

# 概論

## 本章大綱

- ▶ 1-1 程式語言概念與應用
- ▶ 1-2 程式語言的發展
- ▶ 1-3 程式的編譯、執行
- ▶ 1-4 C 語言的輸出入
- ▶ 1-5 C++ 語言輸出入
- ▶ 1-6 程式的追蹤與除錯

## 1-1 程式語言概念與應用

### 程式與電腦的關係

人與人之間溝通的工具稱為語言，世界上的諸多民族，因其發源地不同，所以就有許多語言。例如，華語、英語及德語等。其次，人與電腦溝通的工具，則稱為電腦程式語言。那為什麼沒有電視語言、冰箱語言或冷氣語言呢？那是因為這些機器的功能較為簡單，只要幾個按鈕就能發揮其功能，但是電腦的功能就非常多，多到連用整個鍵盤的所有按鍵都無法表現其功能，所以必須使用一些類似單字所組成的片語與敘述來發揮其所有功能，這些單字與片語的集合就稱為電腦程式語言，簡稱程式語言。就如同人類也無法用 26 個字母表達所有感受與思維，必須藉助這些字

母的排列組合，先組成單字，再由單字組合成片語與句子，才能充分表達其思維。目前流行的程式語言有 Java、C、C++、Visual Basic、C#、Python、JavaScript、PHP 等。

## 電腦程式設計

串連一些程式語言中的指令與資料，使其完成一件工作，就稱為電腦程式設計，簡稱程式設計。

## 運算思維

既然是電腦程式設計，那就要以電腦的運算思維方式去寫程式，而不是以人的思考方式寫程式。例如，以時間為例，人類是分別使用年、月、日、時、分、秒等六個變數代表一個時間點，兩個時間的差額是分鐘不夠時，去借 1 小時來減，日數不夠時，去借 1 個月來減，但電腦可不用這樣想，電腦可是有很強的資料結構，直接用一個 10 位數的長整數『秒』記憶與代表年、月、日、時、分、秒，所有的時間的長度與前後運算也都用『秒』完成。又例如，人類解開根號，是套用國中數學的  $(a+b)^2=a^2+2ab+b^2$ ，如下頁圖，但電腦就不用這麼麻煩，可直接用循序猜值或二分猜值法求解。以上時間的表示與儲存方式，稱為資料結構，解開根號的諸多方法則稱為演算法。電腦最基本演算法就是循序法的 for 與 while，最基本的資料結構是單一變數與一維或二維陣列結構，其中單一變數就很精彩，例如上面的時間問題，又例如，單複選題答案 ABCDE 要如何儲存較省事呢？這都已經在中學生資訊科技闡述，本書將進一步探討進階資料結構與進階演算法，例如，進階資料結構有結構、堆疊、佇列、鏈串列等，進階演算法有貪婪演算法與分而治之演算法，這都會在本書探討。

		3 7 2
1	法則 3	1 3 8 3 8 4 9
2	$3 \times 20 + 7 = 67$ $67 \times 7 = 469$	3 8 4 4 6 9
3	$37 \times 20 + 2 = 742$ $742 \times 2 = 1484$	1 4 8 4 1 4 8 4
4		0

## 程式設計功能

程式設計可將一連串重複的工作，使用程式語言，寫一段程式、儲存，然後就可無限次數重複使用，這樣可節省很多時間。例如以解一元二次方程式為例，使用者必須不斷的計算  $d = \sqrt{b^2 - 4ac}$ ， $x_1 = (-b + d) / 2a$ ， $x_2 = (-b - d) / 2a$ ，才能解出方程式，那如果寫成程式，每次要求解時，只要輸入係數 a,b,c 即可求出方程式的解。

## 應用

程式設計的應用已經與我們生活習習相關，以前只是每個家庭有一部電腦，現在幾乎人手一支手機，手機內即有千變萬化的通訊軟體、遊戲。以前只有科學家、銀行有運用電腦計算，但是現在的自駕車已經可以上路，鴻海郭台銘董事長也說五年後他的工廠不用開燈、不用人、不用供餐、不用供宿，使用機器人就可 24 小時生產，所以現在的程式設計可說是融入我們的生活了。其次，現在人口越來越少，越來越凸顯人口老化問題，以後居家生活照顧就要仰賴機器人了，每個人眼睛一張開，就要學習與電腦機器人一起生活。還有更可怕的事，以前是每個家庭一支電話，現在是每個人一支手機，不久的將來就會身邊的所有東西都有一個通

訊裝置，例如，冰箱、電視、冷氣、汽車都有一個通訊裝置，都可以控制與回報其狀態，那就是物聯網的時代，不僅人與人可以聯繫，就連人與物或物與物都可以自動聯繫，這麼複雜的工作，都是電腦程式的應用。

## 1-2 程式語言的發展

電腦語言依其演化先後，大致可分為低階語言與高階語言。

### 低階語言

低階語言依其演化先後，又可分為機器語言與組合語言。

### 機器語言

機器語言是由一堆 0 與 1 所組成的單字。例如，要在螢幕輸出字元『A』，所需鍵入的單字如下：

```
B2 41 B4 02 CD 21 CD 20
```

由於這些單字的來龍去脈過於複雜，絕非一般人所能體會，只因為要於螢幕顯示一個『A』，就要鍵入以上單字，那如果要計算一個正方形面積，所需鍵入的單字可能比火車還長，那不如用筆算還快，還要計算機作什麼！其次，電腦剛發明時並沒有鍵盤，要鍵入以上單字，必須用 8 個位元一組的指撥開關，分別輸入 8 次（共 8 個單字）。例如，B2 是由 (10110010) 所組成，則必須先將 8 個指撥開關分別設定為 (on,off,on,on,off,off,on,off)，由此可見當時使用機器語言的艱辛，真是未蒙其利，先受其害。

### 組合語言

學習與使用機器語言是那麼艱辛，而且電腦的體積又非常龐大，只有非常少數的人願意嘗試使用，但是當時科學家並未氣餒，為

了推廣這一神奇而且好用的工具，所以有組合語言的研究。所謂組合語言就是使用一些助憶碼，這些助憶碼類似一些簡單的英文單字，以便協助使用者記憶與書寫所要完成的程式。例如，上例中要將字元『A』顯示於螢幕，所需組合語言如下：

```
MOV DL, 41
MOV AH, 2
INT 21
INT 20
```

其中，『MOV』是 MOVE 的縮寫，『41』是字元 A 的 ASCII 碼，『DL』與『AH』則是 CPU 的內部暫存器，『INT 21』是呼叫字元顯示副程式，『INT 20』則是呼叫程式結束處理副程式。也就是程式發展已經有一脈絡可循，一切遵循一些簡單規則，方便使用者的查詢與記憶。有了組合語言的這一好用工具，電腦終於可以突破實驗室，推廣到大學以上的資訊電子相關科系。

## 高階語言

有了組合語言，電腦使用的方便性可說向前邁向一大步，但是電腦的使用卻僅限於資訊電子相關科系，因為不同的 CPU 就要有不同的組合語言，也就是程式沒有相容性。其次，這些助憶碼也要花費很多時間學習，而且每一小小功能的程式長度都要超過一頁，科學家並不以此為滿足，就有高階語言的研發。例如，前例要所以在螢幕顯示字元 A，若使用 C 語言，則只要撰寫如下：

```
printf("A");
```

## 程式語言的種類

一份由電機電子工程學會發行的雜誌 IEEE，透過分析來自 10 種來源的 12 個指標，算出目前熱門程式語言排行榜如下圖，C 語言

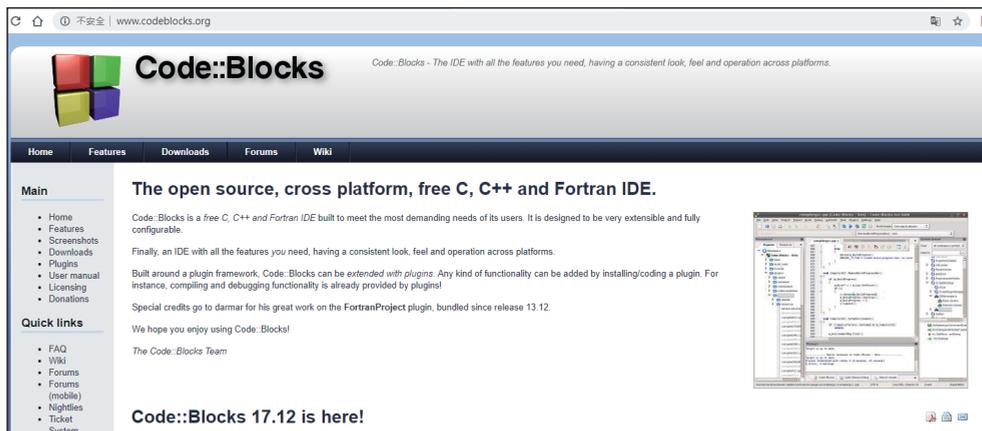
還是目前第一名。其次，大學端設計的『大學程式設計先修檢測 APCS』學科，也以 C 語言為程式設計語言，中學生學習 C 語言可完全無縫接軌高中與大學程式設計課程，所以本書選用 C 語言為高中生進階程式設計的程式語言。

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

## 1-3 程式的編譯與執行

### CodeBlocks

CodeBlocks 是目前最熱門的 C/C++ 免費開放式跨平台整合式開發環境 (IDE)，APCS 考試也提供此 IDE，官網如下圖：



點選『Download』，畫面如下圖：



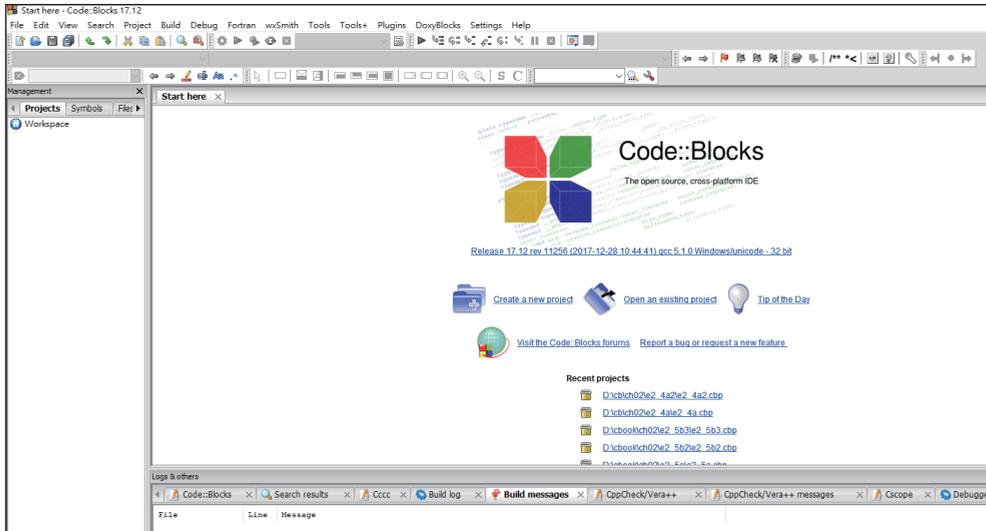
點選『Download the binary release』，畫面如下圖：



初學者請直接點選『codeblocks-17.12mingw-setup.exe』，因為它才有內含『C/C++』編譯軟體，不然要自備編譯軟體。

## 安裝軟體

下載的檔案就是『.exe』執行檔，請在『下載區』按兩下程式，就可自動執行。開啓『CodeBlocks』，畫面如下圖：

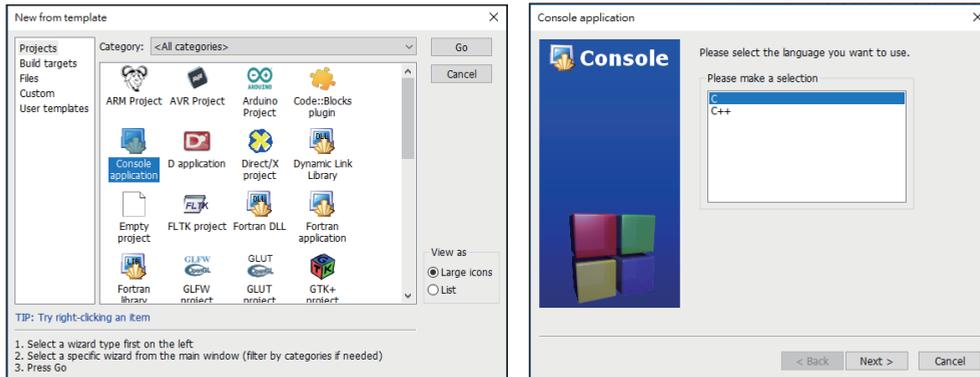


### 範例1-2a

請寫一程式，可以於螢幕輸出『Hello』。本例所需程式僅為『`printf("Hello");`』。

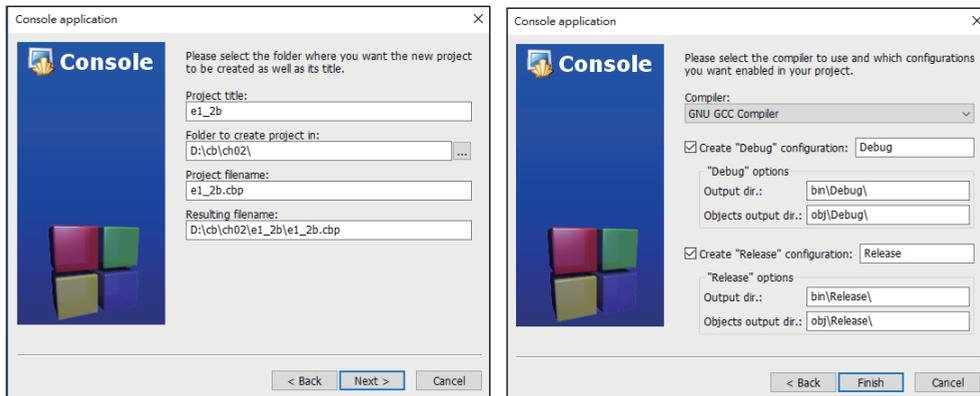
### 操作步驟

1. 開新專案。點選功能表的『File』/『New』/『Project』，畫面出現要求點選專案類型，如下圖左，本例專案類型點選『Console Application』，再點選『Go』，畫面如下圖右，可點選專案類型，本例點選『C』。

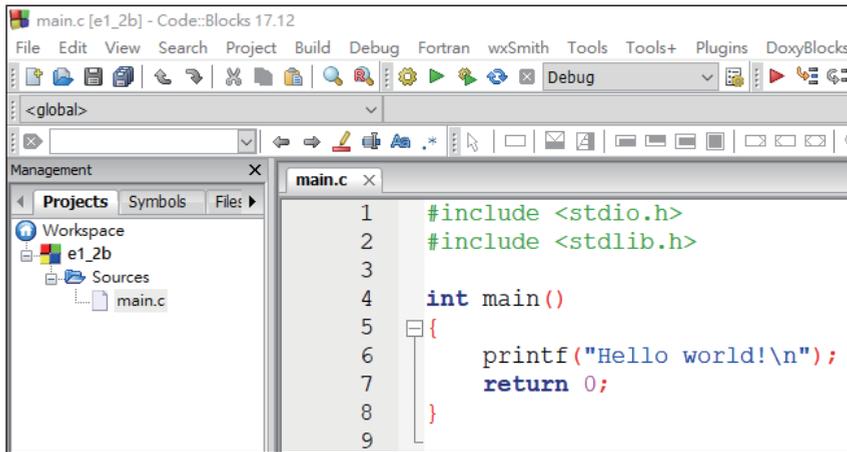


## 2. 點選資料夾與輸入專案名稱。

本例點選資料夾為『d:\cb\ch02』，專案名稱輸入『e1\_2b』，如下圖左，電腦將會自動建立資料夾『e1\_2b』。接著下圖右通通用預設值就好。



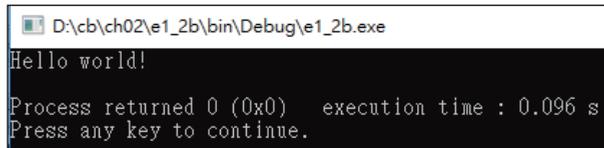
## 3. 完成以上操作，畫面如下圖：（請點開『Source』前面『+』，再按二下『main.c』）



```
main.c [e1_2b] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks
<global>
Management
Projects Symbols File:
Workspace
  e1_2b
    Sources
      main.c
main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      printf("Hello world!\n");
7      return 0;
8  }
```

#### 4. 執行程式。

點選功能表的『Build』 / 『Build and Run』，即可執行程式，畫面如下圖：



```
D:\cb\ch02\e1_2b\bin\Debug\e1_2b.exe
Hello world!
Process returned 0 (0x0)   execution time : 0.096 s
Press any key to continue.
```

5. 以上 `printf()`，就是本例所執行的程式，往後您要寫任何的程式，只要將此敘述刪除，並在此位置寫下您的程式就可以。
6. 完成以上操作，資料夾『e1\_2b』，如下圖。已經有專案檔與 `main.c` 原始程式檔。(APCS 考試請點選下圖的 `main.c` 再上傳)



名稱	修改日期	類型	大小
bin	2018/12/16 上午 ...	檔案資料夾	
obj	2018/12/16 上午 ...	檔案資料夾	
e1_2b.cbpp	2018/12/16 上午 ...	CBP 檔案	2 KB
main.c	2017/12/25 上午 ...	C 檔案	1 KB

7. 以上步驟 1，若點選『C++』，則新增的程式樣版如下：

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

C++ 的輸出入改為 `cout` 與 `cin`，其餘的運算子、運算式、決策、迴圈、陣列都繼承 C。C++ 既然號稱繼承 C，那 C 語言的所有程式，都可以在 C++ 完成，唯獨若使用到 C 的函式，也要載入其標頭檔。例如，若要使用 C 語言的輸出入，那也要載入

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

請留意 C 的標頭檔都有『.h』，C++ 則都沒有，例如 C++ 的輸出入都放在

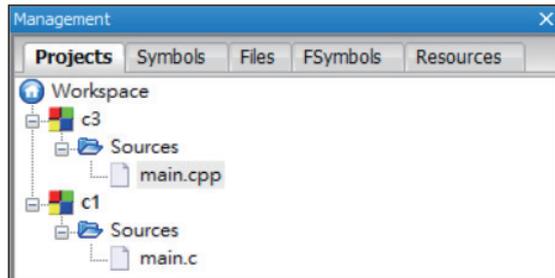
```
#include <iostream>
```

也就是若您開啓 C++ 專案，載入以下標頭檔，那 C 與 C++ 的輸出入都可以使用。

```
#include <stdio.h> //可以使用scanf()、printf()
#include <stdlib.h>
#include <iostream> //可以使用cout、cin
```

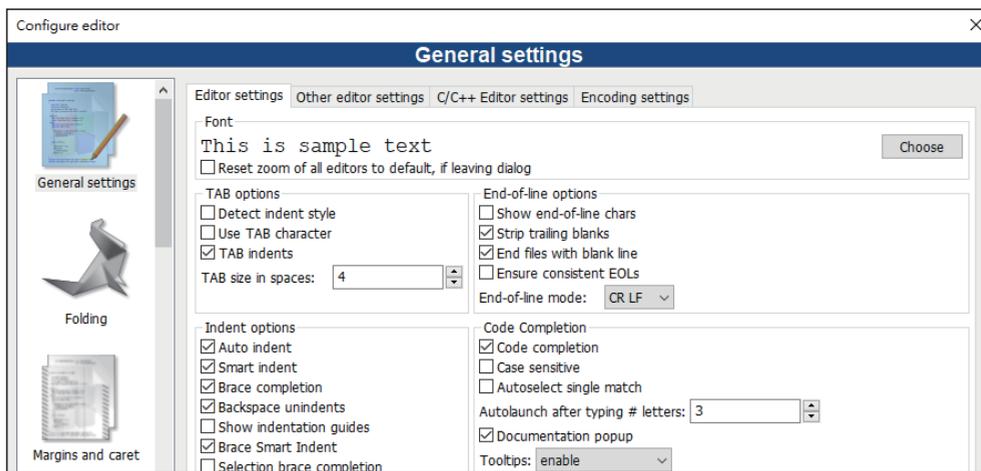
8. 如何切換『Project』？若同一個 Workspace 同時載入兩個 Projects，如下圖，那要如何轉換不同 Projects？請留意是按兩下下圖的『c3』或『c1』，才會轉換 Project，若按兩下下圖的 `main.c` 或 `main.cpp`，那僅是改變顯示程式而已，當您執

行程式時，還是執行上一個 Project。



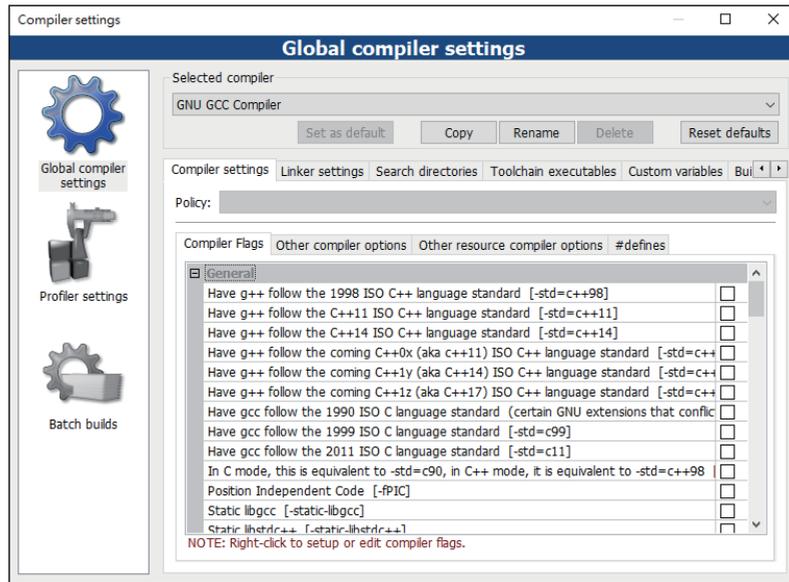
9. 如何調整字型大小。

請點選功能表『Settings』/『Editor』，畫面如下圖，請繼續點選『Choose』即可更改字型大小。

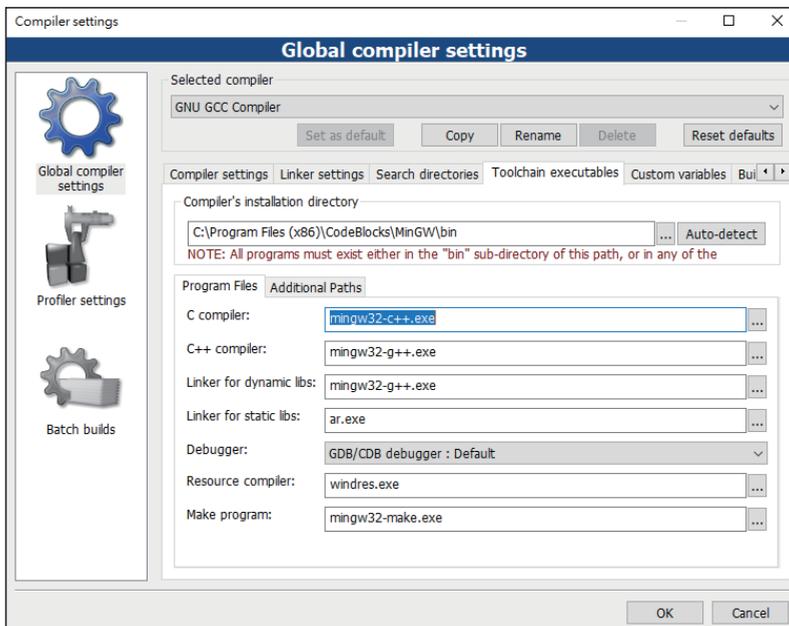


10. 若出現找不到編譯器？（若之前您有安裝其他 C 語言，有可能系統誤判而找不到編譯器）

(1) 請點選功能表『Setting』/『Compiler』，畫面如下，在這一畫面，您可以指定要使用哪一版本編譯器。（C 語言編譯器有三個版本，分別是 1990、1999 與 2011，往後線上查詢時就會明瞭）



- (2) 繼續點選『Toolchain executables』，畫面如下圖，請留意您的編譯器路徑是否正確，若不正確請在此修改。



11. 若左邊的『Management』視窗不見。請點選『View/Manager』。

## ■ 補充說明

1. 鍵入程式應該要有層次感。例如，函式實作應該要使用 `tab` 鍵內縮（初學者常用空白鍵內縮，那會很吃力，且無法完全整齊）。例如，以下的 `printf("Hello");`，請先鍵入 `tab` 鍵，再鍵入程式。

```
int main(int argc, char *argv[]) {  
    printf("Hello");  
    return 0;  
}
```

還有遇到 `for`、`do`、`while` 迴圈或 `if` 等，也都要先用 `tab` 鍵內縮，請看以下各章程式。

## 常犯的錯誤

以下是一些鍵入程式時常犯的錯誤，整理如下，方便初學者除錯。

1. 忘記在敘述行尾端打上分號（；），例如：

```
printf("Hello")
```

或是不應該出現分號的地方，多打了分號。例如：

```
#include <stdio.h>;  
if (a>b);
```

且分號『;』是英數下的半形，不是中文狀態下的全形。

2. 大小寫未區分。C 語言規定大小寫的字母是不同的字元，應注意字母的輸入。例如，

```
printf("Hello");  
Printf("Hello"); //請留意P的大小寫
```

就不一樣了。

3. 拼錯字。例如將

```
printf("Hello");
```

打成

```
printf("Hello"); //請留意m與n或l與1
```

那指令就錯了，將會出現錯誤信息。

4. 括弧未配成對。有一個左大括弧『{』，務必要配一個右大括弧『}』；同樣地，左小括弧『(』務必要配一個右小括弧『)』；左中括弧『[』務必要配右中括弧『]』。初學者經常會忘記打上右括弧，導致語法錯誤。
5. 引號未成對。不論雙引號『"』(用於字串)，或是單引號『'』(用於字元)的使用，都應該成雙成對出現。
6. 不該出現的空白符號。例如，把

```
printf("Hello");
```

打成

```
print f("Hello");
```

也不行。

## 1-4 C的輸出入函式

C語言是使用 `printf()` 和 `scanf()`，來進行所有輸出入工作，請看以下說明。

### `printf()`

`printf()` 可輸出結果，其中括號內須放置一對大括號 ("") 來將欲輸出的資料當作引數傳給 `printf()` 函式。例如，

```
printf("a");  
printf("aa");
```

的輸出結果是

```
aaa
```

但若要跳列，則應加上跳脫字元『\n』，例如，

```
printf("a\n");  
printf("aa");
```

的輸出結果是

```
a  
aa
```

以上程式片段，完整程式如下，請自行實習。

```
#include <stdio.h>  
#include <stdlib.h>  
int main() {  
    printf("a");  
    printf("aa");  
    return 0;  
}
```

其次，如果變數的輸出，則必須在引數敘述中加入一個列印格式（如下表）來控制輸出的格式。輸出過程中如須換行，也是以跳脫字元序列『\n』來完成。

資料型別	列印格式
整數	%d
長整數	%ld
float、double 等浮點數	%f
字元	%c
字串（字元陣列）	%s

例如，

```
int a=3, b=5;
printf("a=%d b=%d", a, b);
```

的結果是

```
a=3      b=5
```

又例如，

```
int a=3, b=5;
printf("a=%d\n", a);
printf("b=%d", b);
```

的結果是

```
a=3
b=5
```

## scanf()

scanf() 可輸入資料，其使用方式與輸出 printf() 是一樣的方式，二者均須使用到一些列印格式來控制輸出入的資料。因此，輸入字元、字串或數值等會有不同的處理方式，請看以下說明。

### 輸入字元

輸入字元的敘述如下：

```
char a;
scanf("%c", &a);
```

將輸入的資料放入在某個變數所指定位址中，取址符號 (&) 用來取得欲放置輸入資料的變數位址。例如，以下程式可輸入一個字元。

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    char a;
    printf("Please press any char: ");
    scanf("%c", &a);
    printf("Your char is %c", a);
    return 0;
}
```

### ■自我練習.....

1. 請連續輸入三個字元，並輸出結果。(若出現問題，請練習線上查詢)

### 輸入字串

正如我們所強調的，在 C 語言中的字串是由字元陣列所組成，因此在輸入字串時，我們也必須利用字元陣列來完成輸入。例如，以下程式可輸入一個字串。

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    char c[6];
    printf("Please enter a string: ");
    scanf("%s", &c);
    printf("Your enter is %s", c);
    return 0;
}
```

### ■自我練習.....

1. 請連續輸入兩個字串，並輸出結果。

## 輸入數值

常用的數值型別有 int、long、float、double 等，若要輸入這些型別的數值，則應使用不同的列印格式，例如，%d、%ld、%f 與 %lf 等。例如，請輸入以下程式，並觀察執行結果。

```
#include <stdio.h>
int main(int argc, char** argv) {
    int a;
    long b;
    float c;
    double d;
    printf("Enter an integer:   ");
    scanf("%d", &a);
    printf("Enter a long integer: ");
    scanf("%ld", &b);
    printf("Enter a float:       ");
    scanf("%f", &c);
    printf("Enter a double:      ");
    scanf("%lf", &d); //若誤用為"%f"也很慘，請看下面敘述
    printf("\n Your enter..... \n");
    printf("integer      a =%d \n", a);
    printf("long integer b =%ld \n", b);
    printf("float         c =%f\n", c);
    printf("double        d =%lf\n", d);
    scanf("%f", &d);
    printf("double        d =%lf\n", d);
    return 0;
}
```

## 自我練習

1. 請鍵入以下程式，輸入 3.4，並觀察執行結果。

```
#include <stdio.h>
int main(int argc, char** argv) {
    double d;
    printf("Enter a double:     ");
    scanf("%f", &d);
    printf("double          d =%lf\n", d);
    return 0;
}
```

## 1-5 C++ 輸出入函式

C++ 的輸出入採用 cout 與 cin，此兩物件放在 <iostream>，所以要載入以下標頭檔：

```
#include <iostream> //C的標頭檔都有.h，C++都沒有.h
```

其次也要指定其命名空間為 std（C++ 是將函式改為物件，要指定物件的命名空間，所謂命名空間就是指物件存放的所在資料夾），如以下敘述：

```
using namespace std; //
```

以下是 CodeBolcks 開啓 C++ 新專案，就放好的程式樣版。

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

同樣的道理，程式就放到 main() 內就好。

### cout

cout 物件可輸出指定任何文數字與變數內容，其與 C 最大的不同，就是不用去管變數的資料型態。其語法如下：

```
cout <<字串或變數 [<<字串或變數] [<<endl] ;
```

語法中的中括號『[]』表示此部分可省略，所以上述語法最簡單的就是

```
cout <<字串或變數 ;
```

例如，

```
int a=5;
cout<<a;//5
```

又例如，

```
int a=5;
cout <<"a="<<a<<endl;
```

可輸出『a=5 』。其中第一個『<<』接雙引號，表示內容為字串，就直接輸出；第二個『<<』接 a，表示其內容是變數，將會輸出此變數對應的內容。endl(endline) 代表此列的輸出結束，跳離此列，往後的輸出移至下一列的開頭。例如，

```
cout <<"Good"<<endl;
cout <<"Good";
```

可輸出

```
Good
Good
```

但是，

```
cout <<"Good";
cout <<"Good";
```

則輸出

```
GoodGood
```

**範例1-5a**

示範輸出以上結果。

**程式列印**

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    cout <<"Good"<<endl;
    int a=5;
    cout <<"a="<<a<<endl;
    return 0;
}
```

**自我練習**

1. 請於 C++ 環境完成以上程式。

**格式**

若要進行欄位寬度、浮點小數位數、輸出靠左或靠右，則可使用 cout 物件所提供的 width、setf 等屬性，分別說明如下：

**width**

width 屬性可設定欄位寬度，且預設向右對齊。例如，

```
int a1=1,a2=10,a3=100,a4=1000;
cout.width(5);cout<<a1<<endl;
cout.width(5);cout<<a2<<endl;
cout.width(5);cout<<a3<<endl;
cout.width(5);cout<<a4<<endl;
```

的結果如下圖。



```
1
 10
 100
1000
```

請留意，`cout.width(5)` 的效期只有一次，也就是每次 `cout` 之前都要設定一次。請讀者試著比較以下程式的不同。

```
cout.width(5);
cout<<a1<<endl;
cout<<a2<<endl;
cout<<a3<<endl;
cout<<a4<<endl;
```

## setf

`setf` 可設定小數點位數，資料靠左或靠右。例如，以下敘述可設定小數點後面固定取兩位。『`ios`』是命名空間名稱，『`::`』是範圍運算子，『`floatfield`』即表示浮點數欄位。

```
cout.setf(ios::fixed,ios::floatfield);
cout.precision(2);
```

以下敘述可設定資料靠右，靠左是 `left`，`adjustfield` 是自動調整欄位。(要靠左或靠右都要加上 `ios::adjustfield`)

```
cout.setf(ios::right,ios::adjustfield);
```

以下程式可設定每個欄位寬度為 8，小數點取兩位，欄位自動靠右。

```
float b1=3.14159;
// 小數點取兩位
cout.setf(ios::fixed,ios::floatfield); //指定浮點數欄位
cout.precision(2); //浮點數欄位取兩位
// 欄位自動靠右
cout.setf(ios::right,ios::adjustfield); //靠左 left 靠右 right
cout.width(8);cout<<b1<<endl; // 欄位寬度為8
cout.width(8);cout<<b1*20<<endl;
cout.width(8);cout<<b1*200<<endl;
```

以上程式輸出如下：

```
3.14  
62.83  
628.32
```

## cin

cin 物件可輸入文數字至指定的變數，其與 C 最大的不同，就是不用指派變數的資料型態，變數也不用取址。其語法如下：

```
cin>>變數 [>>變數] ;
```

例如，

```
char a;  
cin >>a;  
cout <<"a="<<a<<endl;
```

可輸入字元至變數 a。又例如，

```
int b;  
cin >>b;  
cout<<"b="<<b<<endl;
```

可輸入整數至變數 b。又例如，

```
char c[5];  
cin >>c;  
cout<<"c="<<c<<endl;
```

可輸入字串至變數 c。以下程式則可連續輸入 2 個變數，輸入時，變數之間請以空白隔開。

```
int d1,d2;  
cin >>d1>>d2;
```

**範例1-5b**

示範以上 cin 的輸入結果。

**輸出結果**

```
Please input a char:k
a=k
Please input a integer:24
b=24
Please input a string:Horng
c=Horng
Please input integer integer:24 36
d1=24
d2=36
```

**程式列印**

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    char a;
    cout<<"Please input a char:";
    cin >>a;
    cout <<"a="<<a<<endl;
    int b;
    cout<<"Please input a integer:";
    cin >>b;
    cout<<"b="<<b<<endl;
    char c[5];
    cout<<"Please input a string:";
    cin >>c;
    cout<<"c="<<c<<endl;
    int d1,d2;
    cout<<"Please input integer integer:";
    cin >>d1>>d2;
    cout <<"d1="<<d1<<endl;
    cout <<"d2="<<d2<<endl;
    return 0;
}
```

## string

C 因為要精簡編譯器，所以沒有 string 型態，所有字串處理都要使用字元陣列，C++ 則彌補此缺憾，新增 string 型態。以下程式可輸入一個字串，並輸出。

```
string a;
cin>>a;
cout<<a;
```

若要輸入含空白字串，則要使用 getline() 函式，程式如下：

```
string b;
getline(cin,b); //可輸入含空白字串
cout<<b;
```

## ■自我練習.....

1. 請於 C++ 環境完成以下程式，且都輸入 This is a book.，並觀察結果。

題號	程式	輸出結果
1	<pre>#include &lt;iostream&gt; #include &lt;string&gt; using namespace std; int main(int argc, char** argv) {     string a,b;     cin&gt;&gt;a;     cout&lt;&lt;a&lt;&lt;endl;     b=a;//字串的指派     cout&lt;&lt;b;     return 0; }</pre>	

2	<pre>#include &lt;iostream&gt; #include &lt;string&gt; using namespace std; int main(int argc, char** argv) {     string b;     getline(cin,b); //可輸入含空白字串     cout&lt;&lt;b;     return 0; }</pre>	
---	---	--

## 1-6 程式的追蹤與偵錯

程式的錯誤有兩種，一種是文法的錯誤，例如，保留字拚錯單字、變數單字拚錯、缺大括號、缺中括號、缺分號，這些錯誤編譯器都會停留在錯誤點，並給予提示，這就比較容易除錯，請自己多加練習。第二種錯誤是程式可以通過編譯，但執行結果不對，有可能邏輯錯誤、資料型態錯誤、演算法錯誤，這就要一步一步追蹤程式的執行結果，追蹤的方式有兩種，第一種是自己於程式適當地方『輸出變數內容』；第二種是使用 CodeBlocks 所提供的 Debug 工具，以上兩種追蹤方式，分別說明如下：

### 輸出變數

當程式輸出不如預期，就必須自己輸出變數結果，或者程式演算步驟較多，都可提早輸出變數，確認本步驟是否正確。例如，下圖左是原本的程式，那我們可以在適當的地方，使用輸出函式（C 是 printf()、C++ 是 cout）輸出變數內容，如下圖右，首先，資料一輸入，那馬上輸出結果，這樣就可觀察資料輸入是否確實，其次，計算完 d，那也馬上輸出 d，這樣可以確認此步驟是否正確，這樣就可一步一步觀察執行結果，逐步縮小程式錯誤範圍。

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(int argc, char *argv[]) {
    int a,b,c;//係數
    double d,x1,x2;
    printf("input a,b,c :");
    scanf("%d,%d,%d",&a,&b,&c);
    d=sqrt(b*b-4*a*c);
    x1=(-b+d)/(2*a);
    x2=(-b-d)/(2*a);
    printf("x1=%f\n",x1);
    printf("x2=%f\n",x2);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main(int argc, char *argv[]) {
    int a,b,c;//係數
    double d,x1,x2;
    printf("input a,b,c :");
    scanf("%d,%d,%d",&a,&b,&c);
    printf("a=%d,b=%d,c=%d\n",a,b,c);
    d=sqrt(b*b-4*a*c);
    printf("d=%lf",d);
    x1=(-b+d)/(2*a);
    x2=(-b-d)/(2*a);
    printf("x1=%f\n",x1);
    printf("x2=%f\n",x2);
    return 0;
}
```

又例如，下圖左是原始程式，我們也可以在迴圈裡面加上 printf() 輸出變數的執行結果，這樣就可以觀察變數的變化。

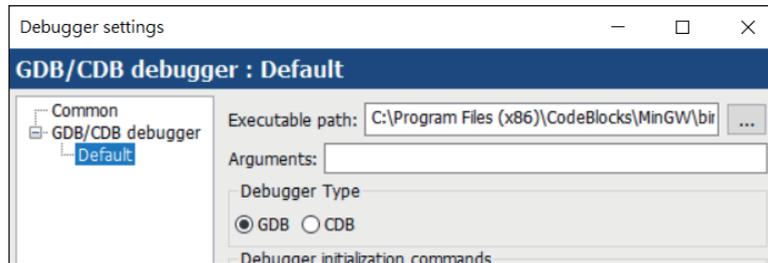
```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a=8,b=3,q=0;
    while(a>=b) {
        a=a-b;
        q++;
    }
    printf("quotient = %d\n", q);
    printf("remainder= %d\n", a);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a=8,b=3,q=0;
    while(a>=b) {
        a=a-b;
        q++;
        printf("a=%d,b=%d,q=%d\n",a,b,q);
    }
    printf("quotient = %d\n", q);
    printf("remainder= %d\n", a);
    return 0;
}
```

## Debug

大部分的編譯器都有提供 Debug 功能，讓您一步一步追蹤程式的運行，此稱為 Debug 功能，CodeBlocks 也不例外，功能選單或工具列都有 Debug，讓您可以一步一步追蹤程式執行的順序與變數

的結果。其次，因為版本的關係，有些版本要自己設定 Debug 程式的所在位置。以版本 17.12 為例，設定的方法如下：請點選功能表『Settings/Debugger /Default』畫面如下圖：



請於上圖點選 gdb32.exe 所在的路徑：（摘自 <https://www.itread01.com/content/1543644082.html>，在此致謝）

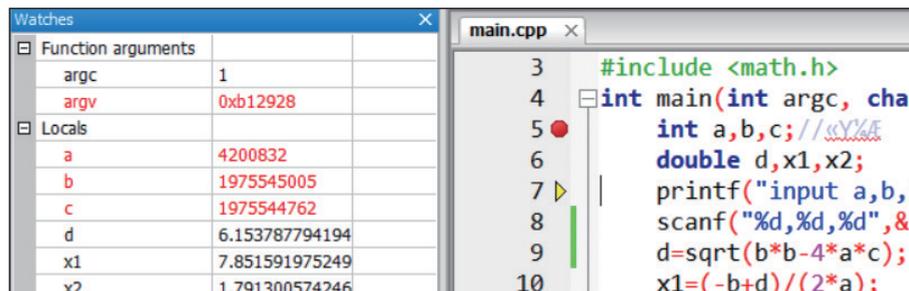
*C:\Program Files (x86)\CodeBlocks\MinGW\bin\gdb32.exe*

### 範例1-6a

示範 Debug 的操作步驟。

### 操作步驟

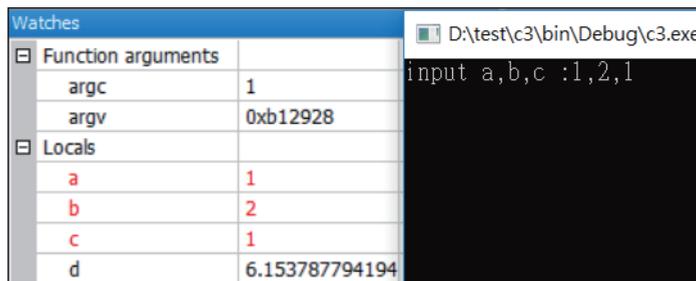
1. 開啟『Watches』窗格。點選功能表『Debug/Debugging windows/Watches』，畫面如下圖：
2. 設定程式中斷點。請在程式列編號的右邊使用滑鼠點一下，如下圖的紅色圓點。（再按一下可取消中斷點）



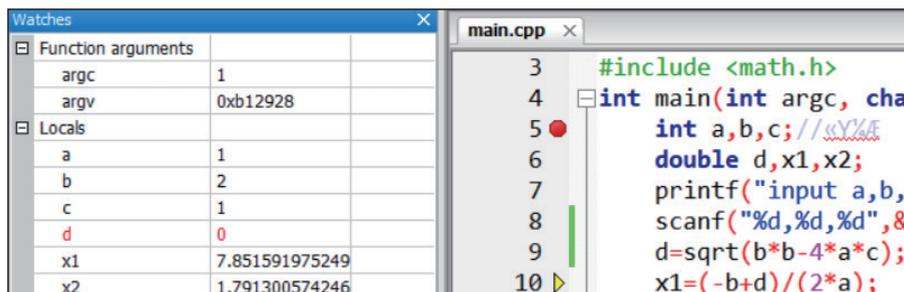
3. 下圖是 Debug 的工具列圖項。



4. 開始 Debug。請按一下工具列的『Run to cursor』，程式就會執行到中斷點，畫面同上上圖，請留意會出現黃色三角形，此黃色三角形的程式列，就是下一步要執行的敘述。
5. 請自行按一下『Next line』，即可執行下一列，該輸入資料時，就點選程式輸入畫面，輸入資料，如下圖右：
6. 下圖是輸入資料後的畫面，Watches 已經列出變數 a,b,c 的結果，請自行觀察是否確實。



7. 下圖是執行  $d = \sqrt{b*b - 4*a*c}$  後的畫面，d 值已經算出來了，請自行觀察結果是否確實。



8. 按一下工具列的『Stop debugger』，即可停止中斷的偵測。

## 自我練習

1. 以下程式，請將中斷點設在 char r 這一敘述，並逐一列執行程式，觀察程式執行順序。

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a;
    char r;
    //printf("input a grade: ");
    //scanf("%d",&a);//&
    a=80;
    if(a>=90)          /* 高於90分爲A */
        r='A';
    else
        if(a>=80)     /* 介於 80與90分爲B */
            r='B';
        else
            if(a>=70) /* 介於 70與80分爲C */
                r='C';
            else
                r='D';          /* 不符合上述情況則爲D */
    printf("the grade is %c",r);
    return 0;
}
```

2. 於以下程式，請將中斷點設在 int i, sum=0，並逐一執行程式，觀察程式執行順序與變數的內容。

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int i, sum=0;
    for(i=1; i<=10; i++) {
        if(i==5)
            continue;          /* 繼續下一個 i */
        if(i==8)
            break;             /* 強制離開迴圈 */
        printf("sum=%d ,i=%d \n",sum,i);
    }
}
```

```
        sum+=i;
    }
    printf("when out of loop, i=%d ,sum=%d\n",i ,sum);
    return 0;
}
```

請自行使用 Debug 偵測 Debug 單元前面的除法運算程式的運行過程。