

## Chapter 1

# 基本程式設計導論

# 低階與高階程式語言

## ■ 計算機系統

硬體：CPU、記憶體、硬碟機 .....等

軟體：作業系統、遊戲、編譯器 .....等

## ■ 程式語言

機械語言：由 0、1 組成

低階語言：組合語言

高階語言：JAVA、C、C++.....等

其他語言：MATLAB、MAPLE.....等

# 原始碼與目標碼

```
int i = 5 ;
```

```
... ..
```

```
... ..
```



```
10001101010001101
```

```
01010101010101011
```

```
00101010111011101
```

原始碼

(人)

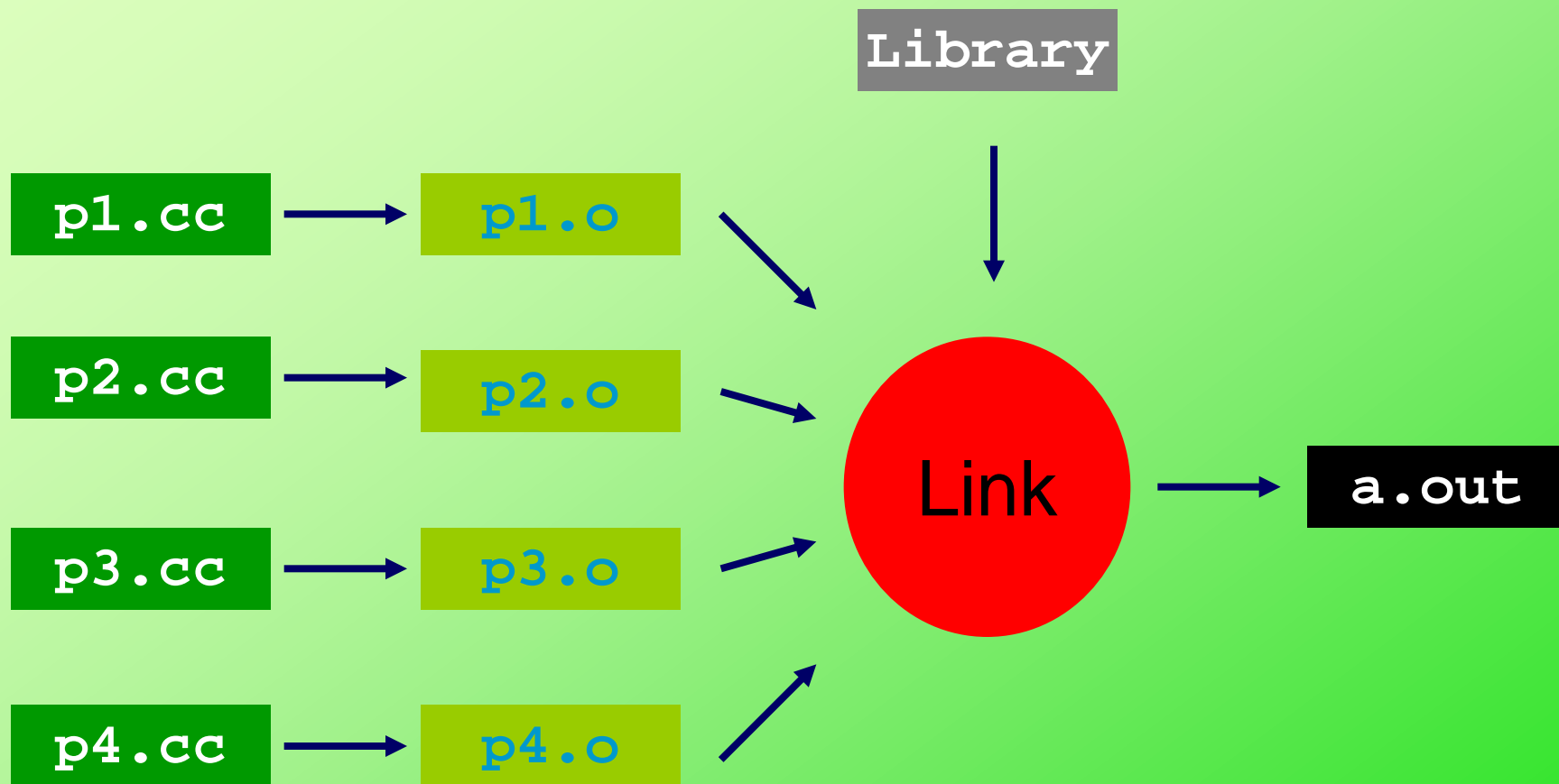
source code

目標碼

(機器)

object code

# 程式的分解與連結



link

# 位元、位元組與字元

- 位元 : 由 0 和 1 組成
- 位元組 : 由八個位元組成，最小的儲存單位
- 字元 : 以一個位元組代表，共有 256 種變化

eg. 'A' =  $1000001_2 = 65$   
'[' =  $1011011_2 = 91$   
'~' =  $1111110_2 = 126$

❖ C++ 字元型別 : `char`

bit , byte  
character

# 美國標準交換碼 (ASCII table)

## 美國標準交換碼：字元與數字對應表

32:	33: !	34: "	35: #	36: \$	37: %	38: &	39: '
40: (	41: )	42: *	43: +	44: ,	45: -	46: .	47: /
48: 0	49: 1	50: 2	51: 3	52: 4	53: 5	54: 6	55: 7
56: 8	57: 9	58: :	59: ;	60: <	61: =	62: >	63: ?
64: @	65: A	66: B	67: C	68: D	69: E	70: F	71: G
72: H	73: I	74: J	75: K	76: L	77: M	78: N	79: O
80: P	81: Q	82: R	83: S	84: T	85: U	86: V	87: W
88: X	89: Y	90: Z	91: [	92: \	93: ]	94: ^	95: _
96: `	97: a	98: b	99: c	100: d	101: e	102: f	103: g
104: h	105: i	106: j	107: k	108: l	109: m	110: n	111: o
112: p	113: q	114: r	115: s	116: t	117: u	118: v	119: w
120: x	121: y	122: z	123: {	124:	125: }	126: ~	

- 通常以 4 個位元組儲存

- 無號整數 (unsigned int)

數字範圍：  $0 \sim 2^{32} - 1$  相當於  $[0, 4294967295]$  約 43 億

- 有號整數 (int)

數字範圍：  $-2^{31} \sim 2^{31} - 1$  相當於  $[-2147483648, 2147483647]$

- 一個整數只有 9 位數的精確度

- 溢位：

當儲存空間不足時，空間內存的數值就不正確

$\Rightarrow 15! = 1307674368000 \rightarrow$  溢位

overflow

# 浮點數

1

## ■ 標準化浮點數表示：

$$x = \pm 0.d_1d_2\dots \times 2^m$$

$x$  : 浮點數

$\pm$  : 正負號

$d_1d_2 \dots$  : 底數 (normalized mantissa)

2 為基底， $m$  為指數， $d_1 = 1$ ，其它  $d_i$  屬於  $\{0,1\}$

## ■ 浮點數的種類：

單精確度：

`float` : 以 4 個位元組儲存數字，有 7 位有效數字

雙精確度：

`double` : 以 8 個位元組儲存數字，有 15 位有效數字

floating-point number



# 浮點數基底的轉換

## ■ 二進位數字 → 十進位數字

$$\begin{aligned} 0.01_2 &= 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 0 + 0.25 = 0.25 \end{aligned}$$

## ■ 十進位數字 → 二進位數字

$$\begin{aligned} 0.1 &= 1 \times 10^{-1} \\ &= 0 \times 2^{-1} + 0 \times 2^{-2} + \dots = 0.\overline{00011}_2 \end{aligned}$$

❖ 十進位的 0.1 無法用有限個二進位數字準確無誤地表示出來

# 截去誤差 (一)

## ■ 截去誤差

數字在基底轉換時因儲存位數的不足所造成的誤差

$$\begin{aligned}
 0.1 &= 0.0\overline{0011}_2 \sim 0.000110011_2 \\
 \text{截去誤差} &= 0.1 - 0.000110011_2 \\
 &= 0.1 + 0.1 + 0.1 + 0.1 + 0.1 \\
 &\sim 0.000110011_2 + \dots + 0.000110011_2 \neq 0.5
 \end{aligned}$$

$$\begin{aligned}
 0.25 &= 0.01_2 \\
 &= 0.25 + 0.25 + 0.25 + 0.25 \\
 &= 0.01_2 + 0.01_2 + 0.01_2 + 0.01_2 = 1
 \end{aligned}$$

■ 有截去誤差 : 0.1、0.2、0.4、0.37

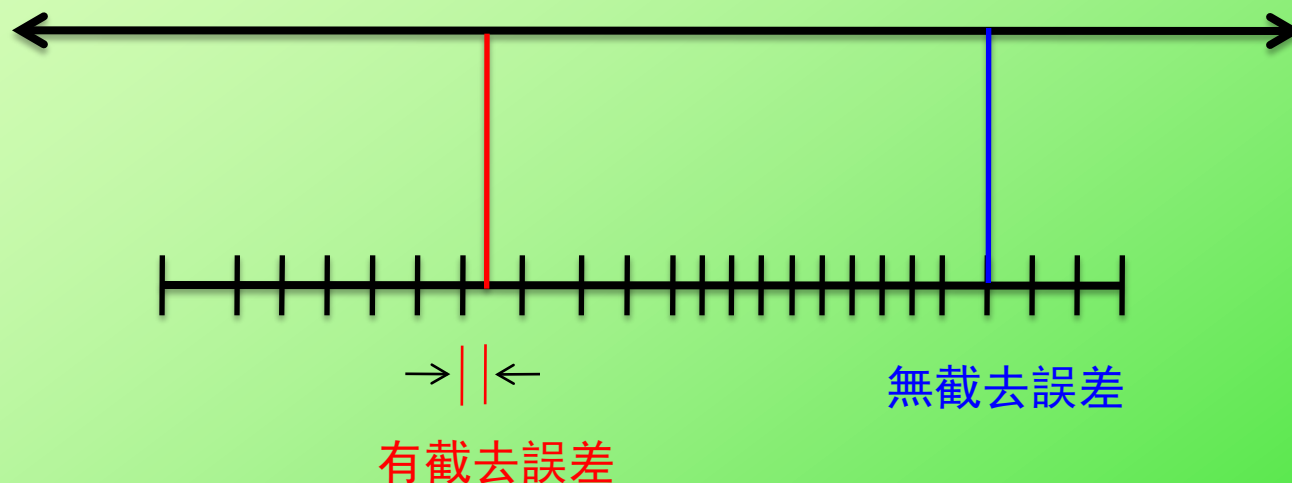
■ 無截去誤差 : 0.125、0.75、27

round-off error

# 截去誤差(二)

實數軸

計算機的  
數字



## 計算機的數字

- 為有限個有效位數
- 有最大數與最小數
- 都是離散的數字
- 總數字個數為有限個

# 計算機的數字

- 有效位數為有限個：

$$10^{10} + 10^{-10} = 10^{10}$$

- 皆為離散的數：

兩相鄰數字之間並沒有其他數字

- 有最大數與最小數：

數字大小與使用的空間大小有關

- 數字總個數為有限個：

隨意給一個浮點數，此數剛好也是計算機所能代表的數的機率幾乎為零

# 浮點數儲存空間不足問題 (一)

## ■ 大小兩數運算

$$a = 1 \times 10^{-10} \quad , \quad b = 1 \times 10^{10}$$
$$(a + b) - b \neq a + (b - b)$$

## ■ 數值微分

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

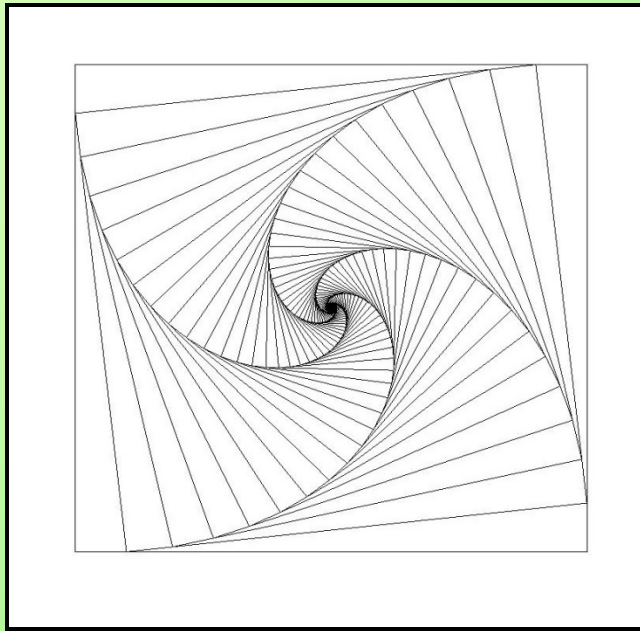
$$a = 1 \quad , \quad h = 1 \times 10^{-20}$$

$$\rightarrow a + h = a \quad \rightarrow \quad f(a+h) = f(a)$$

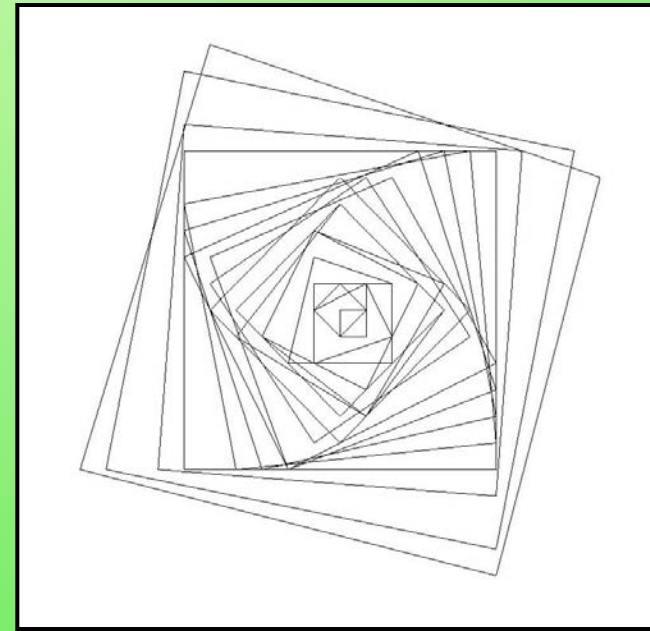
$$\rightarrow \quad f'(a) = 0 \quad \text{與函數 } f(x) \text{ 無關}$$

# 浮點數儲存空間不足問題 (二)

- 浮點數精確度不足也會發生在電腦繪圖上



內聚旋轉方形



近收斂點附近圖形