CH1 遞迴

定義：將問題化小再解決。

※ String Reverse

· sol：（遞迴呼叫前印出）

base case ： size > 0

若 size > 0 就輸出第 size 個字，再呼叫 size-1。

※ Practice：

Q： a > b ⇒ 回傳從 a 加到 b 之值.

```
sol: int sum ( int a, int b) {
         if( a >= b) { // base case
             return a + sum( a-1 , b);
         } // end if
     } // end sum()
```

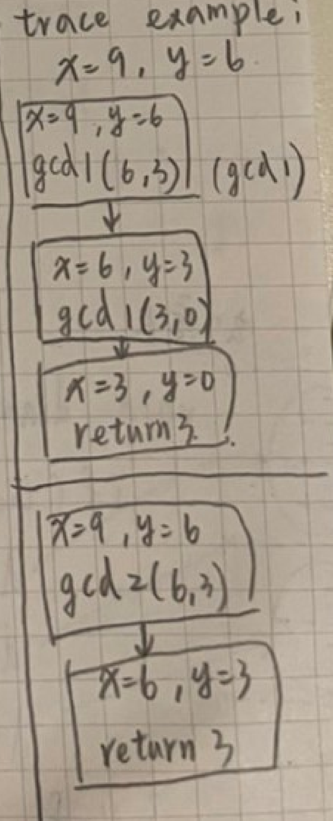
※ 問題：最大公因數

· sol 1：

```
int gcd1 (int x, int y) {
    if ( y == 0) return x; // base case
    else if ( y > x) return gcd1 ( x, y % x);
    else return gcd1 ( y , x % y);
} // end gcd1
```

· sol 2：

```
int gcd2 ( int x, int y) {
    if !(x % y) return y;
    else return gcd2 ( y, x % y)
} // end gcd2
```



△ result : gcd2 is more efficient whenever x ≥ y.

※問題: binary Search

· sol: (找第k小)

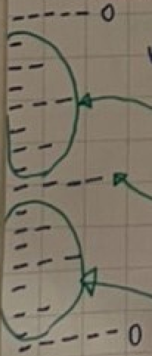
```
int ksmall (k: int, anArray: ArrayType, first: int, last: int) {
    if (k < pivotIndex - first + 1)        ⇒ first 的位置在第k個後面
        return kSmall (k, anArray, first, pivotIndex - 1)
    if (k == pivotIndex - first + 1)   ⇒

        return
    else    return
```

※河內塔 (Tower of Hanoi)

```
void hanoi (int n, int p1, int p2, int p3) {
                    ↑       ↓       ↓       ↓
                   個數     起點     輔助     終
    if (n == 1) //結束
        cout << "move" << p1 << "to" << p3 << endl;
    else {
        hanoi (n-1, p1, p3, p2);
        cout << "move" << p1 << "to" << p3 << endl;
        hanoi (n-1, p2, p1, p3);
    } //
}
```

※刻度尺



```
void drawTicks (int tickLength) {
    if (tickLength > 0) {
        drawTicks (tickLength - 1);
        drawOneTick (tickLength, -1); //畫tickLength個"—"
        drawTicks (tickLength - 1);
    } //if
} //drawTicks
```

※費氏數列

```
int Fibonacci (int i, int num1,int num2){
    if(i==0) return num1;
    return Fibonacci(i-1,num2, num1+num2);
}// end F()
```

※尾端遞迴

尾遞迴 v.s. 遞迴

```
int iterative (int n){                  int recursive(int n) {
    int pre =1;                             if (n<=2) return 1;
    int cur =1;                             else return rabbit(n-1)
    int sum =1;                                      +rabbit(n-2);
    for (int i=3;i<=n;++i){             }// int()
        sum= pre + cur;
        pre = cur;
        cur = sum;
    }// for
    return sum
}// int()
```