

C語言基本輸出與輸入方法

螢幕顯示與輸出格式

- printf() 的輸出格式控制
- printf("格式化字串" , 變數1, 變數1, ...);

控制符號	說明	適用之型別
%d	十進位的整數	int
%u	無正負號的十進位整數	unsigned int
%o	無正負號的八進位整數	unsigned int
%x	無正負號的十六進位整數 (英文部分小寫 a-f)	unsigned int
%X	無正負號的十六進位整數 (英文部分大寫 A-F)	unsigned int
%f	浮點數 (小數點表示法)	float, double
%e	浮點數 (科學符號表示法, 指數使用小寫 e)	float, double
%E	浮點數 (科學符號表示法, 指數使用大寫 E)	float, double
%g	輸出 %f 與 %e 長度較短者	float, double
%G	輸出 %f 與 %E 長度較短者	float, double
%c	輸出字元型別的資料	char

- `printf()` 可輸出變數值, 算式的運算結果。要注意指定算式的輸出格式時, 必須能符合運算結果的資料型別
- 輸出格式的應用：不同的浮點數表示法
 - 以小數點表示浮點數 `%f`
 - 以科學符號表示浮點數 `%e`

- 輸出格式的參數

參數	用途
-	輸出的數值向左靠齊
+	在輸出的數值前面加上正或負符號
非零的數字	以數字表示要提供多少固定欄位給輸出的數值
h	輸出的數值使用 short int 的資料型別表示
l	輸出的數值使用 long int 的資料型別表示

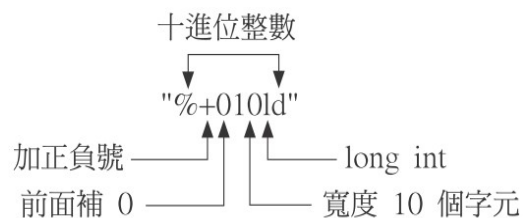
- 指定固定寬度的輸出欄位

- 整數的設定方式：**直接以數字指定寬度**
- 浮點數的設定方式：**可同時指定整數部分和小數部分的位數**，其間以小數點隔開

```
%5d      ← 5 位整數，未滿 5 位以空白字元代替
%010d    ← 10 位整數，未滿 10 位前面補 0
%7.2f    ← 包含小數點共 7 位的浮點數，小數部分佔 2 位，
           未滿 2 位處補 0；整數部分 4 位，未滿 4 位處補空白
%07.2f   ← 與 %7.2f 類似，差別在於整數部分，若未滿 4 位處要補 0
```

- 格式控制符號中參數

- %[- + 0] [寬度] [. 精確度] [h / l] **輸出格式**



- 使用 Escape Sequence 控制輸出

Escape Sequence	效果
\a	發出嗶聲
\b	倒退一格刪除一個字元
\n	換行
\r	歸位, 前面的字元都刪除
\t	向右跳一個 TAB 的空白
\0	結束字元, 後面的字元不印出
\\	輸出反斜線
\'	輸出單引號
\"	輸出雙引號

鍵盤輸入與格式設定

- 格式化輸入函式：scanf() 函式
 - `scanf(“輸入格式”, &變數1, &變數2,...)`
 - &變數名稱：用來接受輸入值, & 表示取得變數在記憶體的字址

```
scanf("%d", &age)      ← 由鍵盤輸入整數型別的變數 age 值
scanf("%f", &tax_rate) ← 由鍵盤輸入浮點數型別的變數 tax_rate 值
scanf("%c", &mychar)  ← 由鍵盤輸入字元型別的變數 mychar 值
scanf("%d %d", &count, &sum) ← 由鍵盤輸入兩整數型別的變數 count, sum 值
scanf("%4d", &code)   ← 只會讀取從鍵盤輸入的前 4 個數字
```

C語言格式化控制符號

字元	char*	%c	字串	char*	%s
		整數	正整數	八進位	十六進位
(unsigned) char*		%hhd	%hhu	%hho	%hhx
(unsigned) short*		%hd	%hu	%ho	%hx
(unsigned) int*		%d	%u	%o	%x
(unsigned) long*		%ld	%lu	%lo	%lx
(unsigned) long long*		%lld	%llu	%llo	%llx
浮點數	float*	%f	浮點數	double*	%lf

C語言的FGETS函式

- scanf讀入字串時，若遇到空白字元，就停止。
- fgets可讀取一整列資料，格式如下：


```
char *fgets(char *str,int num,FILE *stream);
```
- fgets讀取資料直到換行字元「\n」或資料結束 (EOF)，或最多讀入num-1個字元(所得字串會以「\0」結尾)。若有換行字元也會一併讀取進來。
- 若不希望所讀得的字串尾端含有換行字元，可以下列將換行字元則將之取代為「\0」。

```
#include <string.h> // for strlen
int len=strlen(line);
if (line[len-1]=='\n') line[len-1]='\0';
```

C++的CIN函式

- `char ch, str[64];`
`int num;`
`float value;`
`cin >> ch >> str >> num >> value;`

cin.getline函式

- 讀取到換行字元「\n」或檔案結束 (eof) 為止，所讀得的字串不含換行字元「\n」，並會以「\0」結束：
- `char line[64];`
`cin.getline(line, 64);`

C++的cout函式

```
• char ch='A',str[64]="Apple";  
  int num=3;  
  float pi=3.14;  
  
  cout<<"Hello World!"<<endl;  
  cout<<"Output: "<<ch<<","<<str<<","<<num<<","<<pi<<endl;
```

設定C++的COUT函式輸出格式

```
• 必須先引入 iomanip 標頭檔  
  • #include <iomanip>  
  • double value=1.83;  
    cout<<setw(8)<<setprecision(3)<<fixed<<value;
```

C++輸出格式化控制

setw(n)	保留 n 個字元的寬度
setfill(ch)	內容長度不足時，要填補的字元。
right	靠右對齊
left	靠左對齊
setprecision(n)	設定小數點位數為 n 位
fixed	小數位數不足時補 0
oct	以 8 進位表示
dec	以 10 進位表示
hex	以 16 進位表示
showpos	強制顯示正負符號
noshowpos	負數則顯示負號，但正數不顯示正號

讀至N筆資料 (C語言)

- 第一列為欲處理的資料筆數 (此處假設為n) 。處理方式就是讀進n之值，並執行n次的迴圈，讀取往後的每筆資料

```

• int main() {
    int n;
    scanf("%d", &n);
    while (n-->0) {
        /* 依序讀取每筆資料，並處理之 */
    }
    return 0;
}

```


讀至N筆資料 (C++)

```
• int main() {  
    int n;  
    cin>>n;  
    while (n-->0) {  
        // 依序讀取每筆資料，並處理之  
    }  
    return 0;  
}
```

讀至檔案結束 (C語言)

- `scanf`會回傳它成功讀入的元素個數；當讀到檔案結束時，`scanf`則回傳 EOF

```
• int main() {  
    int x;  
    while (scanf("%d",&x) != EOF) {  
        /* 處理目前這筆資料 */  
    }  
    return 0;  
}
```

讀至檔案結束 (C++)

- 當檔案結束時，cin得到的值會變成NULL，也就是0，代表false

```
• int main() {
    int x;
    while (cin>>x) {
        // 處理目前這筆資料
    }
    return 0;
}
```

讀至0結束 (C語言)

- 讀入資料，簡單地加個判斷是否為0。若是，跳出迴圈即可。

```
• int main() {
    int n;
    while (scanf("%d",&n) != EOF) {
        if (n==0) break;
        /* 處理目前這筆資料 */
    }
    return 0;
}
```

讀至0結束 (C++)

- 讀入資料，簡單地加個判斷是否為0。若是，跳出迴圈即可

```
• int main() {  
    int n;  
    while (cin>>n) {  
        if (n==0) break;  
        // 處理目前這筆資料  
    }  
    return 0;  
}
```

課堂練習1

- 質數判別：輸入一個正整數，如果是質數，則輸出 Yes，如果不是，則輸出 No。
 - 質數：除了 1 及 本身外，沒有其他因數
 - 1 是否為質數？

課堂練習2

- 判斷閏年：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。)(西元1004年為閏年，西元1100年不是閏年，西元1600年是閏年)

課堂練習3

- 輸入兩個正整數 x, y ，輸出 x, y 的最大公因數。

$$\bullet 56 \div 44 = 1 \text{ 餘 } 12$$

$$\bullet 44 \div 12 = 3 \text{ 餘 } 8$$

$$\bullet 12 \div 8 = 1 \text{ 餘 } 4$$

$$\bullet 8 \div 4 = 2 \text{ 餘 } 0$$

歐基里德演算法

1. 輸入兩個自然數 A, B
2. A 除以 B ，餘數為 R
3. 如果 R 為零，則跳至步驟 5
4. $A \leftarrow B, B \leftarrow R$ ，跳至步驟 2
5. B 即為 GCD

計算BMI

- 輸入身高(公尺)及體重(公斤),
- 計算BMI=體重/身高平方 ,
- 若BMI< 18.5 則輸出"too thin"
- 若 18.5<=BMI<24 則輸出 "standard"
- 若 BMI>=24 則輸出 "too fat"
(請使用double)

```
#include <stdio.h>

int main(){
    double ht, wt, bmi;
    scanf("%lf %lf", &ht, &wt);
    bmi = wt/(ht*ht);
    printf("%g\n", bmi);
    if (bmi < 18.5) printf("too thin");
    else if ((bmi>=18.5) && (bmi<24)) printf("standard");
    else printf("too fat");
    return 0;
}
```