

#include<stdio.h>

int main(){

int answer;

int k,i,j;

for(k = 1; k < 4; k++){

for(j = 1; j < 10; j++){

for(i = 3 \* k - 2; i < 3 \* k + 1; i++){

answer = i \* j;

if(answer <= 9) printf("%dx %d= %d     ", i, j, answer);

if(answer > 9) printf("%dx %d=%d     ", i, j, answer);

}

printf("\n");

}

printf("\n");

}

return 0;

}

這個程式是要用C語言印出九九乘法表，為了要讓它長得像上方那樣，我印的順序大概是：

1x 1= 1    =>    2x 1= 2    =>    3x 1= 3    =>    1x 2= 2    =>    2x 2= 4    這樣子下去...

而到了3x 9=27後，就變成再執行差不多的程式一次，只不過第一個數都加了三，變成：

4x 1= 4    =>    5x 1= 5    =>    5x 1= 5    =>    4x 2= 8    =>    5x 2=10......

到最下面的三列也是一樣，第一個數則又加了三：

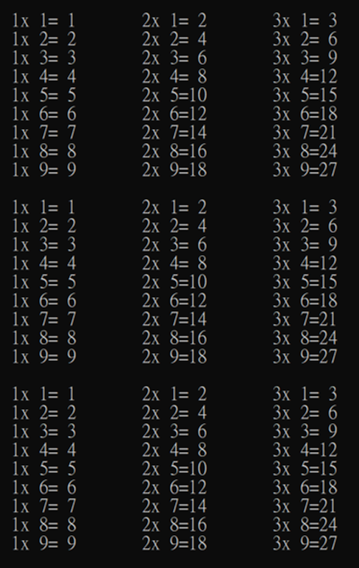
7x 1= 7    =>    8x 1= 8    =>    9x 1= 9    =>    7x 2=14    =>    8x 2=16.....

那我們可以看出來，同樣的事情總共執行了3次，這就是最外面那個for loop在做的事情，

for (k = 0; k < 4; k++){}就是在說k只能小於4（k只能是整數），也就是k最多為3，所以當k被加到3為止就會停了，而k一開始是0，所以從k = 0開始，每次加1，總共要加3次，所以從k是0到k是3的時間，就會讓for loop裡面的程式執行3次，這就是最外圈for loop的功用。

第二個for loop是負責讓每個數字都可以從乘以1到乘以9，總共要乘以9個數，也就是執行9次，所以可以些出for (j = 1; j < 10; j++){}。j就是那個在每個式子裡的乘數，一開始等於0，最多只能等於9（因為是小於10的最大整數），j會被一路從1加到9，除了會讓同個被乘數的每個乘數不一樣之外，這個for loop也會讓裡面的程式執行9次。

最裡面的for loop就是要印出九九乘法表啦，for(i = 3 \* k - 2; i < 3 \* k + 1; i++){} 這行看起來比較複雜一點，i沒有在一開始就被設為某個數，而是等於3\*k - 2，是因為每一段的起始的被乘數都不一樣，第一段是1，第二段是4，第三段是7，那我們就不能直接把i設成一個固定的數，假如我們先i設為1，而i要小於4，它會印出這樣：

[](https://www.blogger.com/blog/post/edit/1920994011767963299/5480905804689785756)

其實就是每段都是1到3的乘法，因為它每段i都還是一樣，那我們要把i設為甚麼呢？我們先回到第一個for loop，k一開始等於1，最後會等於3。所以當k等於1時，第一個i也等於1，當k等於2時，i開始是等於4，最後k是3時，i一開始是7。我們可以發現第一個i（每一段的第一個被乘數）都是k乘以3然後減2的結果（3 x 1 - 2 = 1、3 x 2 - 2 = 4、3 x 3 - 2 = 7），所以在最內層的這個for loop的式子裡，i應該要先等於3\*k-2。

那i < 3 \* k + 1這條式子怎麼得出來的呢？一樣的想法，k等於1時，i應該是要小於4（因為最大值是3），k等於2時，i是小於7（最大值是6），最後k等於3時，i是要小於10（i的最大值為9），那就得出了每段的i（被乘數）都要小於3 \* k + 1（3 x 1 + 1 = 4、3 x 2 + 1 = 7、3 x 3 + 1 = 10），其實也可以寫成i <= 3 \* k就好。

接著我們來看最內層的for loop執行了甚麼呢？首先，先算出每一個式子的答案，所以answer這個變數等於i乘以j（i是被乘數，j是乘數）。算出來後，終於要來印出來這張九九乘法表了，為甚麼有兩行印出的程式呢？注意看一下這張九九乘法表，可以發現如果商是個位數時，商前面會有一個空格，而如果是兩位數時，就沒有，這樣才能讓每個式子都對齊。第一個式子的判斷式是當answer這個變數（商）小於等於9的時候（answer最小只可能是1，1到9都是個位數），在印出的時候，商前面有多一個空格。第二個判斷是當商answer大於9（都是整數，最小是10），在式子裡商前面沒有空格。這些就是最內層迴圈的執行內容。

其實最外面的和第二層迴圈也都有一行印出的程式，"\n"就是換行的意思，第二層的是再說只要每一行（1x 1= 1、2x 1= 2、3x 1= 3 之類的）執行完時，就會跳到下一行再繼續執行。而最外層的是當每一段（到3x 9=27、6x 9=54、9x 9=81）結束時，會跳一行。

這些大概就是這個印出九九乘法表的程式的邏輯，我來講一些簡單的for、printf、if的用法：

for loop：

寫法是 for(i = 0; i < 5; i++){} 前面要先設好i這個變數（在C語言要用int i; 來設，因為i後面是要被設為零這個整數，int是integer整數的意思）

這行的意思是，i一開始等於零，只要i還是小於五，i就會執行這個迴圈裡的程式，然後自己加一（++是加一），而且會再執行一次，執行到i大於等於五了。

例：

for(j = 2.5; j > 1; j -= 0.5){

    printf("%f", j);

}

這裡的j有小數點，是浮點數，前面要用float j; 來宣告變數。

這一段的意思是，j先等於2.5，當j還是大於1時，就會執行大括弧裡面的程式printf("%f",j); 等一下我在printf的介紹帶講這行的意思，執行完後，j就會減0.5（-= 0.5就是減0.5），然後再繼續重複這個for loop直到j小於等於1。

printf：

printf()是一個印出東西的程式，一定要記得寫那個f喔，用法就是：printf("想寫的東西");

不過如果要印出變數的話，由於C語言是個很嚴謹的語言，要跟宣告變數一樣，寫出變數的型態，寫法是：printf("%d", 整數變數);

每種型態的變數的代號都不一樣，%d代表整數變數，%f是浮點數的變數，而且%f也可以寫成%.2f（也可以點其他數），就會變成小數後只有兩位的浮點數。

剛剛for loop介紹的那行printf("%f", j); 的意思現在應該知道了吧，就是印出j這個浮點數。

printf()裡面也能放兩個以上的變數喔，九九乘法表上的：printf("%dx %d= %d     ", i, j, answer); 這行的第一個%d就是第一個變數i，第二個%d就是j，最後的%d是最後的變數answer。

總結大概就是，printf()裡面的雙引號是有被印出來的地方，而有變數時，雙引號中要寫下那個變數的型態，然後雙引號後先寫逗號，再寫要印的變數，如果是兩個以上的變數的話，就照著順序寫每個變數，中間也要用逗號分隔。

if else：

用法是：if (i < 5){} （前面用int設一個i的變數，而且要有放整數的值在裡面）

意思就是，如果這個i的變數裡面的數小於五，那就執行大括號中的程式。

而有if的時候，也可以寫else if和else，像是：

int i = 2;

if(i < 2) printf("1");

else if(i == 2) printf("2");

else printf("3");

第一行是在宣告整數變數i等於2

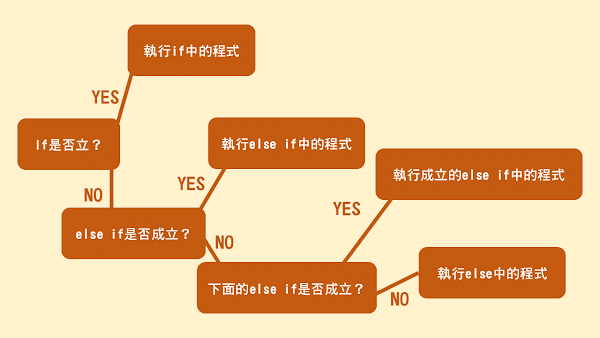
第二行是說，當i小於2時，印出1

第三行是指，當if沒有成立，那如果i是等於2時，印出2

最後一行說，如果if和else if都沒有成立，那其他情況時，印出3

所以執行這個程式後會印出2

else if 可以寫好幾個，整體來說執行過程大概是：

[](https://www.blogger.com/blog/post/edit/1920994011767963299/5480905804689785756)

這就是if else判斷式的執行過程。

for loop、printf()、if else是我在這個九九乘法表裡用到比較多的程式，那這次的C語言介紹就到這囉。