

# 三角函式列表

丁培毅

實習目標：

1. `math.h` 中 `sin()`, `cos()`, `tan()`, `atan()` 練習，請注意數值計算時除以 0 會發生除法的錯誤
2. `for` 迴圈練習
3. 進階 `printf` 格式練習
4. `sprintf` 函式練習
5. 迴圈版本 `gcd()` 實作
6. 函式製作練習，請注意函式裡面可以有幾個 `return` 敘述

# 三角函式列表

1. 請撰寫一個程式，運用 `math.h` 裡面的三角函式，計算並且列印 0 到  $2\pi$  之間的 `sin`, `cos`, 與 `tan` 函數值，請注意列印格式與對齊
2. 範例程式輸出

x	sin(x)	cos(x)	tan(x)
0	0.000000	1.000000	0.000000
PI/16	0.195090	0.980785	0.198912
PI/8	0.382683	0.923880	0.414214
3PI/16	0.555570	0.831470	0.668179
PI/4	0.707107	0.707107	1.000000
5PI/16	0.831470	0.555570	1.496606
3PI/8	0.923880	0.382683	2.414214
7PI/16	0.980785	0.195090	5.027339
PI/2	1.000000	0.000000	16331239353195370.000000
9PI/16	0.980785	-0.195090	-5.027339
5PI/8	0.923880	-0.382683	-2.414214
11PI/16	0.831470	-0.555570	-1.496606
3PI/4	0.707107	-0.707107	-1.000000
		...	
29PI/16	-0.555570	0.831470	-0.668179
15PI/8	-0.382683	0.923880	-0.414214
31PI/16	-0.195090	0.980785	-0.198912
PI	-0.000000	1.000000	-0.000000

## 分析

1. 這個題目第一件事需要知道  $\pi$  對應的數值是多少，雖然你也許記得是 3.1415926...，但是還是用函式庫來計算吧，`pi = atan(1.0)*4.0`;
2. 由輸出資料可以看到用一個 `for` 迴圈就可以計算並且列印這些數值，迴圈的控制變數直接用一個浮點數 `x`，`x` 由 0 變化到  $2\pi$ ，間隔是  $\pi/16$ ，需要注意的是迴圈的結束條件不可以使用 `x == 2*pi`，因為浮點數的表示方法以及計算都會引入相對的誤差，`x == 2*pi` 的結束條件有可能造成一個無窮迴圈，結束條件永遠都不會滿足，使用浮點數的結束條件一定要用 `>`, `>=`, `<`, `<=`，這裡我們應該要用 `x < 2*pi + 1e-5`，這樣子保證一定可以印出 `x` 為  $2\pi$  的那一列。
3. `tan( $\pi/2$ )`, `tan( $3\pi/2$ )` 應該是要避免的，有可能會因為這樣的呼叫而導致程式發生執行時的計算錯誤，也許因為 `x` 和  $\pi/2$  以及  $3\pi/2$  有一點誤差，直接呼叫 `tan(x)` 函式時才沒有出現錯誤
4. 由輸出範例中看到每一個浮點數有 15 格的欄位空間，靠欄位左邊列印，其中小數點後有六位，這需要在格式字串中使用 `%-15.6f`，其中小數點第七位以後 `printf` 幫我們四捨五入，所以會出現 `-0.000000`

5. 接下來比較麻煩的是第一行要印出最簡分數  $\pi/16, \pi/8, 3\pi/16, \dots, 2\pi$ ，這些需要印在 7 格的欄位中靠左列印，前面步驟 2 中  $x$  是一個浮點數，沒有辦法由浮點數直接印出上面的有理數，浮點數  $x$  裡面存放的根本就只是上面這些最簡分數的近似值 (如果用二進位表示是循環小數，誤差不是 0 的話)。要像範例一樣印出最簡分數，需要在程式中維持分子與分母兩個整數，也就是分子 `num` 由 0 到 32，分母 `denum` 是 16，這樣子可以很容易得到正確的有理數 `num / denum`。接下來就是把最大公因數除掉，就可以得到  $1/16, 1/8, 3/16, 1/4, \dots$  的列印結果，最後需要在看到  $1\pi/16, 1\pi/8, 1\pi/4, 1\pi/2, 1\pi/1, \dots$  這種比較奇怪的寫法換成  $\pi/16, \pi/8, \pi/4, \pi/2, \pi, \dots$  也就是出現 1 時需要用條件敘述檢查出來簡化一下。

6. 上一個步驟中計算最大公因數可以用遞迴函式或是迴圈來實作，如果用迴圈的話，基本上用一層迴圈實作右側的輾轉相除法，請寫成一個函式 `int gcd(int a, int b)`

$$\begin{aligned}
 1180 &= 2 \cdot 482 + 216 \\
 482 &= 2 \cdot 216 + 50 \\
 216 &= 4 \cdot 50 + 16 \\
 50 &= 3 \cdot 16 + 2 \\
 16 &= 8 \cdot 2 + 0
 \end{aligned}$$

gcd

6. 基本上是一個迴圈，迴圈裡面主要作一個整數除法  $a$  除以  $b$  得到商  $q$ ，餘數  $r$  ( $a = q * b + r$ )，如果  $r$  為 0 的話，迴圈結束，最大公因數就是除數  $b$ ；如果  $r$  不為 0 的話，迴圈繼續下一次，但是把被除數  $a$  換成上一次的除數，把除數  $b$  換成上一次的餘數...
- 完成上面步驟以後，和要求的輸出非常接近，但是還是有一點點差別，在列印  $a/b$  時如果直接用 `printf("%dPI/%d",a,b)` 的話，印在螢幕上可能是 5 格或是 6 格或是 7 格，因為在步驟 6 中算出  $a$  和  $b$  時，有的  $a$  是十進位一位數，有的是兩位數， $b$  也是這樣，有的時候需要印出 `/`，有的時候不需要，這使得在 7 格的欄位中有的需要補 5 個空格，有的不需要補任何空格。
    - ① 一種解決的方式是去計算  $a$  和  $b$  的位數、結合  $a$  和  $b$  是不是 0 或 1 的判斷來決定要補幾格空格： $a$  的數值在 0 到 31 之間， $b$  的數值在 1 到 16 之間，很容易可以用 `if` 條件判斷來決定其位數，在用 `for (i=0; i<nblanks; i++) printf(" ");` 來補印空格，如果要一般化的話，也可以寫一個函式來判斷一個整數轉換成十進位會有幾位數。
    - ② 另外一種解決的方式是運用 `stdio.h` 裡面的 `sprintf()` 函式，這個函式和 `printf()` 函式的用法幾乎是一樣的，唯一不同的地方是它不把資料直接寫到螢幕上去，而是寫到一個字元陣列裡面，基本用法如 `char buf[100]; sprintf(buf, "...%....",...);` 資料到了 `buf[]` 字元陣列

7. 裡面以後，就可以用 `printf("%-7s",buf)`; 在 7 格的欄位裡面靠左列印出來，`buf` 字串裡面長度不夠時會自動填入空格