

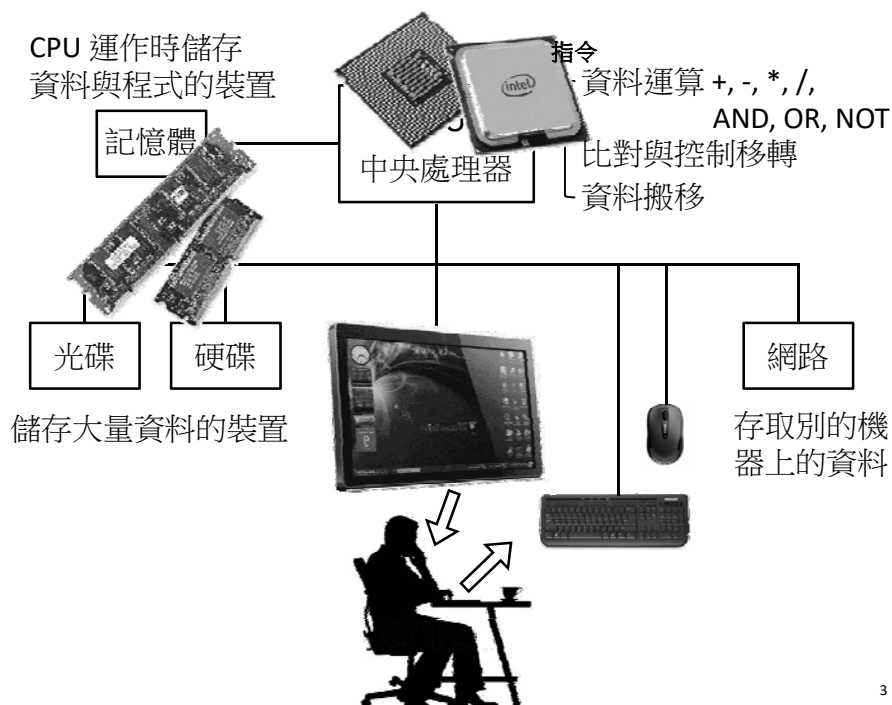
『程式』是什麼？ 『寫程式』到底在做什麼事？

丁培毅
NTOUCSE 程式設計體驗

1



2



3

資料存放到記憶體/由記憶體取出

- 記憶體是 CPU 不可缺少的夥伴, 沒有它的話, CPU 就像是重度失憶症患者, 做完的事、算出來的結果, 沒有一個記得, 其實該怎麼完成任何一件事的步驟也是不記得的, 所有的指令和資料都不知道記在哪裡
- 程式裡看到的記憶體。。。變數

int x, y;

x = 12;

y = x + x / 3;

y = x - y;

int num = 35;

printf("我們班上有 %d 位同學\n", num);

記憶體模型

二進位	編號	位元組
0000	0110	
0001	0111	
	:	
x	0101	12
	0110	
y	0111	46
	1000	
num	1001	35
	1010	

取名字, 規劃存放格式 ----- int

存放資料 ----- x =

取出資料 ----- x, y

x, y, num 稱為變數

4

要求 CPU 把資料顯示在螢幕上

```
#include <stdio.h>
```

```
int main()
```

```
{
    printf(" ");
    printf("My name is Bob.\n");
    return 0;
}
```

```
q:\>testPrintf
Hello World!
My name is Bob.
```

```
int num = 35;
```

```
printf("我們班上有 %d 位同學\n", num);
```

我們班上有 35 位同學

5

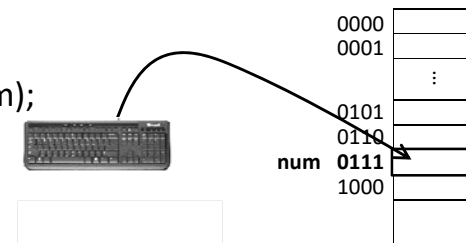
要求 CPU 由鍵盤讀取資料

- 操作者由鍵盤輸入資料以後, CPU 把資料依照指定的格式放進記憶體 (變數) 裡

```
#include <stdio.h>
```

```
int main()
```

```
{
    int num;
    scanf("%d", &num);
    return 0;
}
```



6

CPU 執行算術與邏輯運算

- +, -, *, /, AND, OR, NOT, XOR



```
int x = 13, y = 3;
```

```
y = x + x / y;
```

```
y = x - y;
```

整數除法: 商為 4, 先乘除後加減

浮點數

```
double r, s = 12.5;
```

```
r = s * s;
```

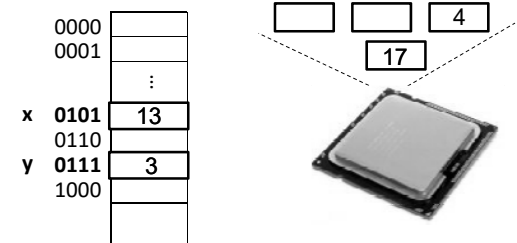
7

CPU 搬動資料

- 記憶體和記憶體之間, 記憶體和暫存器之間

```
int x = 13, y = 3;
```

```
y = x + x / y;
```



- 記憶體和硬體裝置 (硬碟、鍵盤、網路...) 之間

8

測試比對與控制移轉

- CPU 自己沒有智慧, 但是下面這段程式裡面有智慧行為裡最基本的 **條件判斷**

```
int x = 13, y = 3;
y = x + x / y;
y = x - y;
if (y < 0) {
    printf("y 存放的數值是負數\n");
}
else {
    printf("y 存放的數值大於或等於 0\n");
}
```

9

C 程式架構

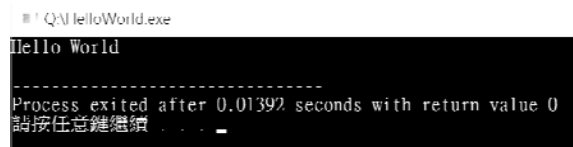
- 每一個 C 程式裡都一定有一個 **main** 函式

```
#include <stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```

大括號裡面就放你要求 CPU 執行的動作序列

程式練習一: Hello World

- 請寫一個 C 程式, 在螢幕上列印出 Hello World

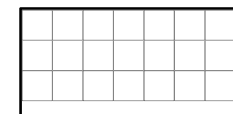


- 請注意練習
 - DevCpp 程式開發環境
 - 開新... 原始碼
 - 執行 (編譯並執行)
 - 練習 e-Tutor 繳交與評測環境
 - 記住基本的 C 程式架構
 - main 函式
 - return 0; 敘述
 - printf(...); 在螢幕輸出文字的工具
 - #include <...> 前處理器指令
 - 如果打英文覺得有點慢的話
 - 編輯/插入文字
 - 工具/編輯器選項/插入程式碼

11

程式練習二: 需要幾片地磚

- 有一塊長寬都超過 500 公分的長方形空地, 希望算出最多能鋪上幾塊完整的正方形地磚?
- 程式要求使用者輸入兩個**整數**代表地板的長度與寬度 (單位為公分), 再輸入一個**整數**代表正方形地磚的邊長 (單位為公分), 程式請計算總共需要幾塊完整的地磚?



- 輸入測試資料:
1230 2100↵
20↵
- 輸出測試資料:
6405↵

- 計算方法:
1230/20 ⇒ 61 餘 10
2100/20 ⇒ 105 餘 0
61 * 105 ⇒ 6405

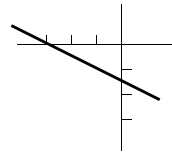
length 1230 ÷ side 20 = num_x 61
width 2100 ÷ 20 = num_y 105
61 × 105 = total_num 6405
int length, width, ...;

程式撰寫範例一

- 請寫一個程式計算直線 $ax + by + c = 0$ 和 X 軸的交點

- 例如 $a = 1, b = 2, c = 3$

和 X 軸的交點為 $(-3.0, 0.0)$



```
#include <stdio.h>
int main() {
    double a=1.0, b=2., c=3, x, y=0;
    x = -(b * y + c) / a;
    printf("和 X 軸的交點為 (%f, 0.0)\n", x);
    return 0;
}
```

q:\>test01

和 X 軸的交點為 (-3.000000, 0.0)

格式命令

13

範例一 (cont'd)

- 3.000000 格式和要求不太一樣, 調整一下

```
#include <stdio.h>
int main() {
    double a=1, b=2, c=3, x, y=0;
    x = -(b * y + c) / a;
    printf("直線 %.1f x + %.1f y + %.1f = 0 ", a, b, c);
    printf("和 X 軸的交點為 (%.1f, 0.0)\n", x);
    return 0;
}
```

q:\>test02

直線 1.0 x + 2.0 y + 3.0 = 0 和 X 軸的交點為 (-3.0, 0.0)

14

範例一 (cont'd)

- 係數 a, b, c 可不可以不要是固定的? 由鍵盤輸入

```
#include <stdio.h>
int main() {
    double a, b, c, x, y=0;
    printf("請輸入直線  $ax + by + c = 0$  的係數: ");
    scanf("%lf%lf%lf", &a, &b, &c);
    x = -(b * y + c) / a;
    printf("直線 %.1f x + %.1f y + %.1f = 0 ", a, b, c);
    printf("和 X 軸的交點為 (%.1f, 0.0)\n", x);
    return 0;
}
```

"%lf%lf%lf"

X X

```
q:\>test03
請輸入直線  $ax + by + c = 0$  的係數: 1.2 -2.5 5.9
直線 1.2 x + -2.5 y + 5.9 = 0 和 X 軸的交點為 (-4.9, 0.0)
```

15

範例一 (cont'd)

- 當使用者可以自由輸入 a, b , 與 c 時, 前一頁的程式有可能發生執行時的錯誤

q:\>test03

請輸入直線 $ax + by + c = 0$ 的係數: 0 1 2
直線 0.0 x + 1.0 y + 2.0 = 0 和 X 軸的交點為 (-1.0, 0.0)

- 仔細想一下, 輸出的 x 發生錯誤, 有問題的敘述可能是

$$x = -(b * y + c) / a;$$

當係數 a 是 0 時, 除以 0 是會發生錯誤的!!

那怎麼辦?!

16

範例一 (cont'd)

- 修改的方法

```
if (a == 0) {
    printf("和 X 軸沒有交點\n");
}
else {
    x = -(b * y + c) / a;
    printf("和 X 軸的交點為 (%.1f, 0.0)\n", x);
}
```
- 當 **a** 的數值很小的時候, 算出來的結果也很嚇人

[illegible]

- 看完前面的範例以後, 應該要很清楚地知道:
 - **程式** 是操控電腦完成某項工作時一步一步的指令
 - **寫程式** 是你在教電腦一步一步執行你想要它完成的工作
- 想要順利地寫出一個程式, 你需要能夠手動一步一步解決問題, 把動作的順序記錄下來, 完整地用 C 語言的語法表示出來 ... 就是你的程式了

範例一 (cont'd)

- 如果改成

```
if (a == 0) {  
    printf("和 X 軸沒有交點\n");  
}  
else {  
    x = -(b * y + c) / a;  
    printf("和 X 軸的交點為 (%.1e, 0.0)\n", x);  
}
```
- 輸出的結果會變成

```
q:\>test03
請輸入直線 a x + b y + c = 0 的係數: 1e-300 1 2
直線 0.0 x + 1.0 y + 2.0 = 0 和 X 軸的交點為 (-2.0e+300, 0.0)
```

- 更有意義的改法 `if (a < 1e-10 && a > -1e-10) ... else ...`

程式撰寫範例二

- 請寫一個程式計算直線 $ax + by + c = 0$ 和直線 $dx + ey + f = 0$ 的交點
- 重點在於手動算出這兩條直線的交會點

$$\begin{aligned} d \neq 0 \text{ 時 } dx + ey + f = 0 &\Rightarrow x + e/d y + f/d = 0 \quad \dots \textcircled{0} \\ &\Rightarrow ax + a e/d y + a f/d = 0 \quad \dots \textcircled{1} \\ &\quad ax + by + c = 0 \quad \dots \textcircled{2} \end{aligned}$$

$$\textcircled{1} - \textcircled{2} \Rightarrow y = (c - af/d) / (ae/d - b)$$

$$y \text{ 代回 } \textcircled{0} \Rightarrow x = -e/d(c - af/d)/(ae/d - b) - f/d$$

程式撰寫範例三

- 讓我們寫一個程式計算一元二次方程式 $ax^2 + bx + c = 0$ 的兩個根 (拋物線與 x 軸交點)

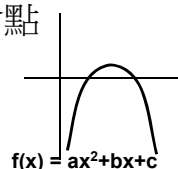
- 重點在於手動算出 拋物線 與 x-軸 的交會點

判別式 $d = b^2 - 4ac$

$$d > 0 \text{ 時 } \begin{cases} x_1 = (-b + \sqrt{d}) / (2a) \\ x_2 = (-b - \sqrt{d}) / (2a) \end{cases}$$

$d = 0$ 時只有一個交點 $x = -b / (2a)$

$d < 0$ 時沒有交點



21

設計這個程式的步驟

1. 設計基本程序

2. 變數設計

a. 讀取三個係數 `scanf("%lf%lf%lf", &a, &b, &c);`

b. 檢查是拋物線還是直線

如果是直線的話, 直接算和 x 軸的交點

c. 如果是拋物線的話, 分辨有沒有實數根

if (d < 0) { 計算判別式 double d;

... d. 判別式小於 0 時沒有實數解, 列印訊息

else if (d == 0) 等於 0 時只有一個實數解, 計算, 列印 double x;

{ 大於 0 時有兩個實數解, 計算, 列印 double x1, x2;

} x = -b / (2 * a); printf("%.3f\n", x);

else // if (d > 0) { x1 = (-b + sqrt(d)) / (2 * a); printf("%.3f, %.3f\n", x1, x2);

... x2 = (-b - sqrt(d)) / (2 * a);

}

22

程式撰寫範例四

- 請寫一個程式計算等差級數 (等差數列的和)

- 重點在於如何手動算出 等差級數 的結果

計算 $a_1 + a_2 + a_3 + \dots + a_n$

梯形公式

其中 $a_i = a_1 + (i-1) * d$, d 是公差

$$\frac{(a_1 + a_n) n}{2}$$

$$\begin{array}{ccccccc} a_1 & + & a_2 & + & a_3 & + & \dots + a_n \\ a_n & + & a_{n-1} & + & a_{n-2} & + & \dots + a_1 \\ \hline a_1 + a_n & & a_1 + a_n & & a_1 + a_n & & a_1 + a_n \end{array}$$

轉換成程式

23