

2017 年程式設計先修暑期夏令營

課程大綱 (助教版)

November 10, 2017

本課程獲教育部扎根高中職資訊科學教育計畫補助

這份講義，僅提供給老師和助教作為教學參考之用，講義的內容，主要針對瘋狂程設上的學習問題，提供參考的思考流程和程式解答，供作教學參考之用。

這份講義還在修改階段，如果有什麼修正的建議，可以提供給我們，感謝大家。

Contents

1	8/14(一) 上午：環境設定	4
1.1	行政事項	4
1.2	認識新朋友	4
1.3	實力測驗（前測）	4
1.4	環境設定	4
2	8/14 (一) 下午：基本語法	5
2.1	輸入、輸出 (cin, cout)	5
2.2	四則運算 (+, -, *, /, mod, $x++$, $++x$, $x += 1$)	5
2.3	兩數交換(day1? Or day2?)	5
3	8/15(二) 上午：流程控制	6
3.1	分支	6
3.2	迴圈	6
4	8/15 (二) 下午：實驗室參訪	7
5	8/16(三) 上午：資料型態	8
5.1	printf() 格式輸出	8
5.2	整數與浮點數	8
5.3	溢位	8
5.4	ASCII 碼	8
6	8/16 (三) 下午：函數	8
6.1	call by value	8
6.2	用 * 畫圖（三角形、方形、聖誕樹）	8
6.3	遞迴	9
7	8/17(四) 上午：陣列&結構	10

7.1	陣列	10
7.2	排序	10
8	8/17 (四) 下午：類別函數庫 STL	10
8.1	容器	10
8.2	演算法	10

1 8/14(一) 上午：環境設定

1.1 行政事項

1. i-learning 登入測試

網頁版

下載 app 版



2. 藍芽點名
3. 宣傳保溫課程、說明校園參訪 & 校外參訪

1.2 認識新朋友

1. 老師、助教自我介紹
2. 同學自我介紹

1.3 實力測驗（前測）

1. 動手玩程式
冰雪奇緣遊戲（15 分鐘）

1.4 環境設定

1. Code::Blocks
安裝說明
2. 瘋狂程設
註冊

加課

桌機板

2 8/14 (一) 下午：基本語法

2.1 輸入、輸出 (cin, cout)

2.2 四則運算

1. 講解： A001:Hello World

(a) 題目說明：

請在命令視窗中印出 “Hello World ! ”。

(b) 解題思維：

本題直接使用 cout 或 printf 函數印出想要顯示的文字即可。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello World!";
8     return 0;
9 }
```

2. 練習： JT-01 : 我可以把程式學好

(a) 題目說明：

請印出字串”Programming is easy!”

(b) 解題思維：

本題直接使用 cout 或 printf 函數印出想要顯示的文字即可。

(c) 程式碼：

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Programming is easy!";
8     return 0;
9 }
```

3. 講解： F001: 兩數相加

(a) 題目說明：

輸入兩整數，輸出兩數之和。

(b) 解題思維：

- i. 先宣告兩個變數。

```
int a, b; // 宣告變數
```

- ii. 使用 cin 取得使用者輸入的兩個數字。

```
cin >> a >> b; // 取得輸入的值，存入 a 和 b
```

- iii. 將剛剛取得的兩個數字相加，並用 cout 輸出。

```
cout << a+b;
```

(c) 程式碼：

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a, b;
8     cin >> a >> b;
9     cout << a+b;
10    return 0;
11 }
```

4. 練習： F019: 長方形面積

(a) 題目說明：

輸入長和寬，輸出面積。

(b) 解題思維：

與上題解題思維大致相同，只是兩數相加變為兩數相乘。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a, b;
8     cin >> a >> b;
9     cout << a*b;
10    return 0;
11 }
```

5. 練習： G001: 長寬高算體積

(a) 題目說明：

輸入長方體的長寬高，輸出其體積。

(b) 解題思維：

與上題解題思維大致相同，變數改成三個，輸出將三數相乘。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a, b, c;
8     cin >> a >> b >> c;
9     cout << a*b*c;
10    return 0;
11 }
```

6. 講解： M90H011: 整數商餘

(a) 題目說明：

輸入兩整數 m 和 n，輸出 m 除以 n 之商及餘數。

(b) 解題思維：

- i. 先宣告兩個變數 m 和 n，再使用 scanf 取得使用者輸入的兩個整數。

```
int m, n;  
scanf("%d%d", &m, &n);
```

- ii. 將剛剛取得的整數 m 除以整數 n，並用 printf 輸出所除結果之商及餘數。

```
printf("\n%d / %d = %d", m, n, m / n);  
printf("\n%d mod %d = %d", m, n, m % n);
```

(c) 程式碼：

```
1 #include <stdio.h>  
2  
3 int main()  
4 {  
5     int m, n;  
6     scanf("%d%d", &m, &n);  
7     printf("\n%d / %d = %d", m, n, m / n);  
8     printf("\n%d mod %d = %d", m, n, m % n);  
9     return 0;  
10 }
```

7. 練習：使用 printf 完成下列兩題

(a) F001 : 兩數相加

(b) F019 : 長方形面積

3 8/15(二) 上午：流程控制－分支

3.1 分支

3.1.1 if

1. 講解： JA-001：兩數排序

(a) 題目說明：

輸入 a 和 b 兩個數，將其依小到大的順序印出來。

(b) 解題思維：

- i. 用 if 進行判斷，如果 a 大於 b，則兩數交換。
- ii. 交換兩數 a 和 b，在 C++ 中可以直接使用 swap 函數，如果是在 C 裡面，則常用的方法是宣告另一個暫存變數 t，然後使用以下敘述：

t=a; x=b; b=t;

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int a, b;
8     cin >> a >> b;
9     if (a>b) swap(a, b);
10    cout << a << " " << b;
11    return 0;
12 }
```

2. 講解： A016：三數排序

(a) 題目說明：

輸入三個正整數 a、b、c，將 a、b、c 從小排到大並輸出。

(b) 解題思維：

- i. 先宣告三整數 a, b, c 並輸入其值。

```
int a, b, c;
cin >> a >> b >> c;
```

- ii. 三數排序時，先比 a 和 b，如果 $a>b$ 則交換兩個數，使 $a<b$ ，之後再比 b 和 c，使 $b<c$ ，此時 c 為最大值。最後再比較和調整一次 a 和 b 即可。

```
if (a>b) swap(a, b);
if (b>c) swap(b, c);
if (a>b) swap(a, b);
```

- iii. 交換兩數 x 和 y，在 C++ 中可以直接使用 swap 函數，如果是在 C 裡面，則常用的方法是宣告另一個暫存變數 t，然後使用以下敘述：

```
t=x; x=y; y=t;
```

(c) 程式碼：

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a, b, c;
7     cin >> a >> b >> c;
8     if (a>b) swap(a, b);
9     if (b>c) swap(b, c);
10    if (a>b) swap(a, b);
11    cout << a << " " << b << " " << c;
12    return 0;
13 }
```

3. 練習： JA-002：四數排序

(a) 題目說明：

輸入 a,b,c,d 四個數，將其依小到大的順序印出來。

(b) 解題思維：

- i. 將最大的整數置換到變數 d。
- ii. 對 a,b,c 由小到大進行三數排序。

(c) 程式碼：

```
1 #include <iostream>
2
```

```

3   using namespace std;
4
5   int main()
6   {
7       int a, b, c, d;
8       cin >> a >> b >> c >> d;
9       if (a>b) swap(a, b);
10      if (b>c) swap(b, c);
11      if (c>d) swap(c, d);
12      if (a>b) swap(a, b);
13      if (b>c) swap(b, c);
14      if (a>b) swap(a, b);
15      cout << a << " " << b << " " << c << " " << d;
16      return 0;
17 }

```

3.1.2 if else

1. 講解： A025: 判斷閏年

(a) 題目說明：

輸入西元年，如果該年是閏年，則輸出 Yes，若該年不是閏年，則輸出 No。(閏年的定義為，四年一閏，逢百不閏，逢四百又閏。例如西元 1004 年為閏年，西元 1100 年不是閏年，西元 1600 年是閏年)

(b) 解題思維：

宣告年份 year，接著再按照閏年的規則判斷是否為閏年就好。

(c) 程式碼：

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int year;
7     cin >> year;
8     if (year%400==0) cout << "Yes";
9     else if (year%100==0) cout << "No";
10    else if (year%4==0) cout << "Yes";

```

```
11     else cout << "No";
12
13 }
```

2. 講解： F021: 奇偶數

(a) 題目說明：

輸入一整數，輸出其奇偶性。

(b) 解題思維：

判斷整數 n 是否為奇數的方法，可求其除以 2 的餘數，若非 0 即為奇數。

```
if (n%2) { /* n 為奇數 */ }
```

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int n;
8     cin >> n;
9     if (n%2) cout << "odd";
10    else cout << "even";
11    return 0;
12 }
```

3. 練習： A006: 輸入之正負零

(a) 題目說明：

輸入一整數 N，如果 N 大於 0，則輸出 N>0，如果 N 等於 0，則輸出 N=0，如果 N 小於 0，則輸出 N<0。

(b) 解題思維：

先宣告整數 N，輸入值之後再判斷是大於零、等於零還是小於零，分別印出相對的敘述。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
```

```
3
4     int main()
5     {
6         int n;
7         cin >> n;
8         if (n>0) cout << "N>0";
9         else if (n==0) cout << "N=0";
10        else cout << "N<0";
11        return 0;
12    }
```

3.1.3 switch (若有多餘時間)

4 8/15 (二) 下午：實驗室參訪

5 8/16(三) 上午：流程控制—迴圈

5.1 回圈

5.1.1 while

1. 講解： JA-003： $1+2+3+\dots+100$

(a) 題目說明：

使用 while 迴圈計算 $1+2+3+\dots+100$

(b) 解題思維：

- i. 宣告一個初始值為零的變數 $sum=0$ ，準備進行累加。
- ii. 使用 while 迴圈，當符合 while 的執行條件 ($n < 100$) 時，會重複執行累加計算，累加完之後會更新 n 的值，當 n 不滿足執行條件時，結束迴圈。

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int n=1, sum=0;
8     while (n<=100) {
9         sum += n; // 累加
10        n++; // 更新 n
11    }
12    cout << sum;
13    return 0;
14 }
```

2. 練習：JA-004： $1+3+5+\dots+99$

(a) 題目說明：

使用 while 迴圈計算 $1+3+5+\dots+99$

(b) 解題思維：

使用 while 迴圈進行累加，只要 ($n < 100$) 就對 sum 進行累加，每次累加完後， n 加上 2。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int n=1, sum=0;
8     while (n<100) {
9         sum += n;
10        n += 2;
11    }
12    cout << sum;
13    return 0;
14 }
```

5.1.2 for

1. 講解： JA-005： $1+2+3+\dots+100$

(a) 題目說明：

使用 for 迴圈計算 $1+2+3+\dots+100$

(b) 解題思維：

for 迴圈的用法是 for(起始值；條件式；更新值)，本題的 for 迴圈寫法如下：

```
for (int i=1; i<=100; i++) sum += i;
```

意思是： i 的起始值是 1，當 $i \leq 100$ 這個條件成立時的時候，會重複執行累加 ($sum += i;$)，並在累加完後更新 i 的值 ($i++$)。此迴圈總共回會複執行 100 次累加的計算。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
```

```

6   {
7       int sum=0;
8       for (int i=1; i<=100; i++) sum += i;
9       cout << sum;
10      return 0;
11  }

```

2. 練習： JA-006 : $1+3+5+\dots+99$

(a) 題目說明：

使用 for 迴圈計算 $1+3+5+\dots+99$

(b) 解題思維：

使用 for 迴圈進行累加，只要 ($n < 100$) 就對 sum 進行累加，n 每次更新都加 2。

(c) 程式碼：

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int sum=0;
8     for (int i=1; i<100; i+=2) sum += i;
9     cout << sum;
10    return 0;
11 }

```

5.1.3 應用題

1. 講解： JT-04：印三角形函數

(a) 題目說明：

輸入 N，印出 N 列的星號 (*)，其中第 I 列有 I 個星，如執行範例所示。

範例輸入：5

範例輸出：

```
Input row number : 5\n*\n**\n***\n****\n*****
```

(b) 解題思維：

這題需要使用雙重 for 迴圈。第一個迴圈計算現在要印第幾列，第二個迴圈計算要印幾個星。

(c) 程式碼：

```
1 #include <stdio.h>\n2\n3 int main()\n4 {\n5     int n, i, j;\n6     printf("Input row number : \n");\n7     scanf("%d", &n);\n8     for (i=1; i<=n; i++) { // 迴圈 1：第 i 列\n9         for(j=0; j<i; j++) { // 迴圈 2：印 i 個星\n10            printf("*");\n11        }\n12        printf("\n");\n13    }\n14    return 0;\n15 }
```

2. 練習：JP-010-2：印倒三角形 (無空白)

(a) 題目說明：

輸入正整數 $n \leq 20$ ，輸出一個 n 層的倒三角形。

範例輸入：5

範例輸出：

```

*****\r\n
****\r\n
***\r\n
**\r\n
*\r\n

```

(b) 解題思維：

印倒三角形時，第一列有 n 個星，下一列有 n-1 個星，以此列推，每換一列就少一個星，所以這題是要使用 for 迴圈來倒數。

(c) 程式碼：

```

1 #include <stdio>
2
3 int main()
4 {
5     int n, i, j;
6     scanf("%d", &n);
7     for (i=n; i>0; i--) { // 迴圈 1：計算此列有 i 個星
8         for(j=i; j>0; j--) { // 迴圈 2：印出 i 個星
9             printf("*");
10        }
11        printf("\n");
12    }
13    return 0;
14 }

```

3. 挑戰： JT-40 印等腰三角形

(a) 題目說明：

輸入 N，印出一個 N 列的等腰三角形，其中第 I 列有 $2*I-1$ 個 #，如程式範例結果所示。

範例輸入：5

範例輸出：

```

Input = 5
      #
      ##
      ###
      #####
      ######
```

(b) 解題思維：

- i. 因為要印出 N 列，所以先寫一個執行 N 次的 for 迴圈，用變數 row 來計算現在是第幾列。
- ii. 第 row 列要印出 (n-row) 個“空白”，及 (2*row-1) 個“#”，所以分別用兩個 for 迴圈印“空白”及“#”。

(c) 程式碼：

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int i, row, n;
6     scanf("%d", &n);
7     printf("Input = %d\n", n);
8     for (row=1; row<=n; row++) {
9         for (i=0; i<n-row; i++) printf(" ");
10        for (i=0; i<2*row-1; i++) printf("#");
11        printf("\n");
12    }
13    return 0;
14 }
```

5.1.4 do while (若有多餘時間)

6 8/16 (三) 下午：函數

6.1 printf() 格式輸出

1. 講解：JB-02：九九乘法表

(a) 題目說明：

印出如輸出之九九乘法表。

```
1*1= 1  2*1= 2  3*1= 3  4*1= 4  5*1= 5  6*1= 6  7*1= 7  8*1= 8  9*1= 9  
1*2= 2  2*2= 4  3*2= 6  4*2= 8  5*2=10  6*2=12  7*2=14  8*2=16  9*2=18  
1*3= 3  2*3= 6  3*3= 9  4*3=12  5*3=15  6*3=18  7*3=21  8*3=24  9*3=27  
1*4= 4  2*4= 8  3*4=12  4*4=16  5*4=20  6*4=24  7*4=28  8*4=32  9*4=36  
1*5= 5  2*5=10  3*5=15  4*5=20  5*5=25  6*5=30  7*5=35  8*5=40  9*5=45  
1*6= 6  2*6=12  3*6=18  4*6=24  5*6=30  6*6=36  7*6=42  8*6=48  9*6=54  
1*7= 7  2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49  8*7=56  9*7=63  
1*8= 8  2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64  9*8=72  
1*9= 9  2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81
```

(b) 解題思維：

- i. 先看第一列，會變動的數是“被乘數”，而且變動是有規律的 1,2,3...,9，所以我們寫一個會執行 9 次的 for 迴圈，讓變數 j 從 1 跑到 9。變數 j 就是要輸出的“被乘數”。

```
1*1= 1  2*1= 2  3*1= 3  4*1= 4  5*1= 5  6*1= 6  7*1= 7  8*1= 8  9*1= 9
```

- ii. 再來觀察“乘數”，同一列的乘數是固定的，乘數隨著列改變，也就是說第 i 列的乘數是 i。總共有 9 列，所以要寫一個會執行 9 次的 for 迴圈。變數 i 就是要輸出的“乘數”。
- iii. “乘積”只要將 i, j 相乘就可以了。
- iv. 這題使用 printf() 格式輸出比較容易。“%2d”表示輸出時，會給這個整數兩給位數，當輸出的整數只有個位數的時候，十位數的位置會自動補上“空格”。

(c) 程式碼：

```
1 #include <csstdio>  
2  
3 int main()  
{  
4     for (int i=1; i<=9; i++) { // 第 i 列的乘數是 i  
5         for(int j=1; j<=9; j++) { // 每一列的被乘數 j 都從 1~9  
6             printf("%d*%d=%2d ", j, i, i*j);  
7         }  
8         printf("\n");  
9     }  
10    return 0;  
11}  
12}
```

2. 練習： JA-007：九九乘法表（兩排）

(a) 題目說明：

印出九九乘法表，如輸出結果所示。

```
2*1= 2  3*1= 3  4*1= 4  5*1= 5  \r\n
2*2= 4  3*2= 6  4*2= 8  5*2=10 \r\n
2*3= 6  3*3= 9  4*3=12 5*3=15 \r\n
2*4= 8  3*4=12 4*4=16 5*4=20 \r\n
2*5=10 3*5=15 4*5=20 5*5=25 \r\n
2*6=12 3*6=18 4*6=24 5*6=30 \r\n
2*7=14 3*7=21 4*7=28 5*7=35 \r\n
2*8=16 3*8=24 4*8=32 5*8=40 \r\n
2*9=18 3*9=27 4*9=36 5*9=45 \r\n
\r\n
6*1= 6  7*1= 7  8*1= 8  9*1= 9  \r\n
6*2=12 7*2=14 8*2=16 9*2=18 \r\n
6*3=18 7*3=21 8*3=24 9*3=27 \r\n
6*4=24 7*4=28 8*4=32 9*4=36 \r\n
6*5=30 7*5=35 8*5=40 9*5=45 \r\n
6*6=36 7*6=42 8*6=48 9*6=54 \r\n
6*7=42 7*7=49 8*7=56 9*7=63 \r\n
6*8=48 7*8=56 8*8=64 9*8=72 \r\n
6*9=54 7*9=63 8*9=72 9*9=81 \r\n
\r\n
```

(b) 解題思維：

可以想成輸出兩個大群組的九九乘法表，當輸出第 r ($r=0, 1$) 個群組時
“被乘數” = $j + r \times 4$. ($j=2, 3, 4, 5$)。

(c) 程式碼：

```
1 #include <csdio>
2
3 int main()
4 {
5     for (int r=0; r<2; r++) { // 兩個群組
6         for (int i=1; i<=9; i++) { // 第 i 列的乘數是 i
7             for(int j=2; j<=5; j++) { // 被乘數 = j+r*4
8                 printf("%d*%d=%2d ", j+r*4, i, i*(j+r*4));
9             }
10            printf("\n");
11        }
12        printf("\n");
13    }
14    return 0;
15 }
```

6.2 自訂函數

1. 講解： JT-04 印三角形函數

(a) 題目說明：

輸入 N，印出 N 列的星號 (*)，其中第 I 列有 I 個星，如執行範例所示。

範例輸入：5

範例輸出：

```
Input row number : 5\n*\n**\n***\n****\n*****
```

(b) 解題思維：

- i. 自己定義畫三角形的函數，使用這個函數時，需要輸入參數 n，這樣函數才知道三角形有幾列。
- ii. 將畫三角形的程式碼寫進函式裡，主程式 main 裡面只需要呼叫函數即可印出三角形。

(c) 程式碼：

```
1 #include <stdio.h>\n2\n3 void triangle(int n);\n4\n5 int main()\n6 {\n7     int n;\n8     printf("Input row number : \n");\n9     scanf("%d", &n);\n10    triangle(n);\n11    return 0;\n12 }\n13\n14 void triangle(int n)\n15 {
```

```

16     int i, j;
17     for (i=1; i<=n; i++) {
18         for (j=0; j<i; j++) { printf("*"); }
19         printf("\n");
20     }
21 }
```

2. 練習：JP-010-2：印倒三角形（無空白）

(a) 題目說明：

輸入正整數 $n \leq 20$ ，輸出一個 n 層的倒三角形。

範例輸入：5

範例輸出：

```

*****\r\n
****\r\n
***\r\n
**\r\n
*\r\n
```

(b) 解題思維：

- i. 自己定義畫三角形的函數，使用這個函數時，需要輸入參數 n ，這樣函數才知道三角形有幾列。
- ii. 將畫倒三角形的程式碼寫進函式裡，主程式 main 裡面只需要呼叫函數即可印出三角形。

(c) 程式碼：

```

1 #include <cstdio>
2
3 void plot(int n);
4
5 int main()
6 {
7     int n;
8     scanf("%d", &n);
9     plot(n);
10    return 0;
11 }
```

```

12
13     void plot(int n)
14 {
15     int i, j;
16     for (i=n; i>0; i--) {
17         for(j=i; j>0; j--) {
18             printf("*");
19         }
20         printf("\n");
21     }
22 }
```

3. 練習： JT-40：印等腰三角形

(a) 題目說明：

輸入 N，印出一個 N 列的等腰三角形，其中第 I 列有 $2*I-1$ 個 #，如程式範例結果所示。

範例輸入：5

範例輸出：

```

Input = 5↵\r\n
#↵\r\n
##↵\r\n
###↵\r\n
####↵\r\n
#####↵\r\n
######↵\r\n
```

(b) 解題思維：

- i. 自己定義畫三角形的函數，使用這個函數時，需要輸入參數 n，這樣函數才知道三角形有幾列。
- ii. 將畫等腰三角形的程式碼寫進函式裡，主程式 main 裡面只需要呼叫函數即可印出三角形。

(c) 程式碼：

```

1 #include <stdio.h>
2
3 void plot(int n);
```

```

4
5     int main()
6 {
7     int n;
8     scanf("%d", &n);
9     printf("Input = %d\n", n);
10    plot(n);

11
12    return 0;
13 }

14
15 void plot(int n)
16 {
17     int i, row;
18     for (row=1; row<=n; row++) {
19         for (i=0; i<n-row; i++) printf(" ");
20         for (i=0; i<2*row-1; i++) printf("#");
21         printf("\n");
22     }
23 }
```

4. 挑戰： JT61:Game Over

(a) 題目說明：

輸入整數 m 和 n, 輸出以 # 排成框, 中間為 Game Over 之圖案, 其中 m 為 G 之前和 r 之後與邊界的空格數,n 為文字與上下邊界的空格數。例如輸入 2 1, 則輸出為

```

#####
#      #
#  Game Over  #
#      #
#####
```

(b) 解題思維：

- 第 1 列有 $11+2m$ 個 #。
- 接下來 n 列頭尾是 #，中間有 $9+2m$ 個空白。
- 再下一列是 # 加 m 個空白，加 Game Over，加 m 個空白和 #。

iv. 接下來 n 列頭尾是 #，中間有 $9+2m$ 個空白。

v. 最後一列有 $11+2m$ 個 #。

(c) 程式碼：

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int m, n;
8     cin >> m >> n;
9     for (int i=0; i<11+2*m; i++) cout << "#"; // Row 1
10    cout << endl;
11    for (int r=0; r<n; r++) { // Next n rows
12        cout << "#";
13        for (int i=0; i<9+2*m; i++) cout << " ";
14        cout << "#" << endl;
15    }
16    cout << "#"; // Middle row
17    for (int i=0; i<m; i++) cout << " ";
18    cout << "Game Over";
19    for (int i=0; i<m; i++) cout << " ";
20    cout << "#" << endl;
21    for (int r=0; r<n; r++) { // Next n rows
22        cout << "#";
23        for (int i=0; i<9+2*m; i++) cout << " ";
24        cout << "#" << endl;
25    }
26    for (int i=0; i<11+2*m; i++) cout << "#"; // Last row
27    cout << endl;
28    return 0;
29 }
```

7 8/17(四) 上午：遞迴

7.1 遞迴

1. 講解：JA-008：遞迴解 $1+2+\dots+n$

(a) 題目說明：

使用遞迴方式算出 $1+2+\dots+n$

(b) 解題思維：

假設 $f(n) = 1 + 2 + \dots + n$ ，則遞迴的計算方法為 $f(n) = n + f(n - 1)$ 。

(c) 程式碼：

```
1 #include <csdio>
2
3 int f(int n);
4
5 int main()
6 {
7     int n;
8     scanf("%d", &n);
9     printf("%d", f(n));
10    return 0;
11 }
12
13 int f(int n)
14 {
15     if (n==1) return 1;
16     return n + f(n-1);
17 }
```

2. 練習：A059：遞迴計算 n 階乘

(a) 題目說明：

輸入一正整數 N ，輸出 $N!$ 。其中 $N! = 1 \times 2 \times 3 \times \dots \times N$

(b) 解題思維：

假設函數 $fact(n) = n!$ ，其遞迴的計算方式為 $fact(n) = n \times fact(n - 1)$ 。

(c) 程式碼：

```

1 #include<iostream>
2 using namespace std;
3
4 int fact(int n);
5
6 int main()
7 {
8     int n;
9     cin >> n;
10    cout << fact(n);
11    return 0;
12 }
13
14 int fact(int n)
15 {
16     if (n) return n * fact(n-1);
17     else return 1;
18 }
```

3. 講解：A029：費式數列

(a) 題目說明：

費氏數列定義如下 $f(0) = 0, f(1) = 1, f(n) = f(n - 1) + f(n - 2)$ 。題目是從螢幕輸入一個正整數 n , 輸出 $f(n)$ 。

(b) 解題思維：

- 本題可用遞迴或非遞迴方式計算。
- 因為程式簡明易了，可直接觀看程式碼尋求理解。

(c) 程式碼：

```

1 #include <stdio.h>
2
3 int f(int n);
4
5 int main()
6 {
7     int n;
8     scanf("%d", &n);
```

```

9     printf("%d", f(n));
10    return 0;
11 }
12
13 int f(int n)
14 {
15     if (n<2) return n;
16     return f(n-1)+f(n-2);
17 }
```

4. 挑戰：JA-009：爬樓梯有幾種爬法

(a) 題目說明：

小明爬樓梯，已知要爬的梯數有 N 階，但小明一次可以爬 $1 \sim 3$ 階，請問總共有幾種爬法？

(b) 解題思維：

當 $n=1, 2, 3$ 時，分別有 1, 2, 4 種爬法，當 $n>3$ 時，爬樓梯的方法為 $f(n) = f(n - 1) + f(n - 2) + f(n - 3)$ 。

(c) 程式碼：

```

1 #include <stdio.h>
2
3 int f(int n);
4
5 int main()
6 {
7     int n;
8     scanf("%d", &n);
9     printf("%d", f(n));
10    return 0;
11 }
12
13 int f(int n)
14 {
15     if (n==1) return 1;
16     if (n==2) return 2;
17     if (n==3) return 4;
18     return f(n-1)+f(n-2)+f(n-3);
```

5. 講解：JB-04：河內塔

(a) 題目說明：

依課堂上講解之河內塔規則，從柱 1 移到柱 3，柱 2 為輔助。輸入環的個數 n，輸出所有移動過程。

(b) 解題思維：

河內塔的解法如下：

- i. 當只有 1 個環的時候，直接把環搬到目標柱子上。
- ii. 當有 n 個環的時候
 - A. 先將 (n-1) 層的河內塔搬到輔助的柱子上。
 - B. 接著將第 n 個環搬到目標柱子上。
 - C. 最後，再將 (n-1) 層的河內塔從輔助的柱子搬到目標柱子上。

(c) 程式碼：

```

1 #include <iostream>
2
3 using namespace std;
4
5 void hanoi(int n, int from, int to, int buf);
6
7 int main()
8 {
9     int n;
10    cin >> n;
11    hanoi(n, 1, 3, 2);
12    return 0;
13 }
14
15 void hanoi(int n, int from, int to, int buf)
16 {
17     if (n==1) {
18         cout << from << " => " << to << endl;
19     } else {
20         hanoi(n-1, from, buf, to);
21         cout << from << " => " << to << endl;

```

```
22         hanoi(n-1, buf, to, from);  
23     }  
24 }
```

8 8/17(四) 下午：陣列

8.1 陣列

1. 講解：A030：百數反印

(a) 題目說明：

輸入 100 個正整數，反向印出此 100 個數。

(b) 解題思維：

本題使用陣列儲存 100 個數，再反向印出即可，是很基本的題目。

(c) 程式碼：

```
1 #include<iostream>  
2 using namespace std;  
3 int main()  
4 {  
5     int data[100];  
6     for (int i=0; i<100; i++) cin >> data[i];  
7     for (int i=99; i>=0; i--) cout << " " << data[i];  
8     return 0;  
9 }
```

2. 講解：JT-30：排序

(a) 題目說明：

輸入 N 及 N 個數 ($N < 100$)，將 N 個數從小到大印出來。

(b) 解題思維：

本題是練習排序的演算法。基本上排序的演算法很多，以下程式使用氣泡排序法，這也是最基本的排序演算法之一。在程式中，i 的範圍可以從 0 到 $n-2$ ，或則倒過來從 $n-1$ 到 1 也可以，基本上就是要執行 $n-1$ 輪的意思，但是 i 的範圍寫法不同，j 的上限寫法也跟著（有）一些變化，這是在閱讀參考連結時，應注意的地方。

(c) 程式碼：

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i, j, t, n, a[100];
6     scanf("%d", &n);
7     for (i=0; i<n; i++) scanf("%d", a+i);
8     for (i=n-1; i>0; i--) {
9         for (j=0; j<i; j++) {
10             if (a[j]>a[j+1]) {
11                 t=a[j];
12                 a[j]=a[j+1];
13                 a[j+1]=t;
14             }
15         }
16     }
17     printf("%d", a[0]);
18     for (i=1; i<n; i++) printf(" %d", a[i]);
19     return 0;
20 }
```

9 附錄一、安裝與設定

在學習程式語言之前，要先設定好程式的開發環境。這個章節會教同學如何安裝及設定開發環境。首先，說明如何安裝兩套軟體「Code::Blocks」以及「瘋狂程設」；再來，說明如何註冊瘋狂程設的帳號；最後，說明如何使用瘋狂程設。

9.1 安裝 CodeBlocks

在 Google 搜尋輸入關鍵字「codeblocks」，選擇搜尋結果「Download binary - Code::Blocks」。也可以直接輸入以下網址 <http://www.codeblocks.org/downloads/26>，參考??。

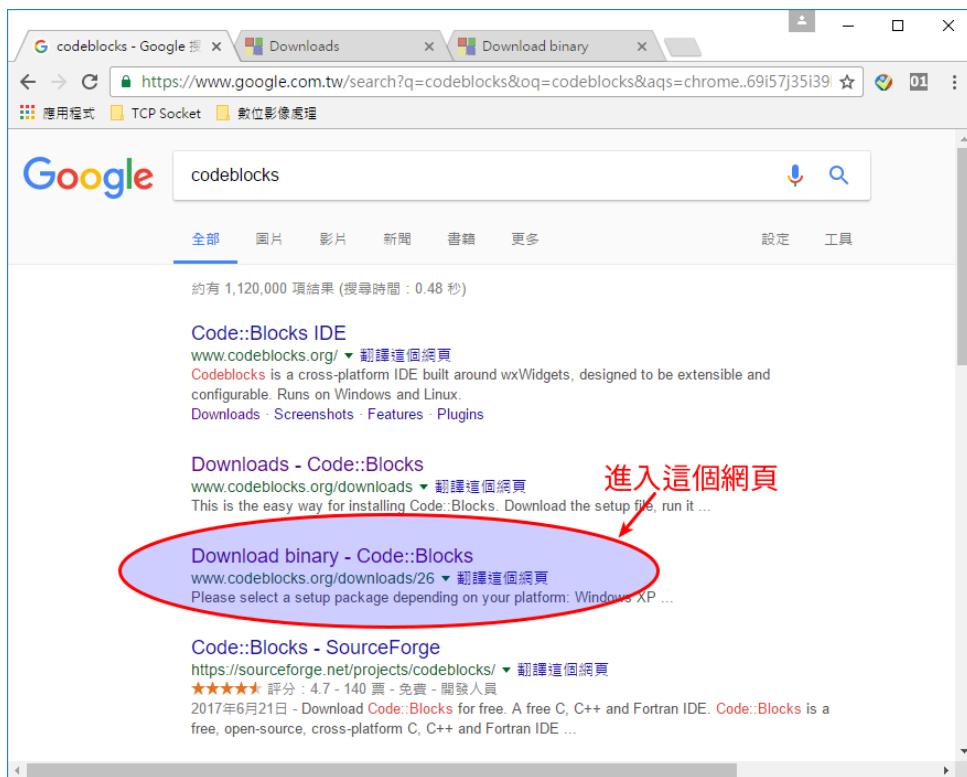


圖 1: codeblocks 搜尋結果

進入官網下載頁面後，選擇包含 mingw-setup 的檔案，點選該檔案右邊的連結下載，參考??。

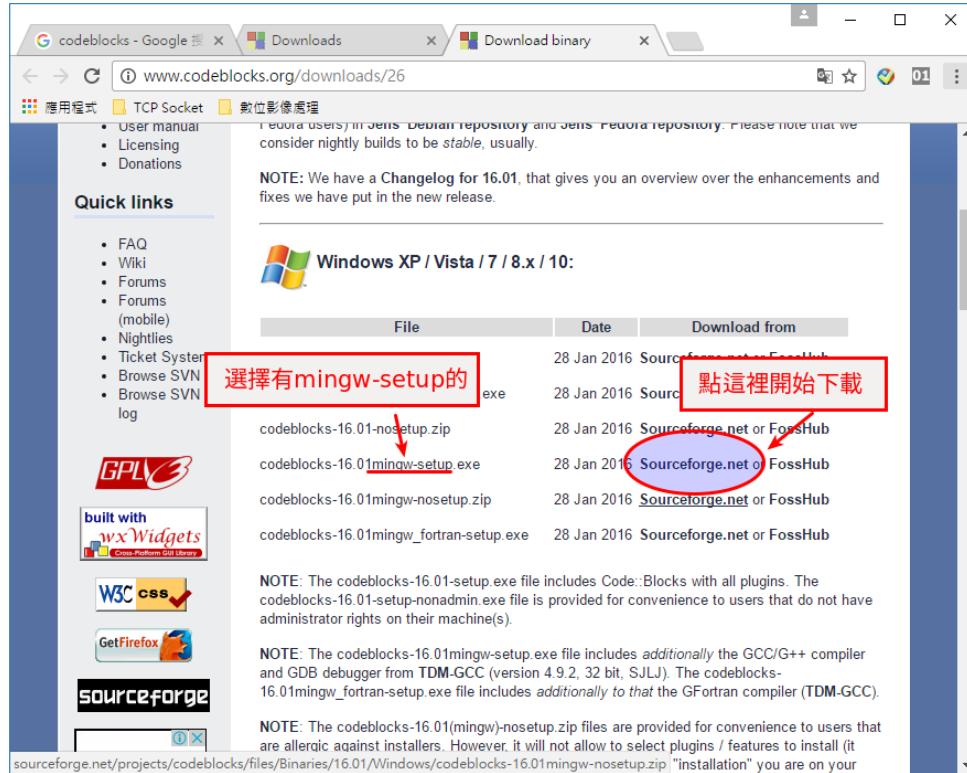


圖 2: Download binary 頁面

找到剛剛下載的安裝檔 codeblocks-16.01mingw-setup.exe，點兩下開啟安裝精靈，參考??。

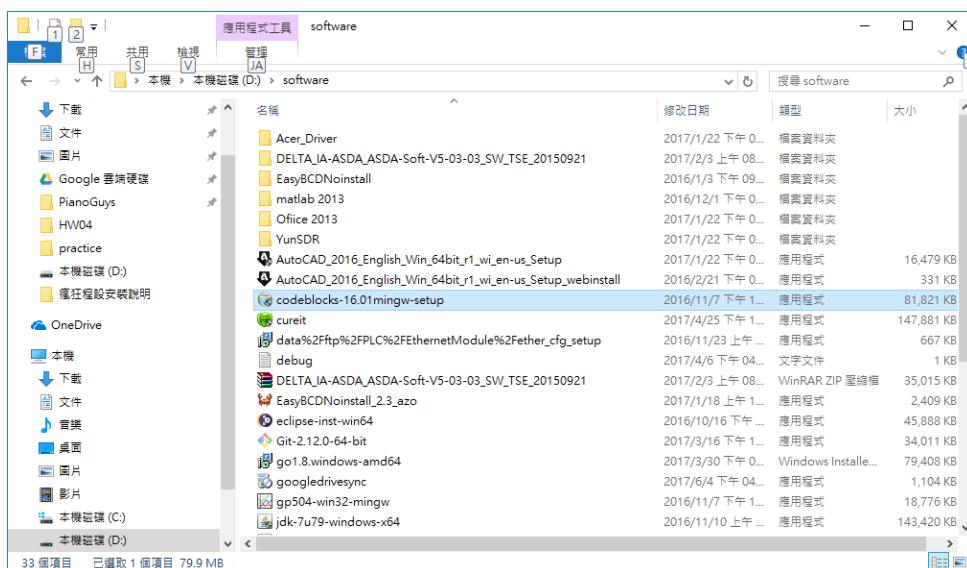


圖 3: 點兩下開啟 CodeBlocks 安裝檔

請不要修改安裝精靈的任何設定，只要一直點「Next」就好。絕對不可以更改檔案路徑，參考??。

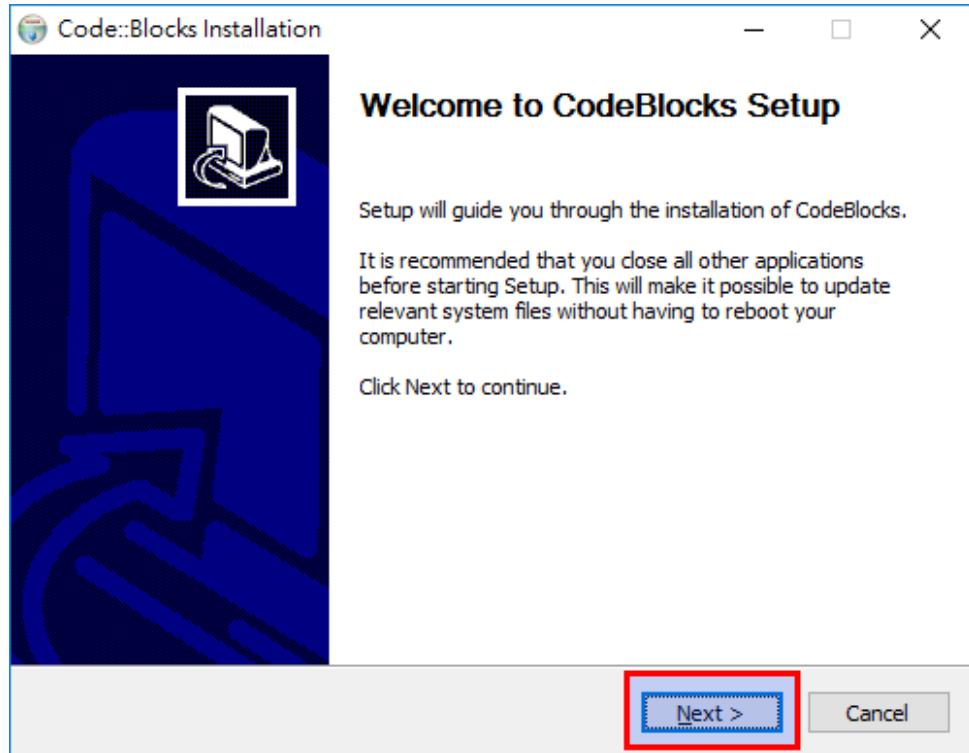


圖 4: 啟動安裝精靈，點「Next」

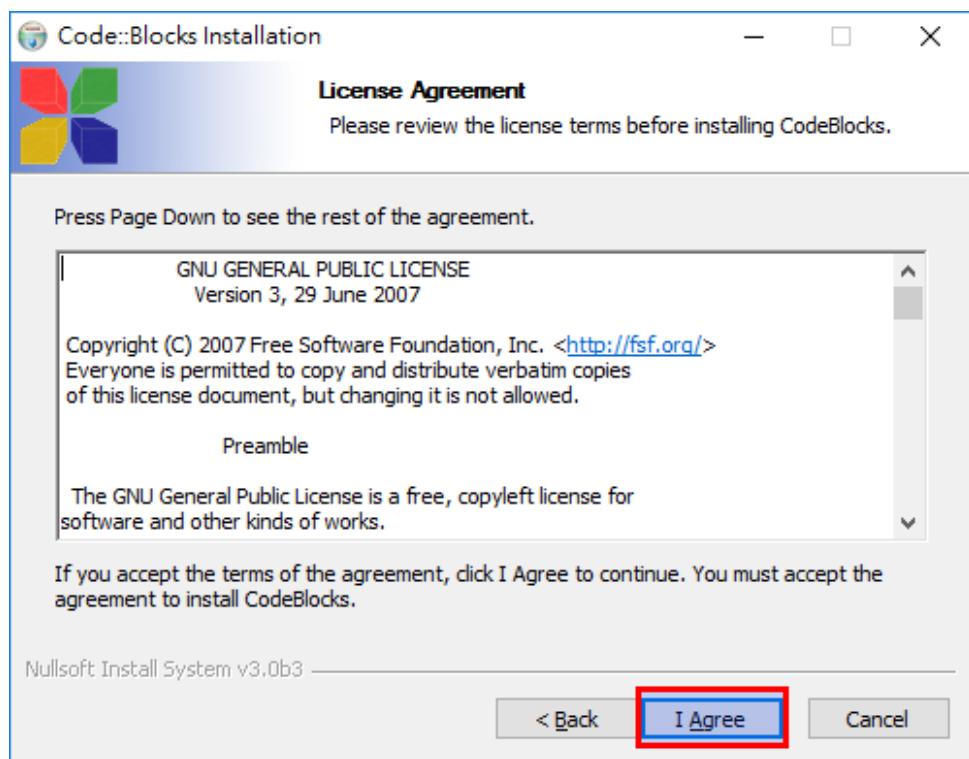


圖 5: 點「I Agree」

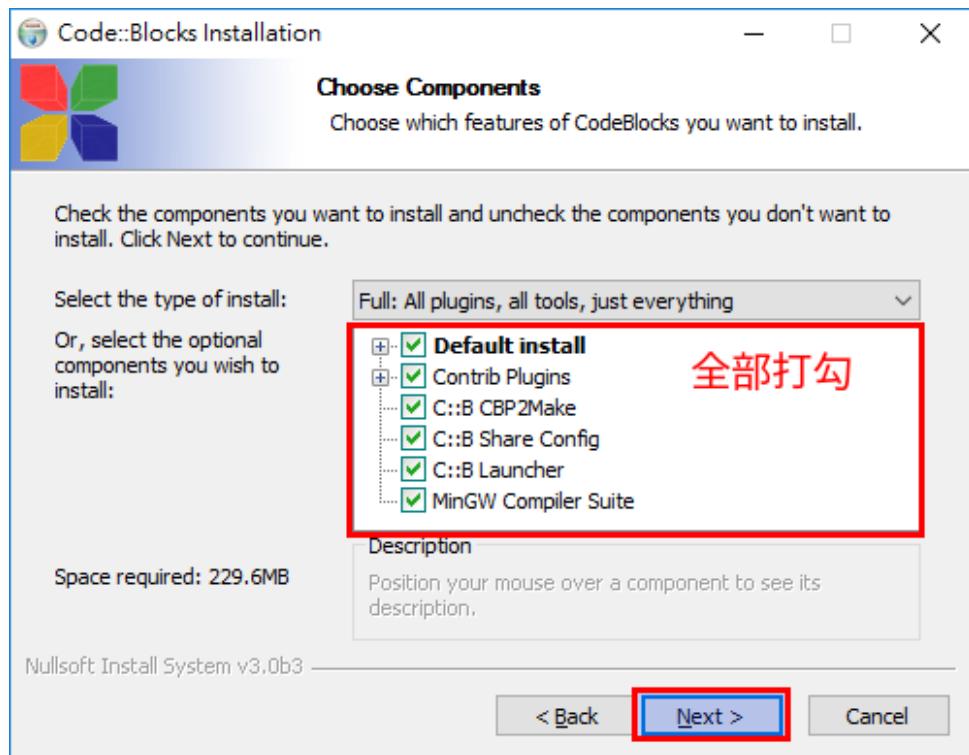


圖 6: 全部打勾，點「Next」

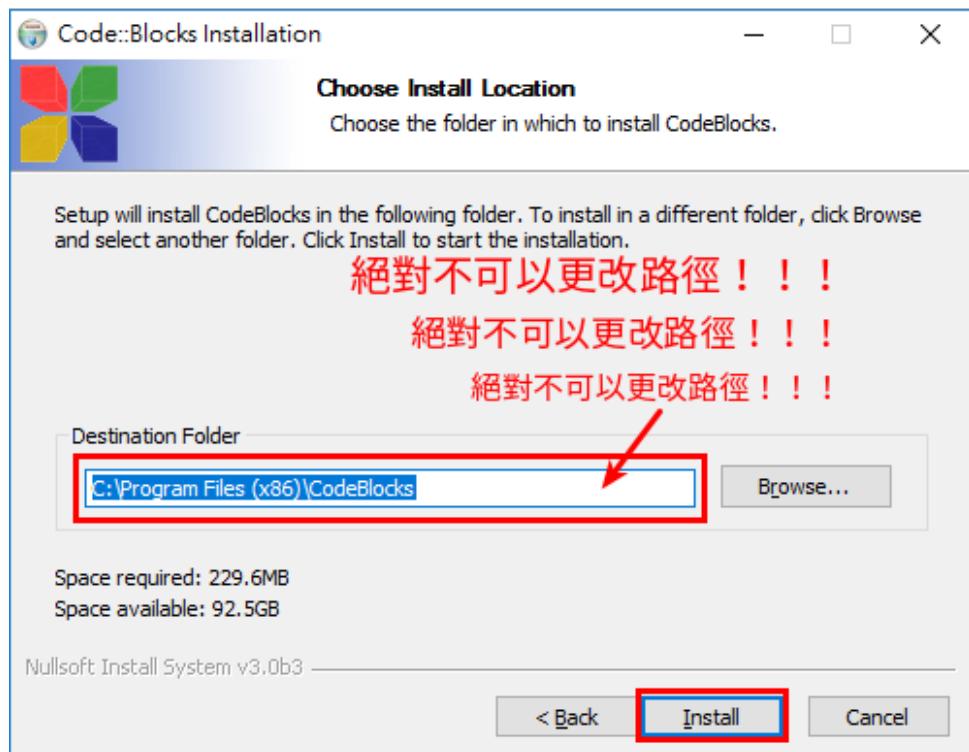


圖 7: 直接點「Install」，絕對不可以更改路徑

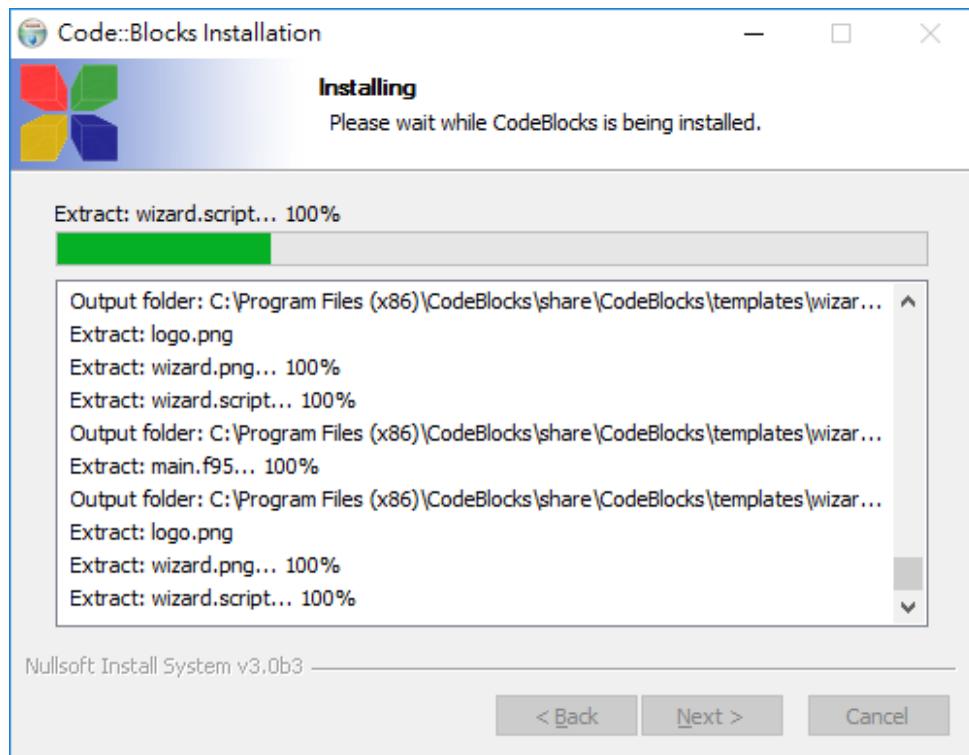


圖 8: 正在安裝，請耐心等待

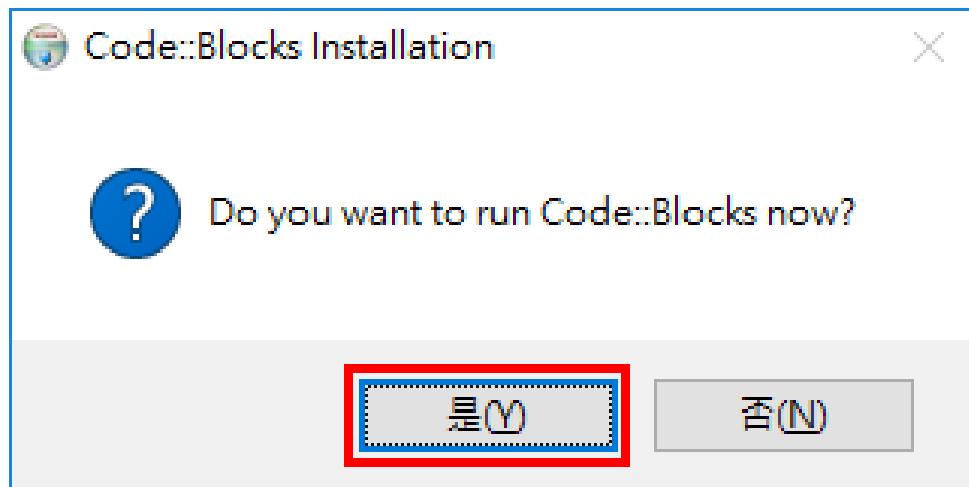


圖 9: 點「是 (Y)」，起動 codeBlocks

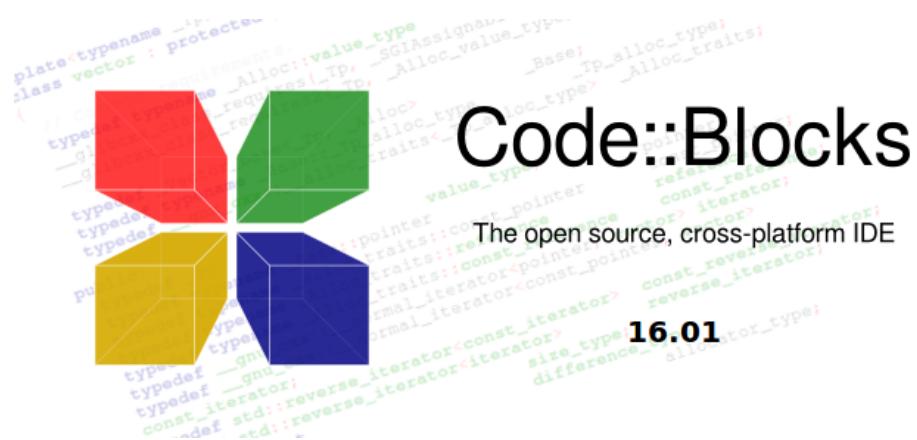


圖 10: 正在開啟 Codeblocks

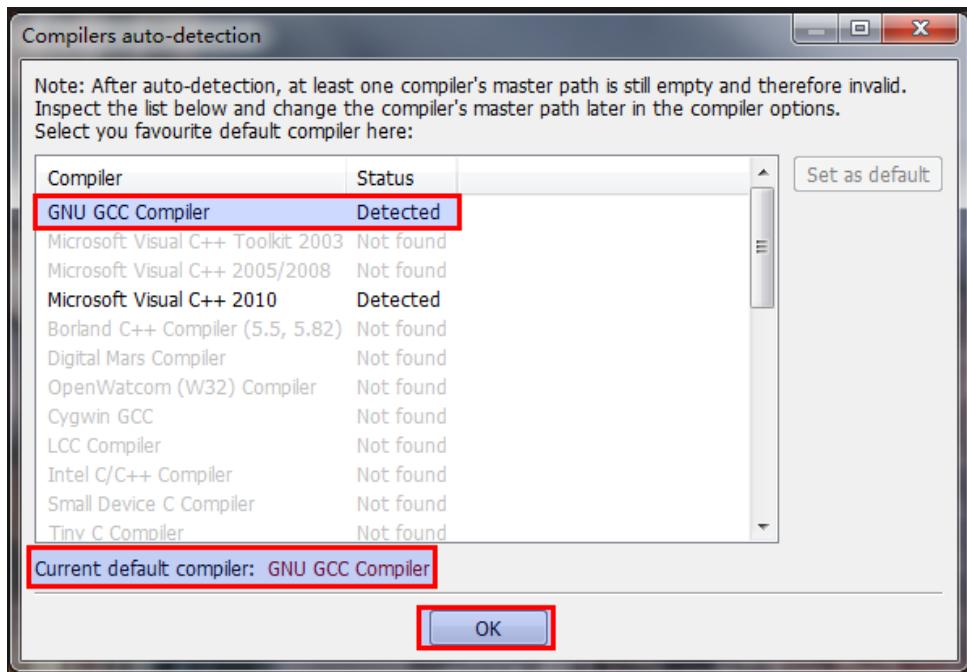


圖 11: 確定有偵測到 GNU GCC Compiler，點「OK」

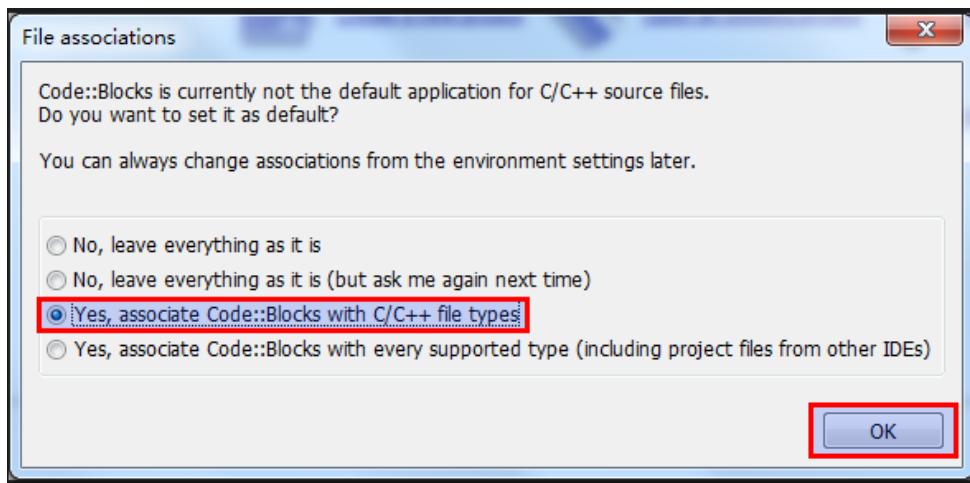


圖 12: 選第三個，將 CodeBlocks 設為開起程式碼的預設程式

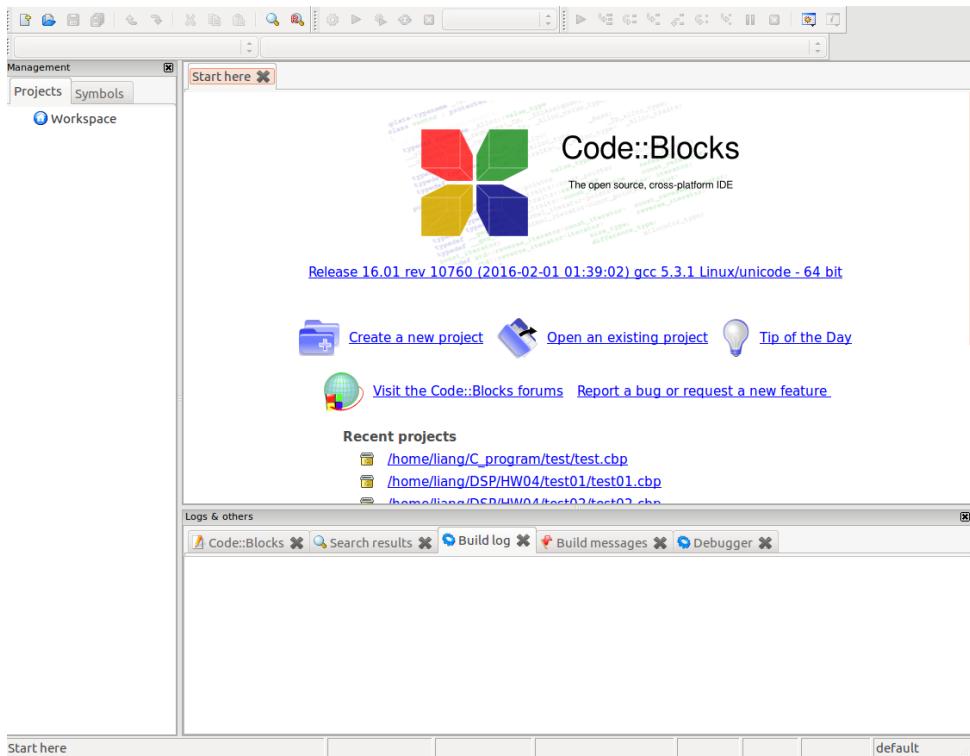


圖 13: 成功開啟 CodeBlocks

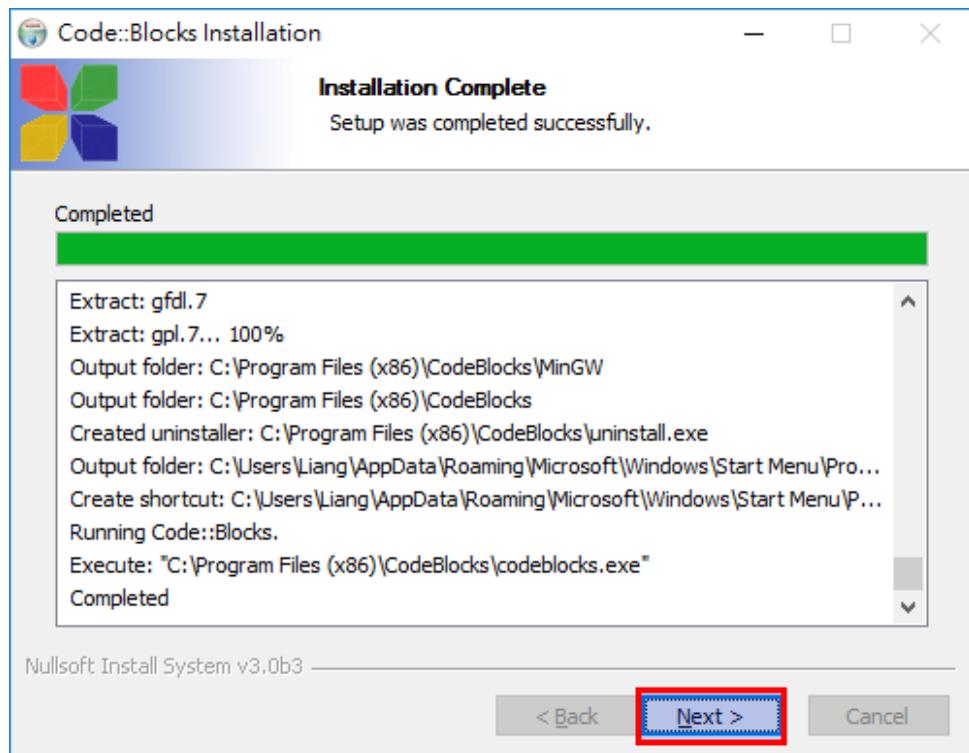


圖 14: 回到安裝精靈，點「Next」

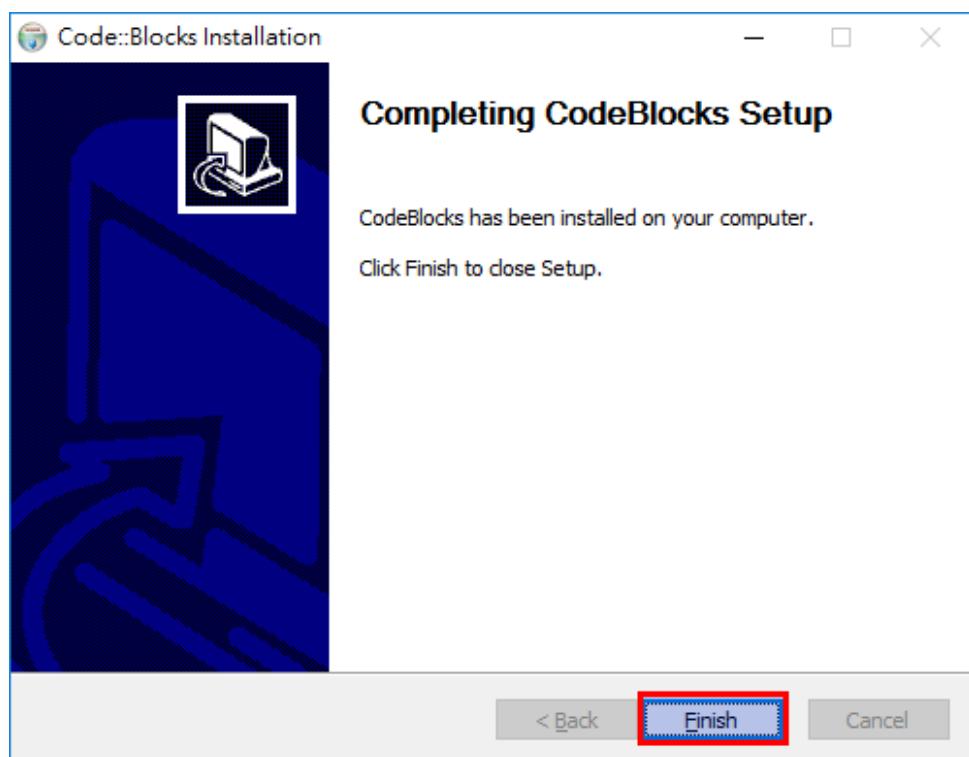


圖 15: 點「Finish」，關閉安裝精靈

9.2 安裝「瘋狂程設」

在 google 瀏覽器上搜尋關鍵字「瘋狂程設」，進入「瘋狂程設：自動閱卷的程式設計機上考試題庫暨考試系統」網站。也可以直接輸入網址：<http://coding-frenzy.arping.me/>，參考??。



圖 16: 「瘋狂程設」搜尋結果

進入瘋狂程設首頁後，點下方的「2. 安裝專用軟體」，參考??及??。



圖 17: 瘋狂程設首頁



圖 18: 下載頁面

請將剛才下載的壓縮檔複製到 C 槽裡，直接放在根目錄 C 下面，然後解壓縮。對 zip 檔按「右鍵」-> 解壓縮到 CodingFrenzy@coding-frenzy.arping.me\，參考??。

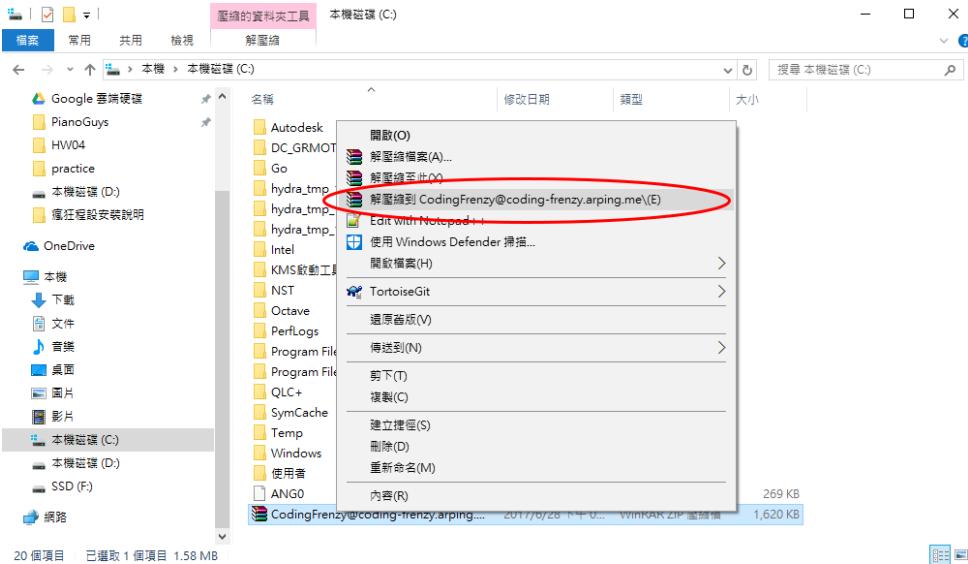


圖 19: 右鍵 -> 解壓縮到 CodingFrenzy@coding-frenzy.arping.me\

之後，C 槽裡面會出現一個資料夾「CodingFrenzy@coding-frenzy.arping.me」，點開此資料夾，參考??。

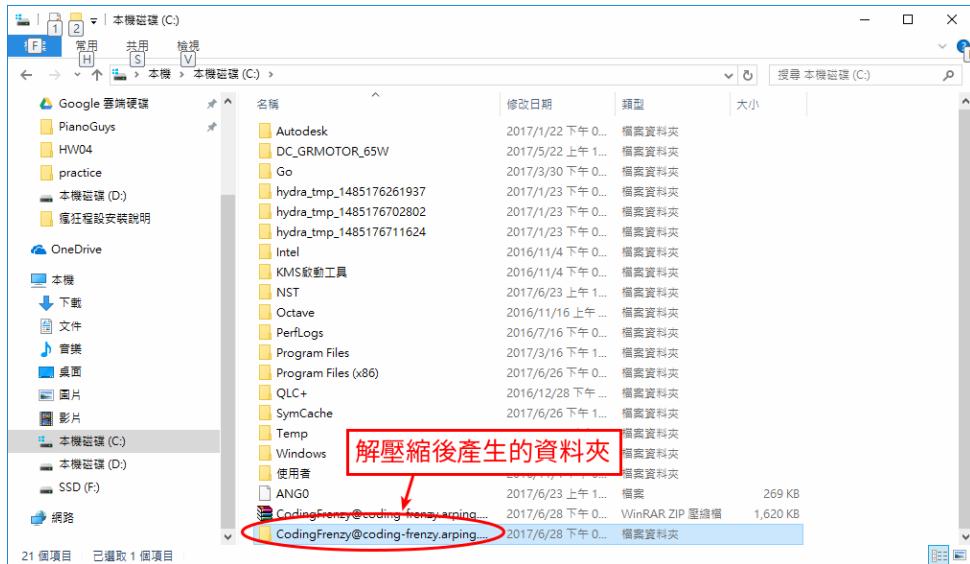


圖 20: 點開解壓縮後的資料夾

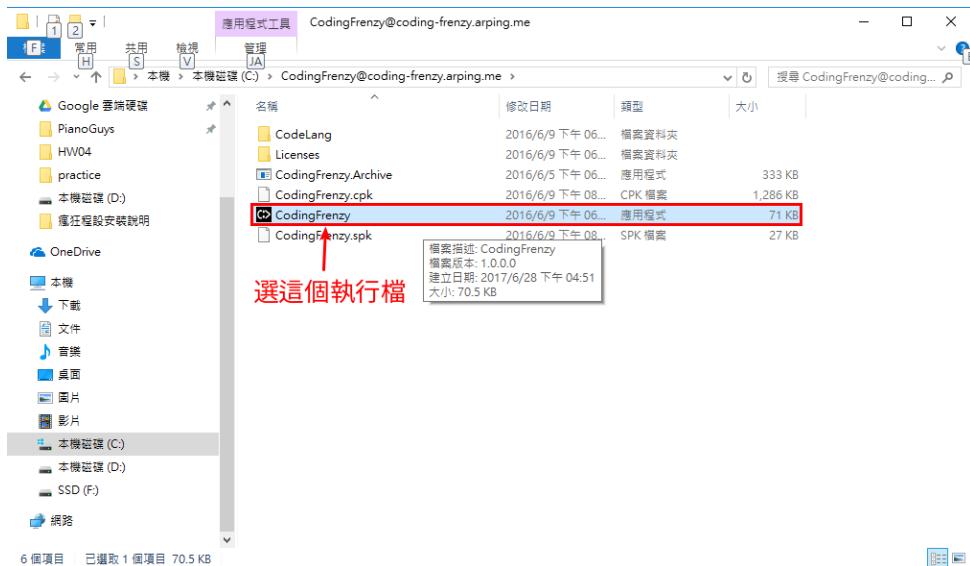


圖 21: 開啟 CodingFrenzy 執行檔

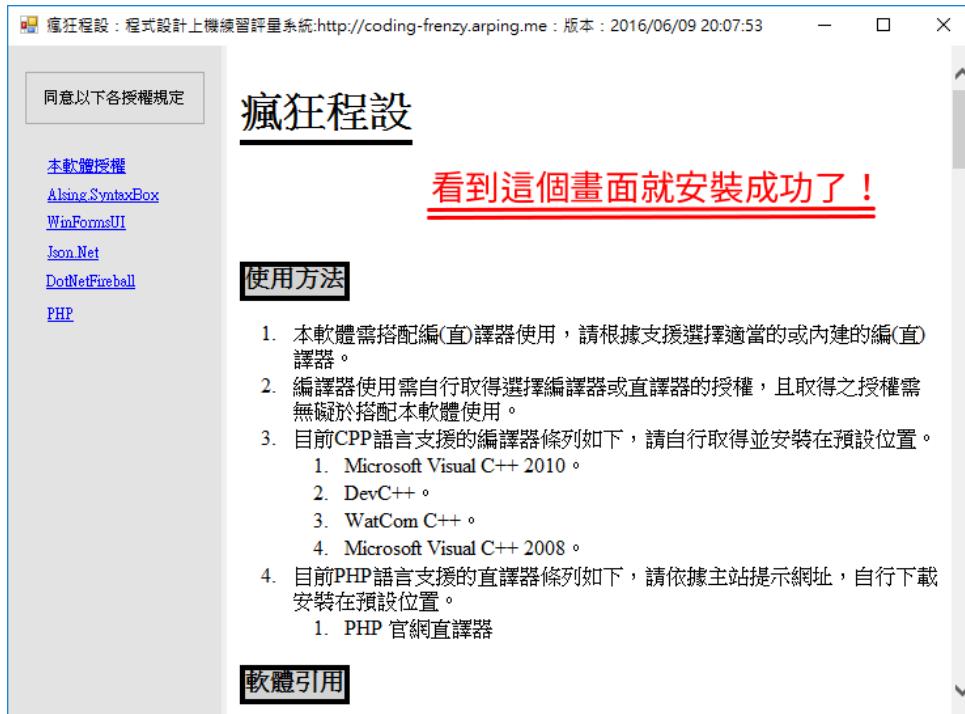


圖 22: 安裝成功

回到資料夾裡，會多出一些東西，不用理它們。對 CodingFrenzy.exe 執行檔按「右鍵」->「傳送到」->「桌面(建立捷徑)」，參考??。

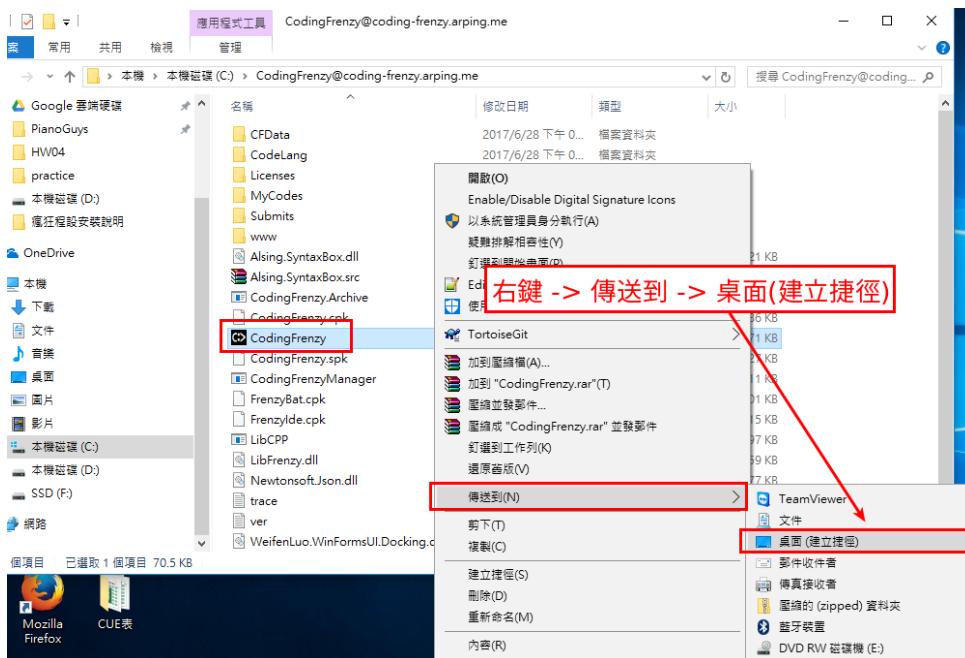


圖 23: 建立桌面捷徑

按鍵盤「win+D」切換到桌面，找到 CodingFrenzy 桌面捷徑，以後就可以從這邊直接執行瘋狂程設了，參考??。

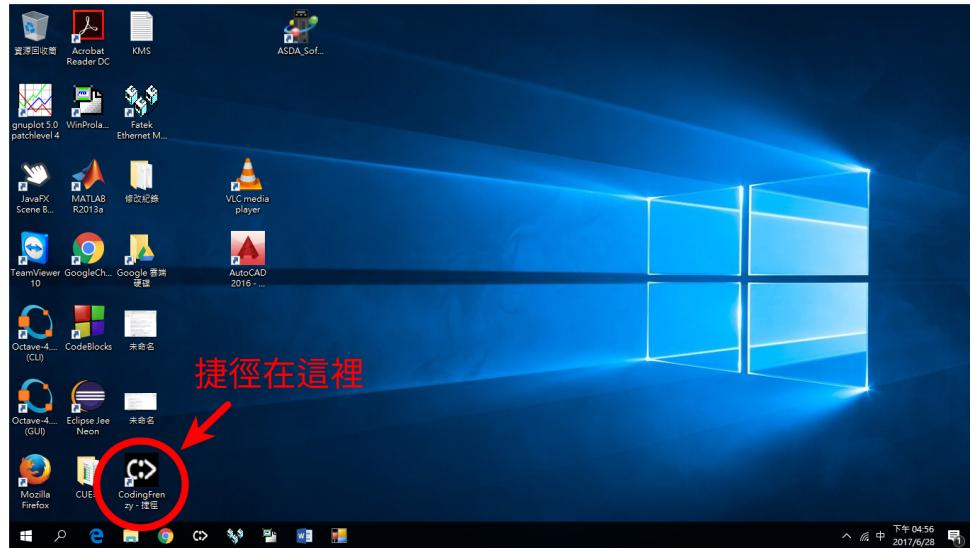


圖 24: 桌面捷徑建立成功

9.3 註冊瘋狂程設的帳號

現在要教同學如何註冊瘋狂程設的帳號。網頁版及桌面版的瘋狂程設皆可註冊，方法大同小異，範例是使用網頁版進行註冊。

進入「瘋狂程設」首頁後，點選左下角的「現在註冊一個帳戶吧！」，參考??。



圖 25: 註冊新帳戶

在框框處輸入你的電子信箱，完成後，按下方的按鈕「取得帳號金鑰」，參考??。

註冊成功會顯示紅字「註冊成功，請前往信箱提取開頭為 xxxx 的金鑰來設定密碼。」，參考??。



圖 26: 輸入 e-mail，取得金鑰

* 範例中的金鑰開頭四碼是 e017，你的金鑰開頭四碼與範例不同是正常的。



圖 27: 成功取得金鑰

到你的電子信箱去收信，會收到一封主旨為「Coding-Frenzy Account is Created」的信，若沒收到可以去垃圾信件夾找找看，參考??。

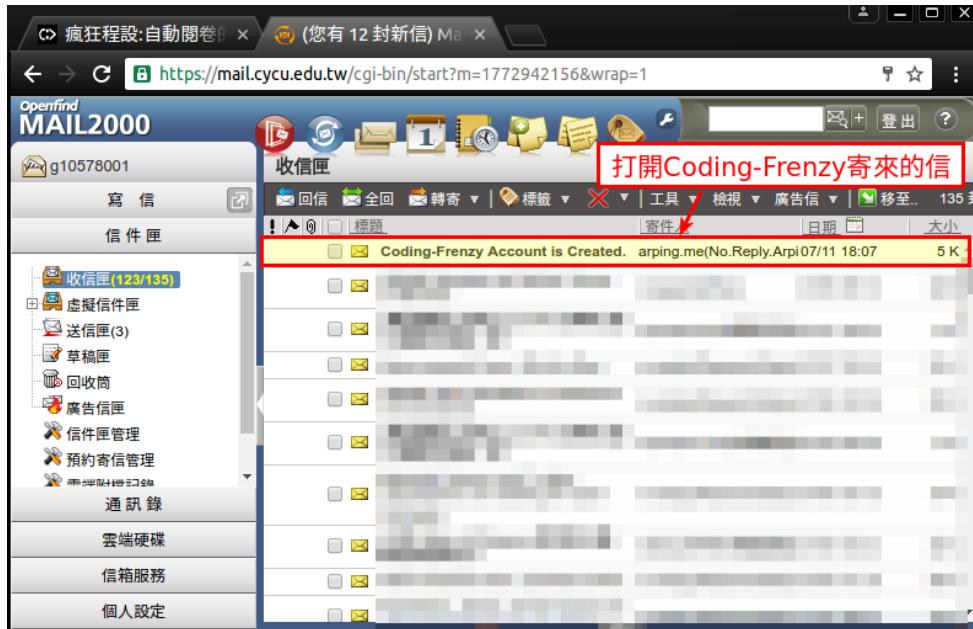


圖 28: 去你的電子信箱收信

點 LINK 後面的連結，會自動開啟一個新的分頁，進入「帳戶設定」頁面，參考??。



圖 29: 點 LINK 後面的連結

系統會自動填寫金鑰，將個人資料填寫完畢後，按下「我同意修改資料，並放棄個資法的求償權利。」，參考??。

The screenshot shows a web page titled "瘋狂程設" with a sub-section for "修改個人資料(欄位空白表示不修改)". The page includes the following fields:

- *電子郵件: g10578001@cycu.edu.tw
- *帳戶金鑰(必填): e017a6a59aea3c5325f29d3c90e688a9 (highlighted with a red box)
- 我忘記我的金鑰了
- *學校名稱(或公司名): 中原大學
- *系所名稱(或部門名): 電機工程
- *姓名或暱稱: 梁家其3
- *學號(或教職員編號): g10578001
- *設定密碼: (highlighted with a red box)
- *確認密碼: (highlighted with a red box)

Below the form, there is a note: "本站致力於減少個人資料蒐集，但所有填寫資料皆會在網站使用中被公開，如果你不願意個人資料被公開，請隨意填寫即可，或放棄創建帳號。" At the bottom, there is a button labeled "我同意修改資料，並放棄個資法的求償權利。" and a link "個資法相關資料".

Annotations with red boxes and arrows:

1. 金鑰會自動填寫 (points to the account key field)
2. 依課程規定填寫 (points to the school name and department fields)
3. 設定密碼 (points to the password input fields)
4. 填完這裡 (points to the "I agree" checkbox at the bottom)

圖 30: 填寫基本資料

按下按鈕後，畫面沒有變化是正常的，這時候請先「登出」(頁面上方)，參考??。



圖 31: 登出

回到首頁後，用剛剛設定的學號及密碼登入，參考??。

若看到此畫面，恭喜你完成註冊，參考??。



圖 32: 登入畫面

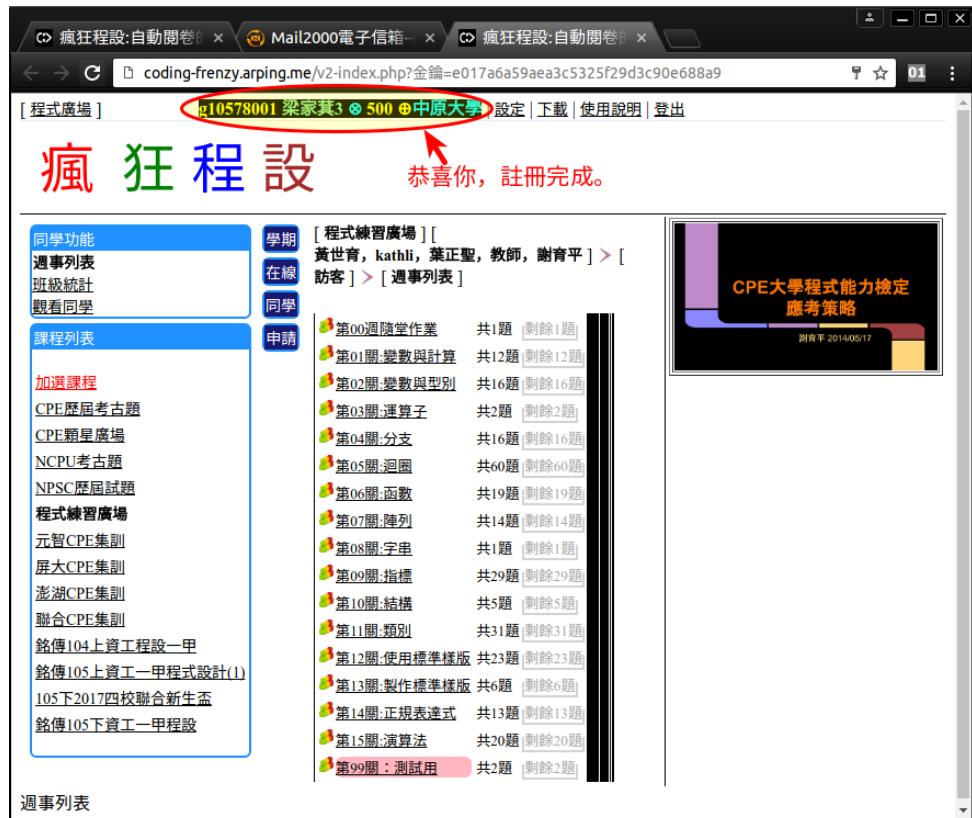


圖 33: 登入成功

9.4 如何使用瘋狂程設

現在要說明如何使用瘋狂程設練習寫程式。

1. 點擊「程式練習廣場」。
2. 點擊「第 01 關: 變數與計算」。
3. 點擊「練習」A001：Hello World。

參考??。



圖 34: 選擇練習題目

4. 查看「題目資料」。
5. 有些題目的「解文」會有解題步驟。
6. 輸入程式碼。

注：程式碼的解說會在後續章節詳細說明，現在請直接照圖片上的範例輸入程式碼，或是複製「解文」裡的程式碼。

參考??。

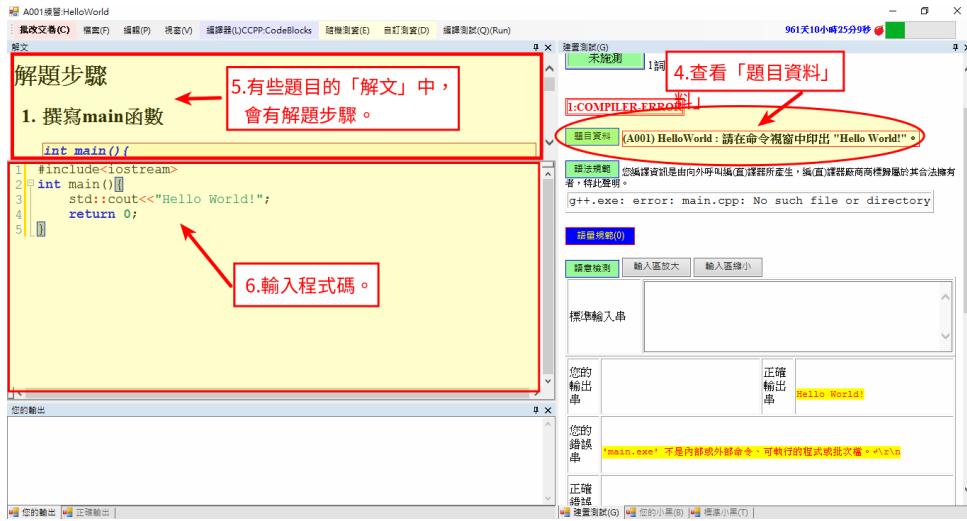


圖 35: 依題目要求輸入程式碼

7. 設定「編譯器」，選擇「CCPP:CodeBlocks」，如果找不到此編譯器，請重新安裝 CodeBlocks 。

參考??。

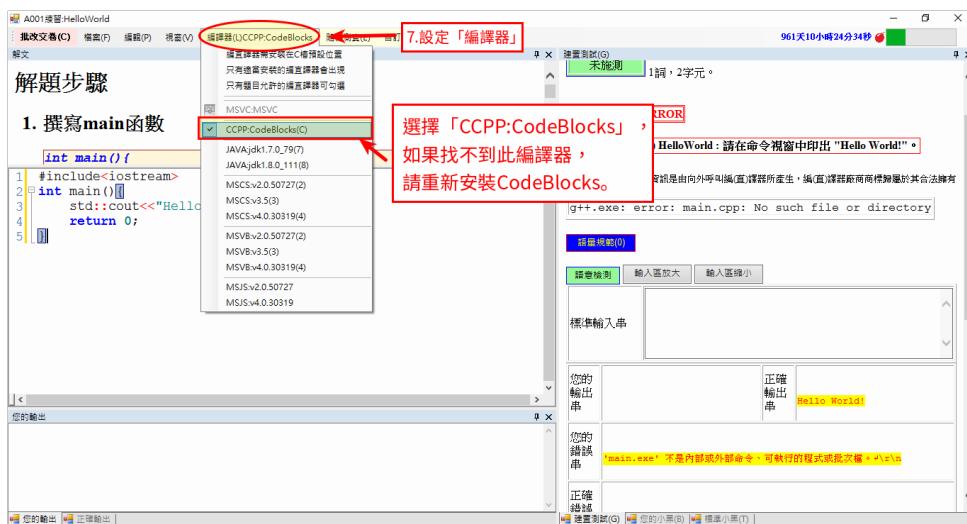


圖 36: 設定「編譯器」

8. 點擊「隨機測資」或「自訂測資」。
9. 如果程式有錯誤。
10. 查看「編譯錯誤訊息」。

參考??。

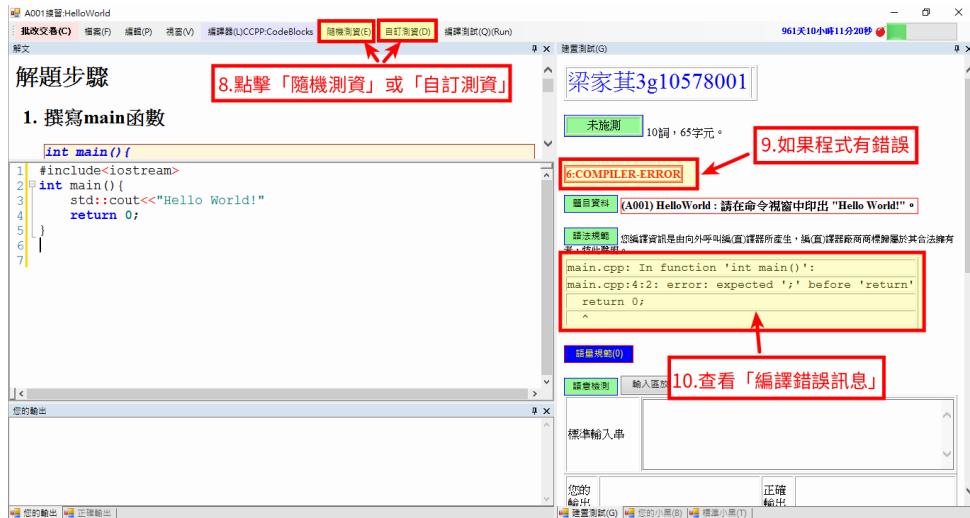


圖 37: 測試程式碼

11. 修改程式碼。
12. 再次使用測資測試。
13. 程式正確，「顯示 CORRECT」。
14. 點擊「批改交卷」。
15. 點擊「確定」，關閉提示視窗。
16. 顯示「通過」。
17. 關閉作答視窗。

參考??。

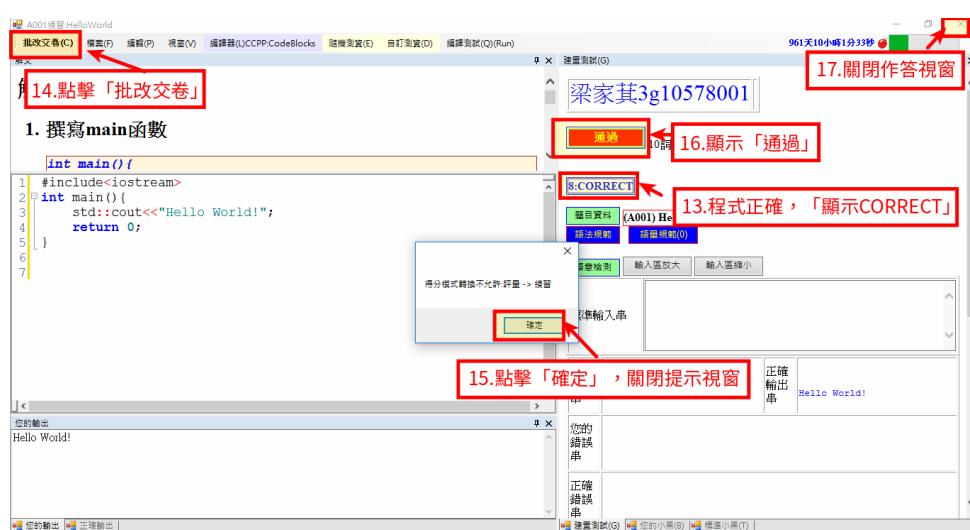


圖 38: 批改交卷

18. 在主畫面按【F5】更新作答成績。
19. 通過「挑戰」模式，顯示「金牌」。
20. 通過「練習」模式，顯示「練習」。

參考??。



圖 39: 查看成績

10 附錄二、輸入/輸出 (Input/Output, I/O)

基本上，C++ 語言涵括了 C 語言的功能，所以我們也可以在 C++ 的環境中，撰寫 C 的程式碼。一般而言，大部份的程式都會有輸入和輸出的需求，而 C++ 和 C 的輸入和輸出方式，各有其優點，在不同的場合，有時候使用 C++ 的輸入輸出方式會比較方便，有時候使用 C 的輸入輸出函數會比較方便，如果可以把兩種方式都學會，在撰寫程式碼的時候，可以有很多便利，所以這裡先介紹兩種語言基本的輸入和輸出的使用方法。

10.1 C 語言的輸入和輸出

C 語言使用 `scanf` 和 `printf` 兩個函數來做輸入和輸出，使用 C 語言的輸入輸出函數，C 程式碼必須引入檔頭 `<stdio.h>`，如果在 C++ 的環境中，則可以引入 `<stdio.h>` 或 `<cstdio>`。以下先介紹 `printf` 函數的用法，接著再介紹 `scanf` 函數的用法。

10.1.1 `printf` 輸出函數

C 語言使用 `printf` 函數將訊息列印至標準輸出 (standard output)，一般而言，標準輸出指的是螢幕，除了列印字串，也可以列印各種型態的變數，執行過後會回傳所列印的字元數。`printf` 函數至少要有一個參數，而且第一個參數一定是一個字串，執行結果會把這個字串列印到螢幕上，例如：

```
printf("format string");
```

其執行結果如下：

```
format string
```

字串中可以加入跳脫符號，用來控制輸出的樣貌，常用的控制字元如??所示，例如 `\n` 代表換行字元，如果希望在字串的某個地方換行，可以把 `\n` 插入到字串裡面。

另外也可以在第一個字串中加上格式指定字元 (format specifier)，用來指明要列印的變數或數值型式，字串中有幾個格式指定字元，後面就必須加上相對應個數的參數，執行結果會先把各參數依指定型式放到字串中，然後再將字串輸出。常見的格式指定字元如??所示。

表 1: 字串中常用的跳脫符號

字元格式	字元功能
\0	空格
\b	倒退
\t	移到下一定位，即【Tab】鍵。
\n	游標移到下一列。
\”	插入雙引號。
\`	插入單引號。
\\"	插入反斜線。
\a	發出警告聲。

表 2: printf 常見的格式指定字元

指定碼格式	功能
%c	以字元方式輸出。
%d	10 進位整數輸出。
%o	以 8 進位整數方式輸出。
%u	無號整數輸出。
%x,%X	將整數以 16 進位方式輸出。
%f	浮點數輸出。
%e,% E	使用科學記號顯示浮點數。
%g,%G	浮點數輸出，取%f 或%e (%f 或%E)，看哪個表示精簡。
%%	顯示。%
%s	字串輸出。
%lu	long unsigned 型態的整數。
%p	指標型態。

舉例而言，

```
var1 = 5;
printf("The value of a and b = %d %f", var1, 3.14159);
```

其中%d 代表要放一個整數，%f 代表要放一個浮點數，這兩個數分別會從後面的 var1 及 3.14159 取得，其執行結果如下：

```
The value of a and b = 5 and 3.14159
```

以下是使用一些格式指定字元的範例程式碼：

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf(" 顯示字元 %c\n", 'A');
6     printf(" 顯示字元編碼 %d\n", 'A');
7     printf(" 顯示字元編碼 %c\n", 65);
8     printf(" 顯示十進位整數 %d\n", 15);
9     printf(" 顯示八進位整數 %o\n", 15);
10    printf(" 顯示十六進位整數 %X\n", 15);
11    printf(" 顯示十六進位整數 %x\n", 15);
12    printf(" 顯示科學記號 %E\n", 0.001234);
13    printf(" 顯示科學記號 %e\n", 0.001234);
14    return 0;
15 }
```

上述程式碼的執行結果：

```
顯示字元 A
顯示字元編碼 65
顯示字元編碼 A
顯示十進位整數 15
顯示八進位整數 17
顯示十六進位整數 F
顯示十六進位整數 f
顯示科學記號 1.234000E-03
顯示科學記號 1.234000e-03
```

10.1.2 scanf 輸入指令

C 語言使用 scanf 函數從標準輸入 (standard input) 來讀取變數的值，一般而言，標準輸入指的是鍵盤。scanf 的第一個參數是一個字串，通常裡面都放格式指定字元和空白，表示要讀入的變數的型態，後面必須加上與格式指定字元同樣個數及相應型態的變數，並且在變數前要加上 & 字元，以下是一個簡單的範例：

```
1 #include <stdio.h>
2
3 int main() {
4     int input;
```

```
5     printf(" 請輸入數字：");
6     scanf("%d", &input);
7     printf(" 您輸入的數字：%d\n", input);
8     return 0;
9 }
```

其執行結果如下：

```
請輸入數字：10
您輸入的數字：10
```

10.2 C++ 語言的輸入和輸出

C++ 語言使用 cin 和 cout 來做輸入和輸出，使用 C++ 語言的輸入輸出，程式碼必須引入檔頭 <iostream>。另外 C++ 有所謂的命名空間，所有標準輸入和輸出相關的函數、指令和參數等，都定義在 std 的命名空間中，使用這些函數、指令或參數的時候，必須在前面加上 std::，例如使用 cout 的時候，必須寫成 std::cout，如果覺得這樣很麻煩，可以在引入檔頭之後，加上以下指令

```
using namespace std;
```

代表要引入 std 命名空間中的所有函數、指令和參數等，這樣就可以直接使用 cout 等指令，不用再加上 std:: 的前綴。

以下先介紹 cout 的用法，接著再介紹 cin 的用法。

10.2.1 cout 輸出指令

C++ 語言使用 cout 指令將資料送到標準輸出，若沒有特別設定，會由電腦螢幕上顯示。使用方式以範例說明如下：

```
cout << "String";
cout << "String" << var1;
cout << var1 << var2;
```

其中 << 為輸出運算子，字串與變數會分別以相應的預設格式輸出，不需要使用格式指定字元。例如以下範例程式：

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout<<"Hello world!"<<endl;
6     cout<<"Hello world!"<<12345<<endl;
7     cout<<12345<<67890<<endl;
8     return 0;
9 }
```

其輸出結果如下：

```
Hello world!
Hello world!12345
1234567890
```

— — —

在 cout 輸出指令中，可以單獨輸出 endl 符號，代表換行字元，或者也可以使用 '\n' 字元來達到相同的效果。例如以下範例：

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int x=20,y=15;
6     cout<<" 求兩數和\n";
7     //相同於 cout<<" 求兩數之和 "<<endl;
8     cout<<x<<"+"<<y<<"="<<x+y<<"\n";
9     //相同於 cout<<x<<"+"<<y<<"="<<x+y<<endl;
10    return 0;
11 }
```

其執行結果如下：

```
求兩數和
20+15=35
```

10.2.2 cin 輸入指令

C++ 語言使用 cin 指令從標準輸入讀取資料到相對應的變數中，若沒有特別設定，會由鍵盤輸入讀取，使用者輸入資料後按下【Enter 鍵】時，會自動以空白 (Space) 鍵或 Tab 鍵作為資料的分隔字元，故輸入之資料不可含空白鍵或 Tab 鍵。使用方式以範例說明如下：

```
cin >> var;  
cin >> var1 >> var2;
```

分別代表從鍵盤讀取資料到 var 及 var1 和 var2，其格式依變數型態自動判別。例如以下範例程式：

```
1 //求兩數四則運算  
2 #include <iostream>  
3 using namespace std;  
4 int main()  
5 {  
6     float x,y; //宣告 x,y 為浮點數  
7     cout<<" 輸入 x=";  
8     cin>>x;  
9     cout<<" 輸入 y(不可為 0)=";  
10    cin>>y;  
11    cout<<"x+y="<<x+y<<endl;  
12    cout<<"x-y="<<x-y<<endl;  
13    cout<<"x*y="<<x*y<<endl;  
14    cout<<"x/y="<<x/y<<endl;  
15    system("PAUSE");  
16    return 0;  
17 }
```

其執行結果如下：

```
輸入x=34.98  
輸入y(不可為0)=10.007  
x+y=44.987  
x-y=24.973  
x*y=350.045  
x/y=3.49555
```