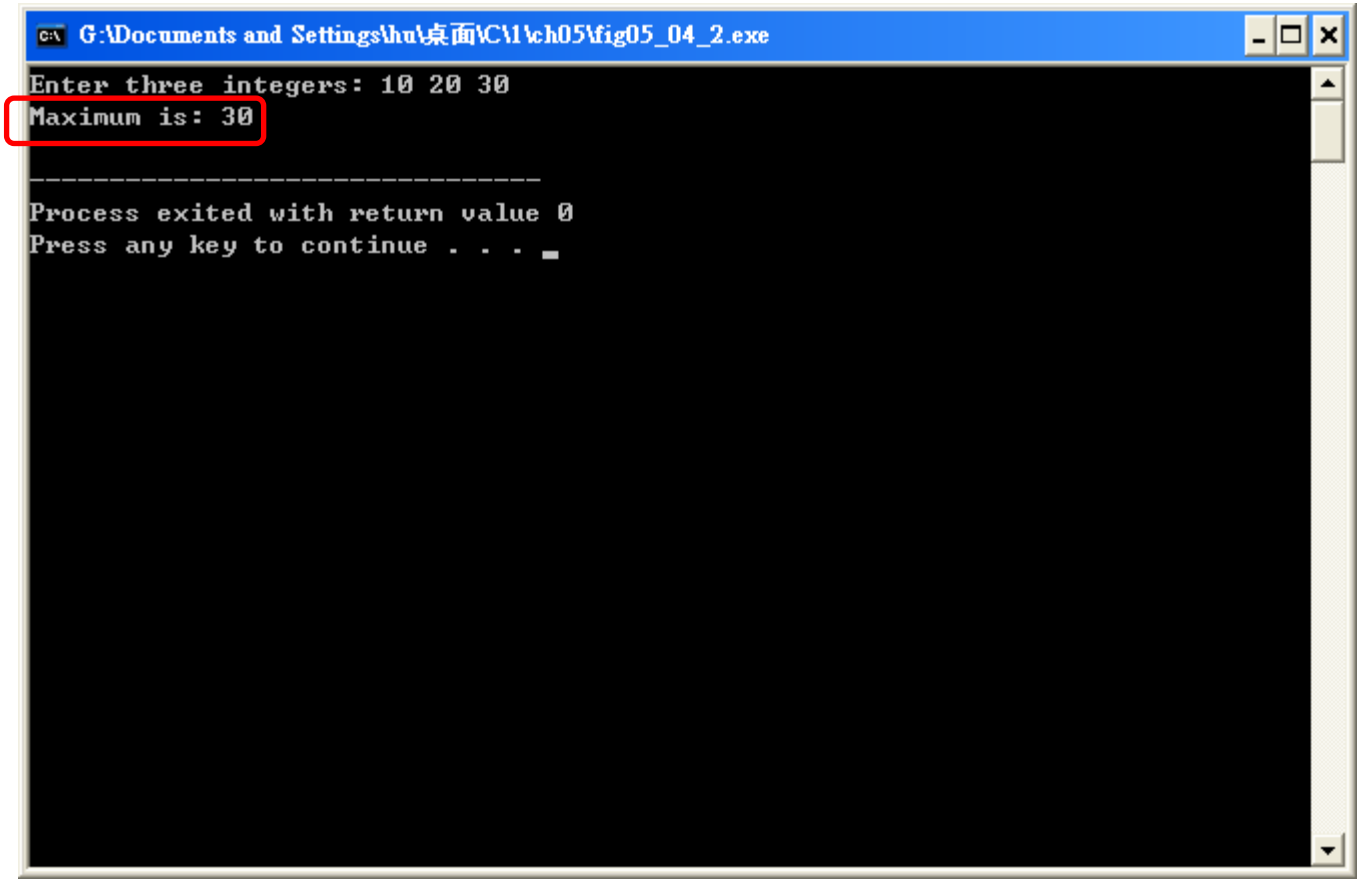
3. 傳值或傳參考呼叫(續)

- 當以傳參考來傳遞引數時，呼叫者允許受呼叫函式修改原來的變數值。
- 當受呼叫函式不需修改呼叫者的原始變數時，應該使用傳值呼叫。

fig05_04_2.cpp

```
1  /* fig05_04.c
2     Finding the maximum of three integers */
3  #include <stdio.h>
4
5  int maximum( int x, int y, int z, int *maxx ); /* function prototype
6
7  int main( void )
8  {
9     int number1; /* first integer */
10    int number2; /* second integer */
11    int number3; /* third integer */
12    int max = 0;
13
14    printf( "Enter three integers: " );
15    scanf( "%d%d%d", &number1, &number2, &number3 );
16
17    /* number1, number2 and number3 are arguments
18       to the maximum function call */
19    maximum( number1, number2, number3, &max );
20    printf( "Maximum is: %d\n", max );
21
22    return 0;
23 }
24
25
26 int maximum( int x, int y, int z, int *maxx )
27 {
28    *maxx =x; /* assume x is largest */
29
30    if ( y > *maxx ) { /* if y is larger than max, assign y to max */
31       *maxx = y;
32    } /* end if */
33
34    if ( z > *maxx ) { /* if z is larger than max, assign z to max */
35       *maxx = z;
36    } /* end if */
37
38    return *maxx;
39 }
40
```

傳參考呼叫


G:\Documents and Settings\hu\桌面\C\1\ch05\fig05_04_2.exe
Enter three integers: 10 20 30
Maximum is: 30

Process exited with return value 0
Press any key to continue . . .

4. 亂數產生

- 在電腦的應用程式裡，標頭檔stdlib.h中的rand函式來模擬機會的運行。

```
i = rand();
```

- 其中rand函式會產生一個介於0和RAND_MAX (定義在<stdlib.h>標頭檔中的符號常數) 之間的整數。
- RAND_MAX的值至少需爲32767，此也是兩個位元組 (即16 bit) 所能表示的最大整數值。

4. 亂數產生(續)

- 例如，一個模擬擲銅板動作的程式，0來代表正面以1來代表反面。

```
I = rand() % 2
```

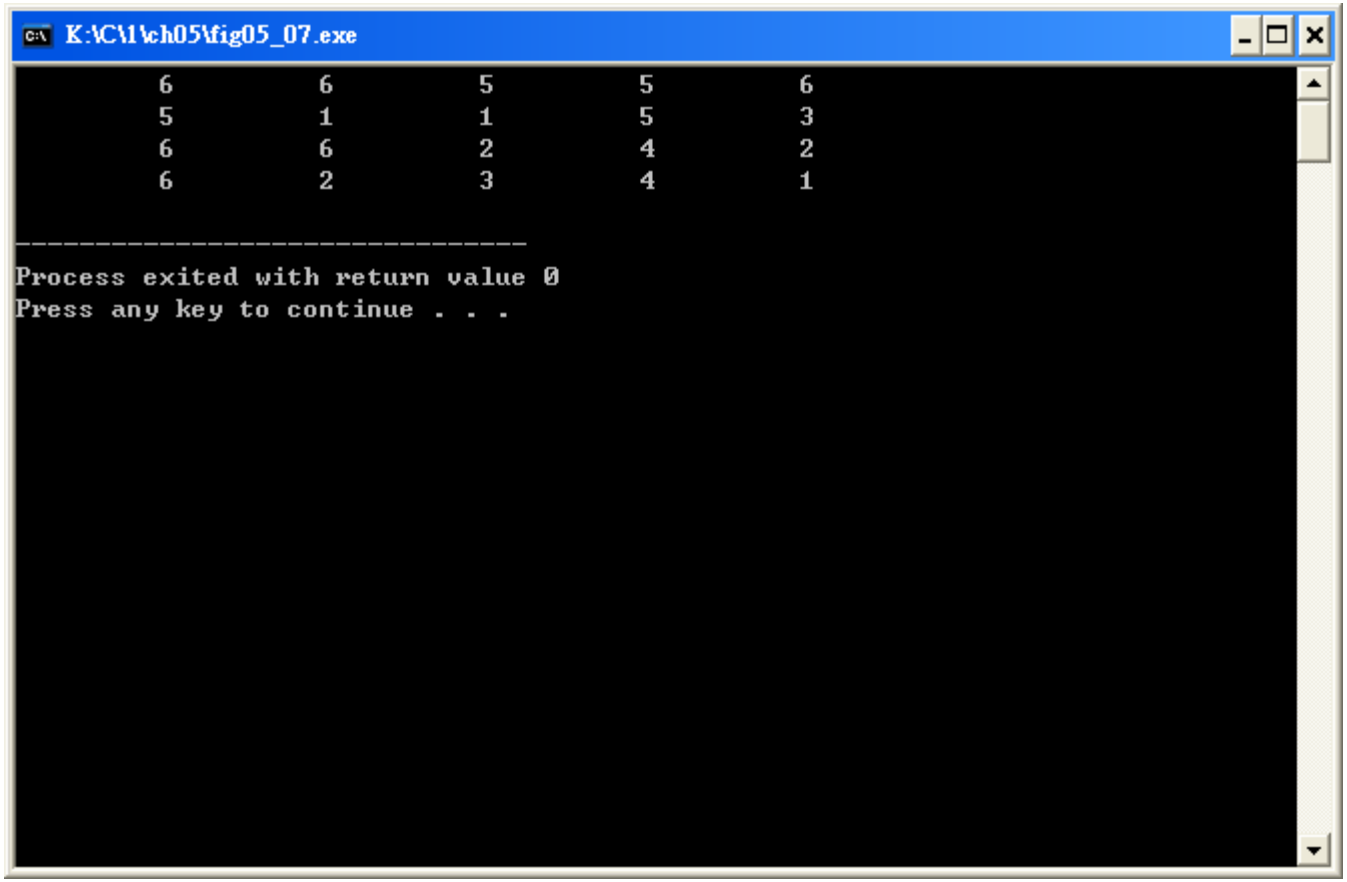
- 模擬擲骰子動作的程式，則需要1到6來代表骰子的6個面。

```
I = 1 + rand() % 6
```

4. 亂數產生(續)

```
1  /* Fig. 5.7: fig05_07.c
2     Shifted, scaled integers produced by 1 + rand() % 6 */
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  /* function main begins program execution */
7  int main( void )
8  {
9     int i; /* counter */
10
11    /* loop 20 times */
12    for ( i = 1; i <= 20; i++ ) {
13
14       /* pick random number from 1 to 6 and output it */
15       printf( "%10d", 1 + ( rand() % 6 ) );
16
17       /* if counter is divisible by 5, begin new line of output */
18       if ( i % 5 == 0 ) {
19          printf( "\n" );
20       } /* end if */
21    } /* end for */
22
23    return 0; /* indicates successful termination */
24 } /* end main */
```

圖 5.7　以 1+rand()%6 所產生的移位過且比例化過的整數


K:\C\1\ch05\fig05_07.exe
6 6 5 5 6
5 1 1 5 3
6 6 2 4 2
6 2 3 4 1
Process exited with return value 0
Press any key to continue . . .

4. 亂數產生(續)

- rand函式所產生的是虛擬亂數 (pseudo-random numbers)。
- 重複呼叫rand所產的數列，看起來也是隨機的。
- 只不過每次執行時，都會出現相同的數列。

4. 亂數產生(續)

- 隨機化 (randomizing)，可用標準函式庫函式 srand來進行。
- srand函式需要一個unsigned整數引數，它爲rand函式提供了種子 (seed)，讓rand能夠在每次執行時產生不同順序的亂數。
-

```c
/* Fig. 5.9: fig05_09.c
   Randomizing die-rolling program */
#include <stdlib.h>
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
   int i; /* counter */
   unsigned seed; /* number used to seed random number generator */

   printf( "Enter seed: " );
   scanf( "%u", &seed ); /* note %u for unsigned */

   srand( seed );  /* seed random number generator */

   /* loop 10 times */
   for ( i = 1; i <= 10; i++ ) {

      /* pick a random number from 1 to 6 and output it */
      printf( "%10d", 1 + ( rand() % 6 ) );

```

圖 5.9　將擲骰子程式隨機化

```
23         /* if counter is divisible by 5, begin a new line of output */
24         if ( i % 5 == 0 ) {
25            printf( "\n" );
26         } /* end if */
27      } /* end for */
28
29      return 0; /* indicates successful termination */
30   } /* end main */
```

```
Enter seed: 67
         6         1         4         6         2
         1         6         1         6         4
```

```
Enter seed: 867
         2         4         6         1         6
         1         1         3         6         2
```

```
Enter seed: 67
         6         1         4         6         2
         1         6         1         6         4
```

圖 5.9　將擲骰子程式隨機化

4. 亂數產生(續)

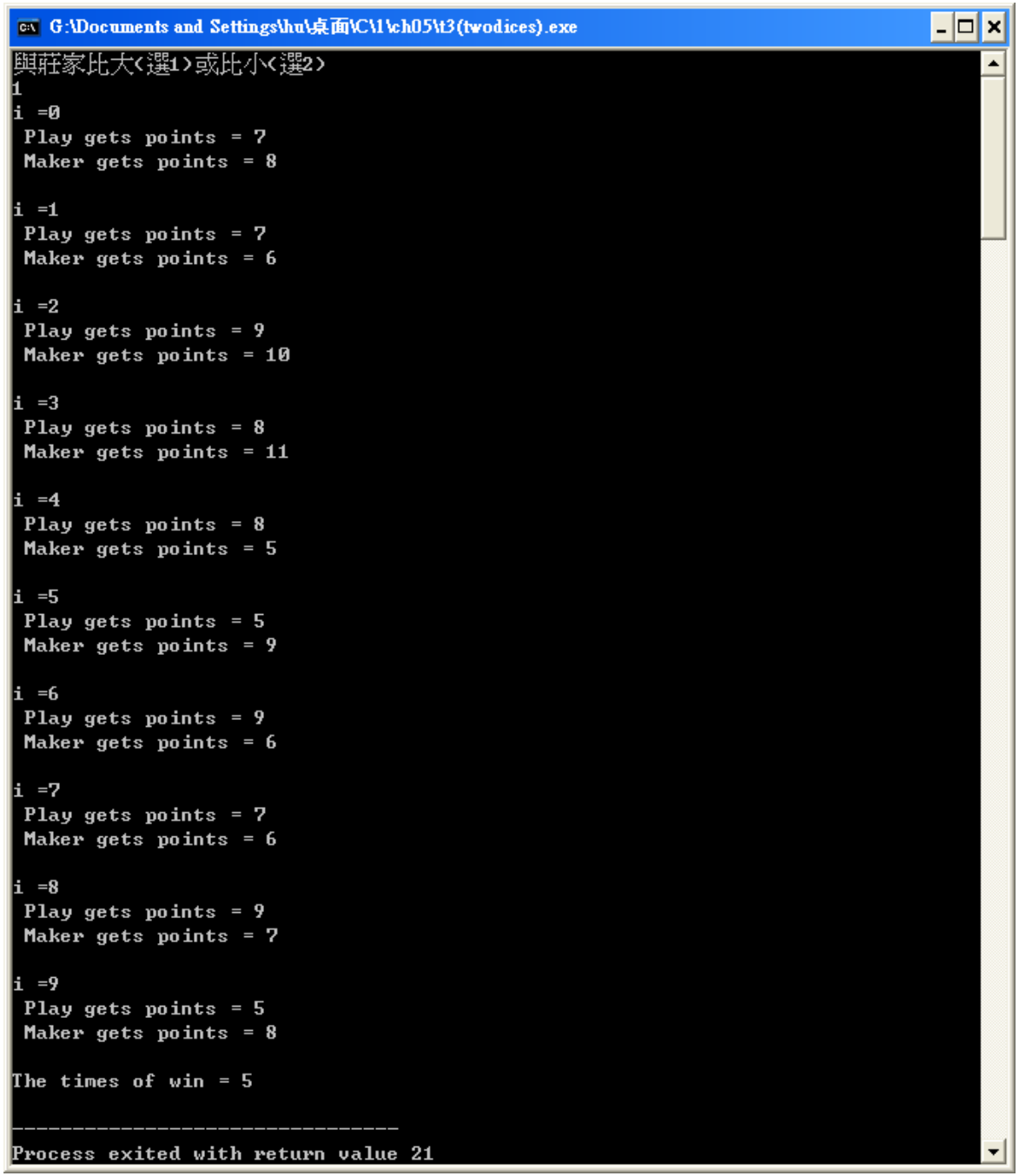
- 如果不需每次輸入一個seed，便能達到隨機化的效果，可以用如下的敘述式

```
srand( time( NULL ) );
```

- 此舉使電腦自動地讀取內部的時鐘，做爲seed的值。
- time的函式原型位於<time.h>裡。

t3(twodices).c

```c
//與莊家比大或比小，統計十次之結果
#include <stdio.h>
#include <stdlib.h>
#include <time.h> /* contains prototype for function time */

int rollDice( void ); /* function prototype */

int main( void )
{
   int d,sum1, sum2,i, times=0;
   printf( "與莊家比大(選1)或比小(選2)\n");
   scanf("%d", &d);
   srand( time( NULL ) );

   for(i=0; i<10; i++){
      printf("i =%d\n", i);

      sum1 = rollDice();
      printf( " Play gets points = %d\n", sum1 );

      sum2 = rollDice();
      printf( " Maker gets points = %d\n\n", sum2 );

      if( (d==1) && (sum1 > sum2)) times++;
      else if( (d==2) && (sum1 < sum2)) times++;
   }
   printf( "The times of win = %d\n", times );

}

/* roll dice, calculate sum and display results */
int rollDice( void )
{
   int s; /* first die */
   int die2,die1; /* second die */
   int workSum; /* sum of dice */

  // srand( time( NULL ) );
   die1 = 1 + ( rand() % 6 ); /* pick random die1 value */
   die2 = 1 + ( rand() % 6 );

   workSum = die1 + die2; /* sum die1 and die2 */
   return workSum;
}

```


G:\Documents and Settings\hu\桌面\C\1\ch05\t3(twodices).exe
與莊家比大(選1)或比小(選2)
1
i =0
Play gets points = 7
Maker gets points = 8
i =1
Play gets points = 7
Maker gets points = 6
i =2
Play gets points = 9
Maker gets points = 10
i =3
Play gets points = 8
Maker gets points = 11
i =4
Play gets points = 8
Maker gets points = 5
i =5
Play gets points = 5
Maker gets points = 9
i =6
Play gets points = 9
Maker gets points = 6
i =7
Play gets points = 7
Maker gets points = 6
i =8
Play gets points = 9
Maker gets points = 7
i =9
Play gets points = 5
Maker gets points = 8
The times of win = 5
Process exited with return value 21

作業

- 請做作業5-2 。