

遞歸 (Recursion) :

- 一個程序或函數直接或間接對自己進行呼叫，這種型式稱為遞歸 / 遞歸呼叫。
遞歸可以為直接遞歸、間接遞歸及“超前引用”之間接遞歸。

使用遞歸時必須符合三個條件：

1. 可將一個問題轉化為一新問題，而此問題之解法與原問題是相同，但問題之對象比原來之縮小，即它們是有規律的遞增或遞減。
(如：求 $N!$ 等於求 $N * (N-1)!$ ，而 $N!$ 與 $(N-1)!$ 之求解法是相同的。)
2. 可以通過轉化過程使問題回到對原問題的求解。
3. 必須有一明確的結束遞歸的條件，否則遞歸會陷入死循環，無止境地進行下去。

一些遞歸應用的例子：

1. 求 $N!$ (N 的階乘)

- a. 先了解階乘準確的遞歸定義；

$$\begin{aligned} N! &= 1 * 2 * 3 * \dots * n \\ &= n * (n-1) * (n-2) * \dots * 2 * 1 \\ &= n * (n-1)! \end{aligned}$$

- b. 將上面之定義寫成為函數之成式；

$$\begin{aligned} f(n) &= n * f(n-1) \quad (n > 0) \\ f(0) &= 1 \end{aligned}$$

練習：試用遞歸的方法求 $1 + 2 + 3 + 4 + 5 + \dots + n$ 的值。

2. 將讀入之一串字符反序輸出，以字符 ‘&’ 為結束符號。

- a. 讀入一個字符；
- b. 讀該字符後的子字符串並反序輸出；
- c. 然後輸出讀入之字符 (指 a. 讀入之字符)；
- d. 在 b. 中若子字符串是空 (即遇到 ‘&’ 時)，表示子字符串，不再處理子字符串。

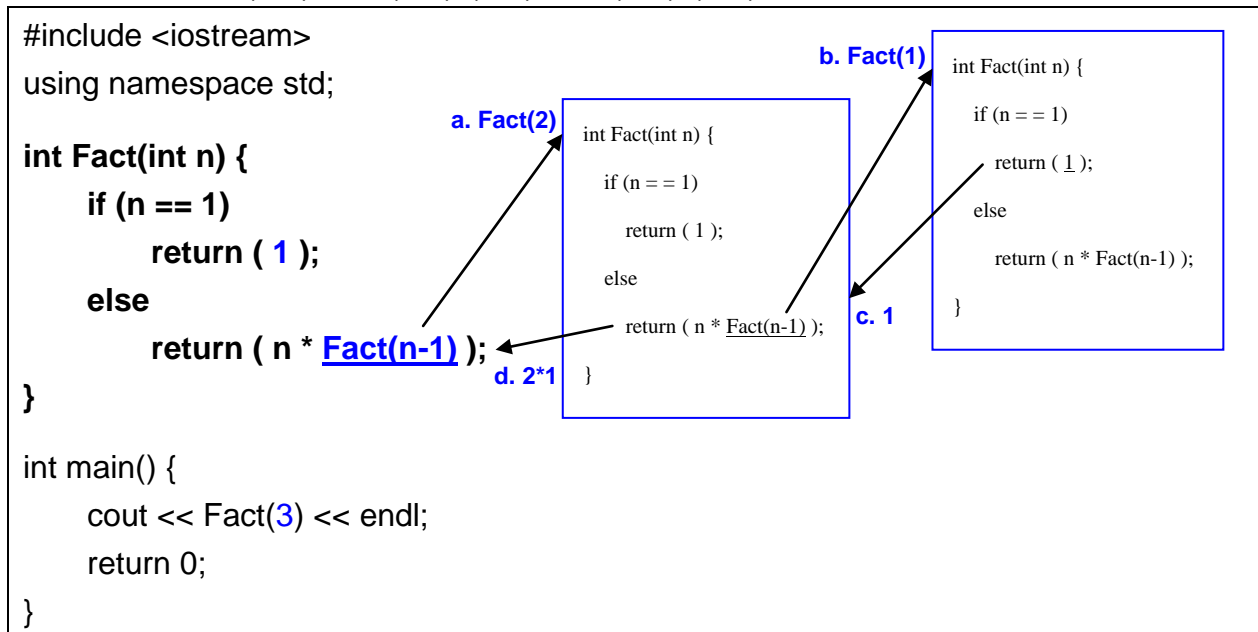
練習：利用遞歸，將一個十進制數 K 轉化為 N 進制數 ($N \leq 10$)。

提示：將十進制數轉化為 N 進制數的方法是：

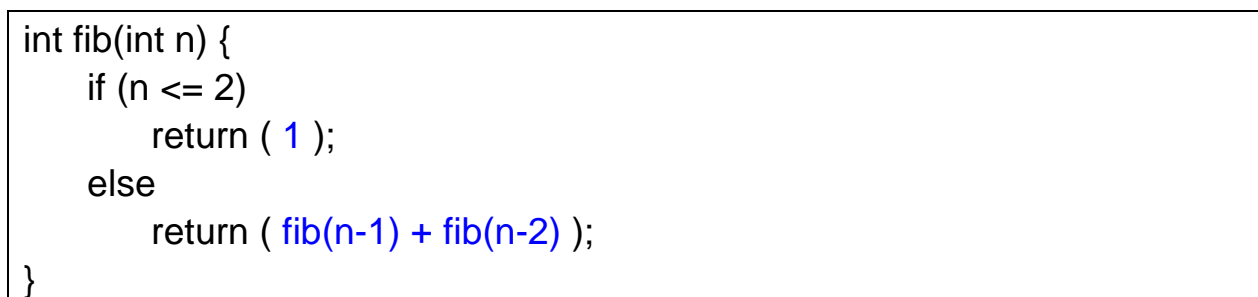
除 N ，反序取餘數。

遞歸方法 (Recursive)

1. 階乘：
 $0! = 1$
 $1! = 1$
 $2! = 2 * 1$
 $3! = 3 * 2 * 1$
 $n! = n * (n-1)! = n * (n-1) * (n-2)! = n * (n-1) * (n-2) * \dots * 1$



2. 兔子問題：
 Fibonacci () 1,1,2,3,5,8,13,21,...
- $fib(5) = 5$
 $fib(6) = 8$
 $fib(10) = ?$
- $$fib(n) = \begin{cases} 1 & (\text{if } n=1 \text{ or } 2) \\ fib(n-1)+fib(n-2) & (\text{if } n>2) \end{cases}$$



3. 求兩數的 HCF ***

Exercises:

1. 裴波那契數列(Fibonacci series): 1,1,2,3,5,8,13,其定義為 $f(0)=f(1)=1$,
 $f(n) = f(n-1) + f(n-2)$, n 為項數 , $n \geq 2$ 。試寫一程式 , 求該數列第 n 項之
函數 $f(n)$, $n \geq 0$ 。
2. 試寫一程序 , 用戶輸入一整數 n , 然後反序將其輸出。
3. 試用遞歸求 GCD / HCF(最大公因數)。

GCD C++ 解 :

```
int rgcd( int x, int y)
{
    if ( y != 0 )
        return rgcd( y, x%y);
    else
        return x;
}
```

GCD Free Pascal 解 :

```
function rGCD(x,y:integer):integer;
begin
    if y<>0 then
        rGCD := rGCD( y, x mod y)
    else
        rGCD := x;
end;
```