* 1. 获取n到m之间所有的水仙花数（水仙花数：由3个数字组成的如153，1^3+5^3+3^3=153，每位的数字的立方相加等于数字本身，就是水仙花数）

*/\*\*  
 \** ***@param*** *$n  
 \** ***@param*** *$m  
 \** ***@return*** *array  
 \* 求n,m之间的水仙花数  
 \*/***function** flower($n,$m){  
 //判断$n,$m是否为三位数 否则返回false  
 **if**(strlen($n) != 3 || strlen($m) != 3) **return false**;  
 //比较$n $m 大小 如$n>$m 互换值  
 **if**($n > $m){  
 $n = $n ^ $m;  
 $m = $n ^ $m;  
 $n = $n ^ $m;  
 }  
 //存放水仙花数  
 $arr = [];  
 **for** ($i = $n; $i <= $m; $i++){  
 $bai = floor($i/100); //获取百位数字  
 $shi = floor($i%100/10); //获取十位数字  
 $ge = $i%10; //获取个位数字  
 //计算判断是否为水仙花数  
 $sum = pow($bai,3) + pow($shi,3) + pow($ge,3);  
 **if**($sum == $i){  
 $arr[] = $i; //存入数组  
 }  
 }  
 **return** implode(',',$arr);  
}

* 1. 获取字符串中首先出现三次的那个英文字符(需要区分大小写)

*/\*\*  
 \** ***@param*** *$str  
 \** ***@return*** *bool  
 \* 字符串中首先出现三次的那个英文字符  
 \*/***function** demo2($str){  
 $len = strlen($str); //字符串总长度  
 //定义数组存每个字符出现的次数  
 $arr = [];  
 **for** ($i = 0; $i < $len; $i++){  
 **if**(**isset**($arr[$str[$i]])){ //判断该下标是否存在  
 $arr[$str[$i]]++; //出现次数加1  
 }**else**{  
 $arr[$str[$i]] = 1; //第一次出现  
 }  
 //判断有无三次字符出现  
 **if**($arr[$str[$i]] == 3){  
 **return** $str[$i]; //返回该字符  
 }  
 }  
 **return false**;  
}

* 1. 判断一个字符串是否为回文字符串，回文字符串是指从头往后读与从后往前读是同样的顺序，如“abba”

*/\*\*  
 \** ***@param*** *$str  
 \** ***@return*** *bool  
 \* 判断一个字符串是否为回文字符串，回文字符串是指从头往后读与从后往前读是同样的顺序，如“abba”  
 \*/***function** demo3($str){  
 //确定字符串长度  
 $len = strlen($str);  
 //创建空串 用于存逆序字符串  
 $rev = '';  
  
 **for** ($i = $len - 1; $i >= 0; $i--){  
 $rev .= $str[$i]; //拼接逆序字符串  
 }  
 //判断是否是回文字符串  
 **return** $str == $rev;  
}

* 1. 传入一个数字n，返回1到n之间的斐波那契数列（斐波那契数列：1 1 2 3 5 8 13....每一个值都是前两个值的和）

*/\*\*  
 \** ***@param*** *$n  
 \** ***@return*** *string  
 \* 传入一个数字n，返回1到n之间的斐波那契数列（斐波那契数列：1 1 2 3 5 8 13....每一个值都是前两个值的和）  
 \*/***function** demo4($n){  
 //创建空数组用于存斐波那契数列每项的值  
 $arr = [];  
 **for** ($i = 0; $i < $n; $i++){  
 **if**($i == 0 || $i == 1){ //前两项数列值为1  
 $arr[] = 1;  
 **continue**;  
 }  
 //每一个值都是前两个值的和  
 $arr[] = $arr[$i - 1] + $arr[$i - 2];  
 }  
 **return** implode(',',$arr);  
}

* 1. 传入一个十进制数字，返回数字对应的英文字母：  
      例： 1 = a ； 2 = b； 26 = z； 27 = aa； 52 = az； 53 = ba； 703 = aaa；

*/\*\*  
 \** ***@param*** *$num  
 \** ***@return*** *string  
 \* 传入一个十进制数字，返回数字对应的英文字母：  
 \* 例： 1 = a ； 2 = b； 26 = z； 27 = aa； 52 = az； 53 = ba； 703 = aaa；  
 \*/***function** demo5($num){  
 //创建a-z数组  
 $letter = range('a','z');  
 //获取数组长度  
 $len = count($letter);  
  
 //用于存结果字符  
 $arr = [];  
 **while**($num){  
 $shang = floor($num/$len); //获取商  
 $yu = $num % $len; //获取第一位的值  
  
 //判断是否为0  
 **if**($yu == 0){  
 $shang--; //商减一  
 array\_unshift($arr, $letter[$len - 1]); //该位存z  
 }**else**{  
 array\_unshift($arr, $letter[$yu - 1]); //存该余数对应的字母  
 }  
 //赋值给num 开启下次循环  
 $num = $shang;  
 }  
 **return** implode('',$arr);  
}

* 1. 传入一个数字n代表台阶的个数，每次只能走1阶或者2阶台阶，返回走到第n阶台阶一共有多少种走法

*/\*\*  
 \** ***@param*** *$n  
 \** ***@return*** *int  
 \* 传入一个数字n代表台阶的个数，每次只能走1阶或者2阶台阶，返回走到第n阶台阶一共有多少种走法  
 \*/***function** demo6($n)  
{  
 $one = $two = 0;  
 **for** ($i = 0; $i < $n; $i++){  
 **if**($i == 0 || $i == 1){ //前两项数列值为1  
 $one = 0;  
 $two = 1;  
 }  
 //每一个值都是前两个值的和  
 $three = $one + $two;  
 $one = $two;  
 $two = $three;  
 }  
 **return** $three;  
}