

# 數學函式



# 現實的程式世界

---

絕對不是像練習題一樣的三兩行所構成

經驗告訴我們，發展與維護大型程式最好的方法，  
以較小的單元或模組來建構程式

小單元比整個大程式好管理

技巧：各個擊破

# C語言程式的構成 = 程式設計師的新函式 + 標準函式庫

輸入/輸出(`scanf()` /`printf()`)  
字串處理  
字元處理  
常用的數學運算



常用的  
數學運算

• 在數學中，

– 一個函式表示每個輸入值對應唯一輸出值

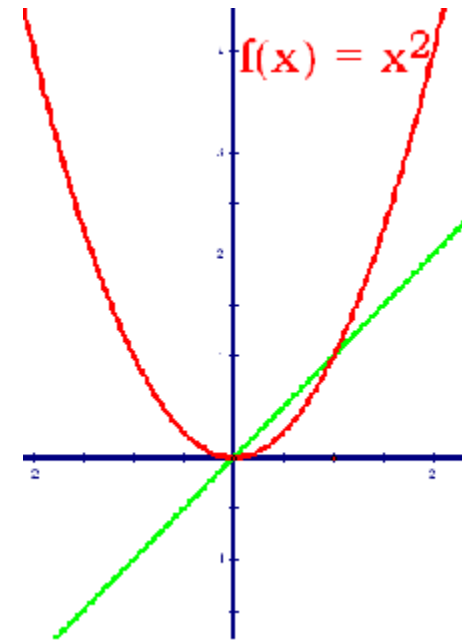
例如:面積

–  $f(x) = x^2$    $y = x^2$

–  $f(3) = 9, f(4) = 16$

所有的輸入值的集合：  
定義域

所有的輸出值的集合：  
值域



```
#include <cmath>
```

呼叫數學函式

# 數學函式的使用

---

```
#include <stdio>
#include <cmath> /*引用數學函式*/
int main(){
    printf("%.2f",sqrt(900.0))
    cout
return 0;}
```

寫下函式名稱，接著是左括號，然後是本函式的引述，最後再一個右括號，

左邊程式可以計算出900的平方根

900.0是sqrt()函式的引數，左邊程式可以輸出30.00

函式的引數可以是常數、變數或是運算式，如果c1=14.0, d=2.0, f=4.0的話：

```
printf( "%.2f" ,sqrt(c1+d*f))
```

可以計算並且印出14.0+2.0+4.0的平方根，答案是5.00

# 數學函式庫的資料型別

---

- 使程式設計師可以執行某些常用的數學運算
- 必須引用數學函式  
`#include <cmath>`
- 參數類型與返回值類型都是double，double與float的值類似，可以使用%lf(long float)格式轉換說明符輸出



# 數學函式

Function	Description	Example
<code>sqrt( x )</code>	square root of $x$	<code>sqrt( 900.0 )</code> is 30.0 <code>sqrt( 9.0 )</code> is 3.0
<code>exp( x )</code>	exponential function $e^x$	<code>exp( 1.0 )</code> is 2.718282 <code>exp( 2.0 )</code> is 7.389056
<code>log( x )</code>	natural logarithm of $x$ (base $e$ )	<code>log( 2.718282 )</code> is 1.0 <code>log( 7.389056 )</code> is 2.0
<code>log10( x )</code>	logarithm of $x$ (base 10)	<code>log10( 1.0 )</code> is 0.0 <code>log10( 10.0 )</code> is 1.0 <code>log10( 100.0 )</code> is 2.0
<code>fabs( x )</code>	absolute value of $x$	<code>fabs( 5.0 )</code> is 5.0 <code>fabs( 0.0 )</code> is 0.0 <code>fabs( -5.0 )</code> is 5.0
<code>ceil( x )</code>	rounds $x$ to the smallest integer not less than $x$	<code>ceil( 9.2 )</code> is 10.0 <code>ceil( -9.8 )</code> is -9.0
<code>floor( x )</code>	rounds $x$ to the largest integer not greater than $x$	<code>floor( 9.2 )</code> is 9.0 <code>floor( -9.8 )</code> is -10.0
<code>pow( x, y )</code>	$x$ raised to power $y$ ( $x^y$ )	<code>pow( 2, 7 )</code> is 128.0 <code>pow( 9, .5 )</code> is 3.0
<code>fmod( x, y )</code>	remainder of $x/y$ as a floating point number	<code>fmod( 13.657, 2.333 )</code> is 1.992
<code>sin( x )</code>	trigonometric sine of $x$ ( $x$ in radians)	<code>sin( 0.0 )</code> is 0.0
<code>cos( x )</code>	trigonometric cosine of $x$ ( $x$ in radians)	<code>cos( 0.0 )</code> is 1.0
<code>tan( x )</code>	trigonometric tangent of $x$ ( $x$ in radians)	<code>tan( 0.0 )</code> is 0.0

再看一個例子，  
你就知道，  
數學專家有多好

## 次方問題

請設計一個 $\text{pow}(x,y)$ 函式，  
計算 $x$ 的 $y$ 次方，  
直到 $x=0$ 停止。

問題



# 叫上數學專家pow

---

```
#include <iostream>
#include <cmath>
int main()
{
    double x,y;
    cin>>x;
    while(x!=0){
        cin>>y;
        cout<<(int)pow(x,y)<<endl;
        cin>>x;
    }
    return 0;
}
```



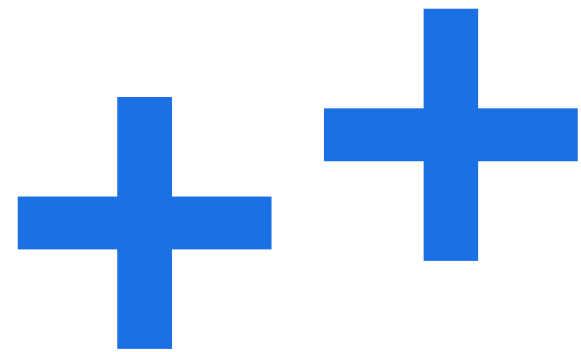
呼叫專家：  
呼叫pow()  
來算x的y次方

# 不要叫專家，自己來試試吧！

```
#include <iostream>
#include <cmath>
int main()
{
    int x,y,t,count;
    cin>>x;
    while(x!=0){
        cin>>y;
        t=1;
        count=1;
        while(count<=y){
            t=t*x;
            count=count+1;
        }
        cout<<t<<endl;
        cin>>x;
    }
    return 0;
}
```



感覺到pow()  
的方便了吧！




# 延伸學習



# 強迫轉型

---

```
#include <iostream>
#include <cmath>
int main()
{
    double x,y;
    cin>>x;
    while(x!=0){
        cin>>y;
        cout<<(int)pow(x,y)<<endl;
        cin>>x;
    }
    return 0;
}
```



這是強迫轉  
型，將結果由  
double轉成  
int