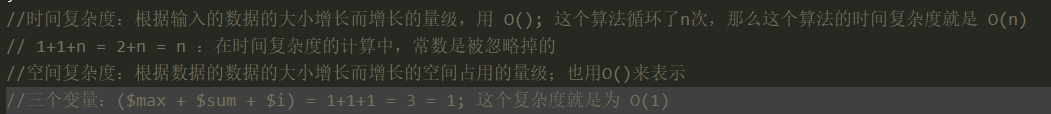
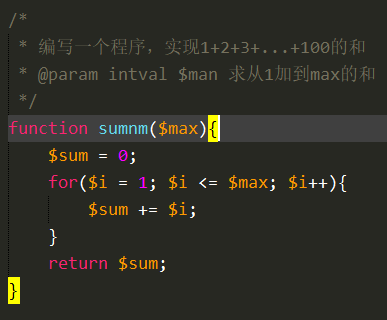
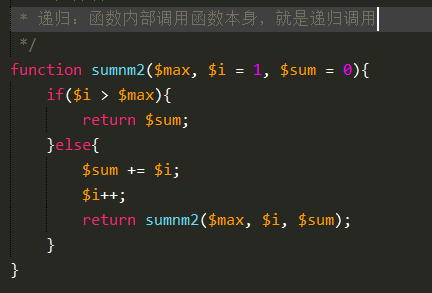
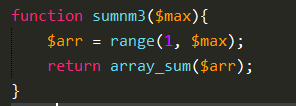
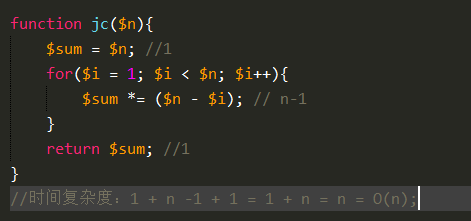
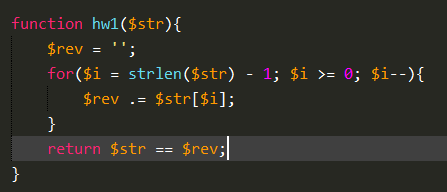
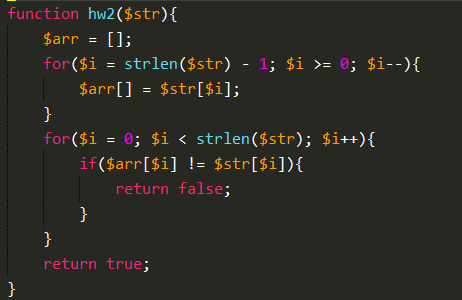
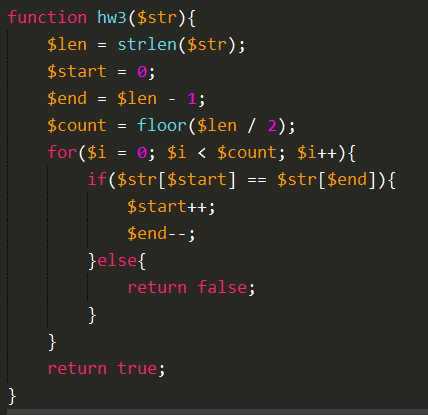
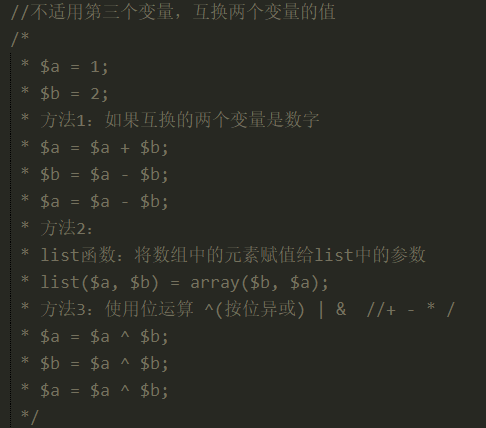
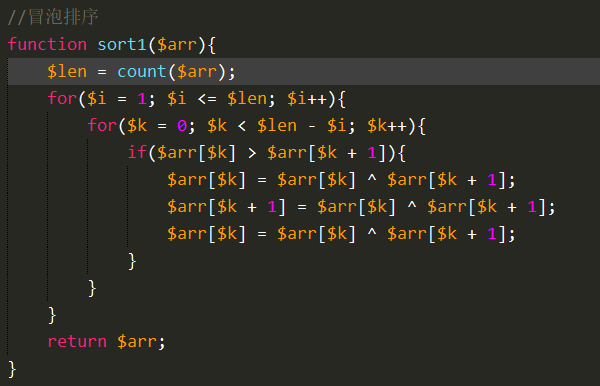
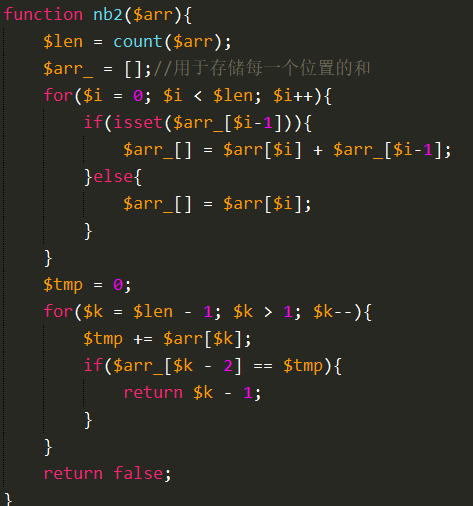
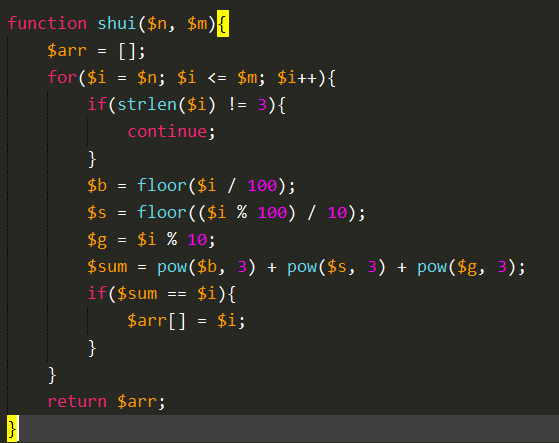
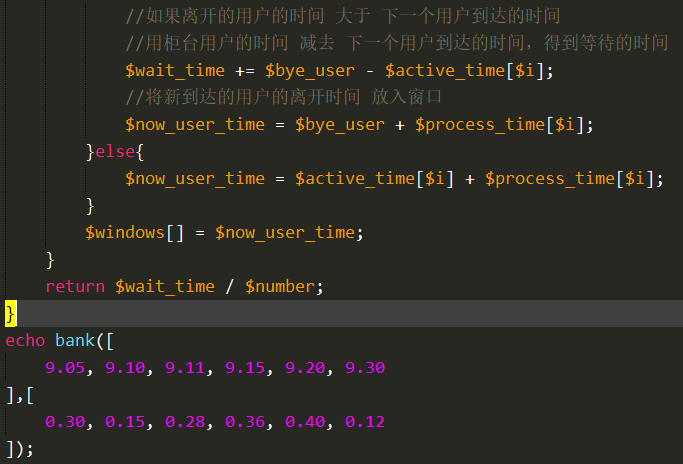
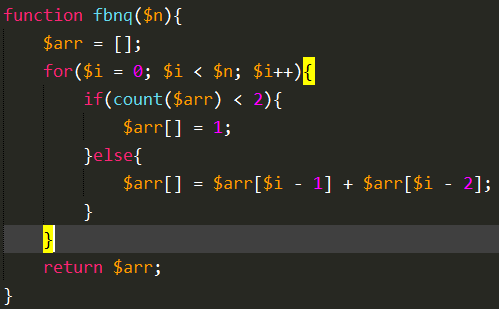
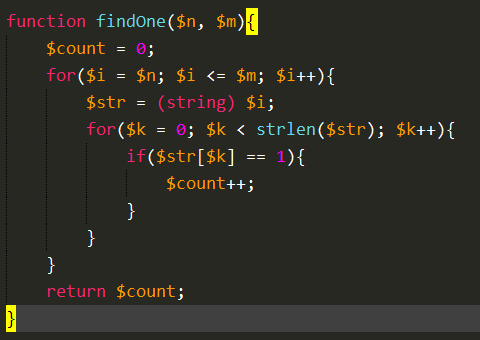
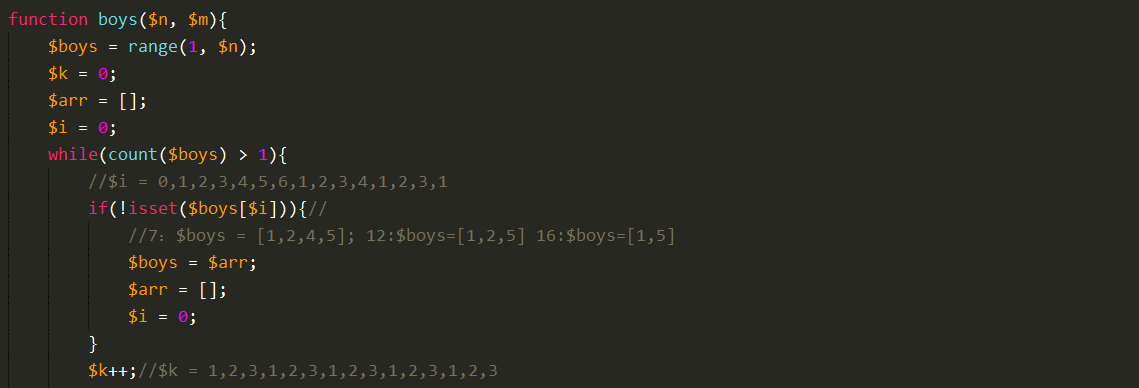
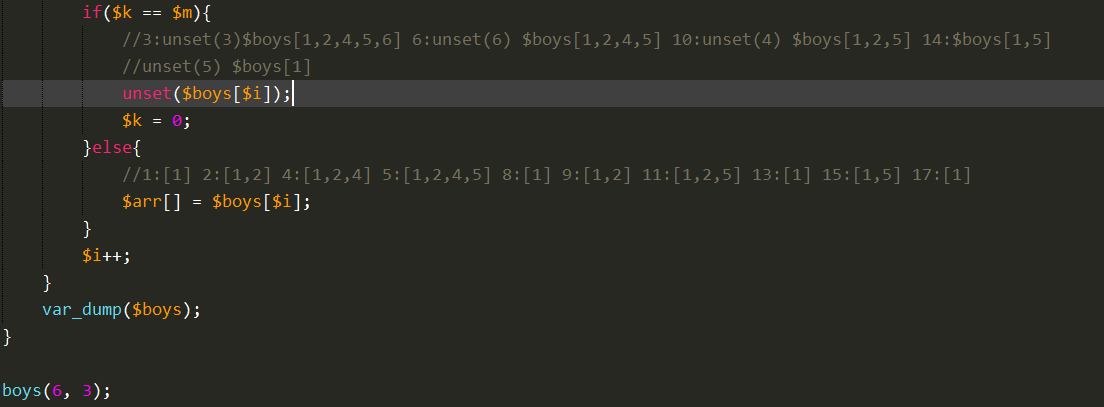
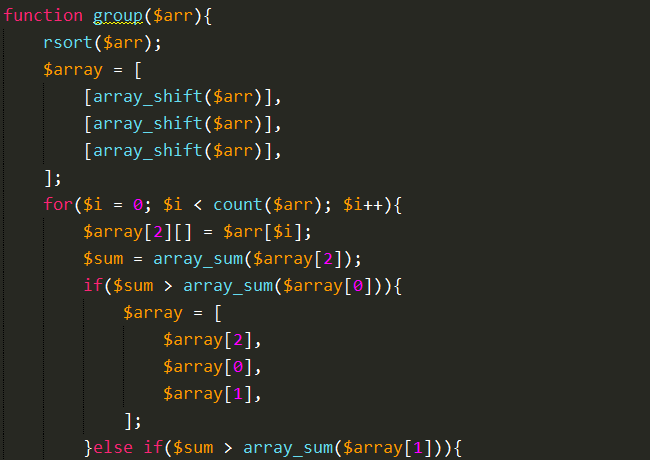
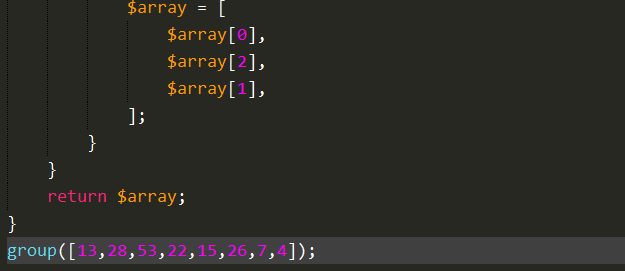
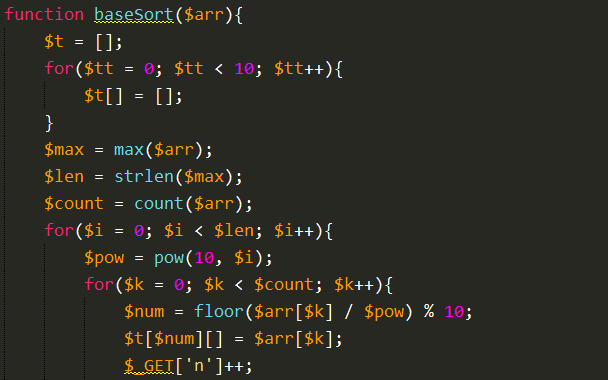
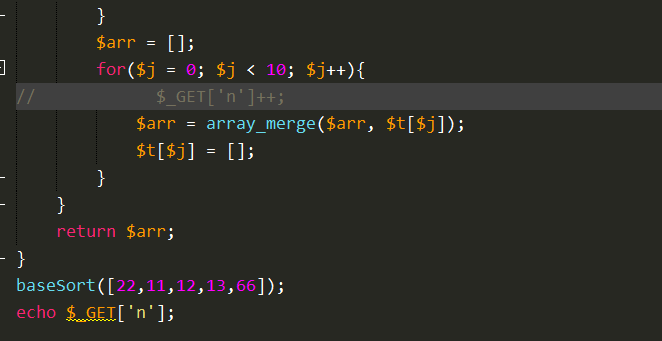
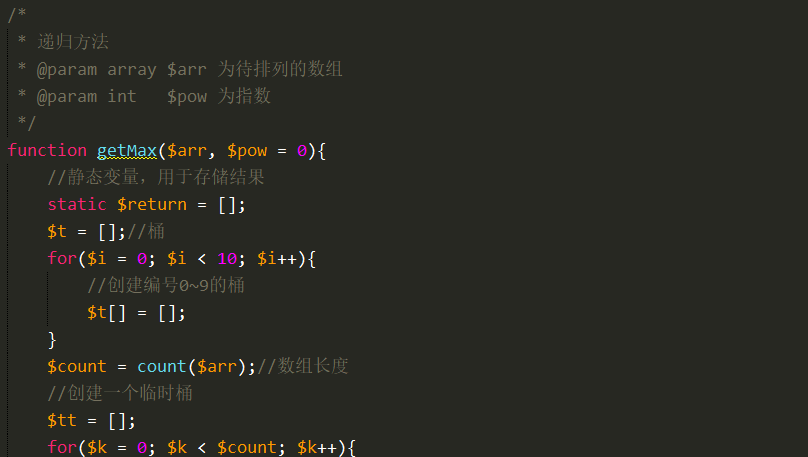
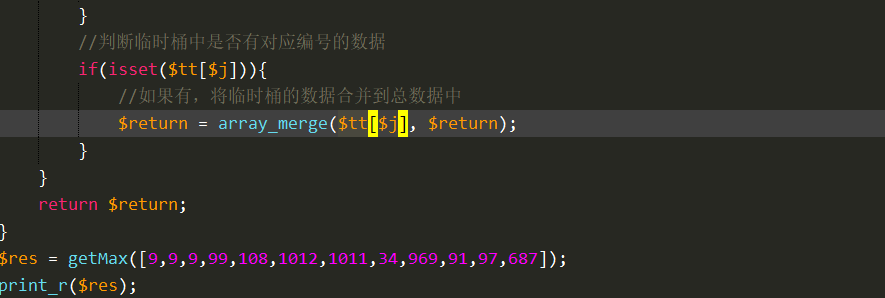
# 算法攻略



1. 编写一个程序，实现1+2+3+...+100的和；
   1. 思路：
      1. 创建变量，用于存储结果
      2. 循环，定义循环的次数
      3. 定义一个初始的值，对这个值进行自增，然后与变量相加
   2. 实现代码：
      1. 方式1：  
         
      2. 方式2：（递归）  
         
      3. 方式三：  
         
2. 编写一个程序，实现N的阶乘，公式为：N! = N \*（N-1）\*（N-2）\*...\*1
   1. 思路：
      1. 创建一个变量用于存储结果
      2. 创建变量n，控制循环的次数
      3. 创建一个增量，自增存储每次计算的数字
   2. 代码实现：  
      
3. 判断是否是回文字符串
   1. 思路1：
      1. 将字符串反转
      2. 反转后进行对比是否一致
      3. 代码：  
           
         
   2. 思路2：
      1. 字符串的首位和末尾进行对比
      2. 如果一致，首位+1.末位-1，再次对比
      3. 如果不一致，说明不是回文
      4. 循环次数定义：比如 123321循环3次，12321循环2次
   3. 实现：  
      
4. 排序算法：
   1. 冒泡、快排、基数、选择
   2. 不使用第三个变量互换两个变量的值  
      
   3. 冒泡排序：
      1. 思路：
         1. 将数列中的每一个元素进行一一对比，小的下降，大的上升
      2. 实现：  
         
   4. 快速排序：
      1. 思路：
         1. 取出数中的一个元素 作为中间点
         2. 创建两个数组
         3. 将原数组中的元素循环与中间点进行对比
         4. 大的放在右边的数组中，小的放在左边的数组中
         5. 再次处理左右两个数组，直到数组长度为1
      2. 实现：  
         
5. 取中间值
   1. 实现：  
      
6. 水仙花：  
   
7. 银行柜台
   1. 题目：银行有四个柜台，给定两个数组，一个是用户到达的时间，一个是用户办理业务耗费的时间，求所有用户平均等待的时间
   2. 代码  
        
      
8. 题目：给定一个数字，将之转换为对应的英文字母  
    1=a; 2=b; 26=z; 27=aa; 52=az, 53=ba
   1. 代码：  
      
9. 斐波那契数列
   1. 代码：  
      
10. 不适用 + - \* / 实现 求 a与b的和
    1. 思路：使用二进制进行加法计算
       1. 不考虑进位的情况：1+1 = 0; 1+0 = 1; 0+1 = 1;类似位运算种的 ^ 异或运算
       2. 考虑进位的情况： 1+1 = 1; 0+1=0; 1+0=0; 类似位运算种的 & 按位与，在左移以为就是要进的位数
       3. 如果不需要进位，也就是说第二个的运算结果等于0， 那么第一次运算就是和，
       4. 如果需要进位，那么对第一与第二的结果进行再次的一二步运算，直到第二次的运算结果为0，求出a与b的和
    2. 代码：  
       
11. 计算n~m之间，数字1出现的次数
    1. 思路：
       1. 循环n~m的数字
       2. 然后将对应的数字转换为字符串，进行循环，判断字符串种每一个元素是否等于1，然后进行计数
    2. 代码：  
       
12. 给定两个参数n，m，从1~n开始数数，数到m删除，从下一个元素接着数，再次数到m删除，求剩下的数字
    1. 代码：  
         
       
13. 给定一个数组，包含n个数字，将数组分成三份，每一份的和尽可能相近
    1. 思路：
       1. 首相将数组从大到小进行排序
       2. 取出最大的三个数字，作为待分配的数据
       3. 将第四个数字与第二个数字相加
       4. 然后将待分配数组重新排序，从大到小
       5. 然后再将下一个元素与最小的相加
       6. 直至数组循环结束
    2. 代码：  
         
       
14. 基数排序
    1. 思路：
       1. 创建0~9个桶，分别编号0~9
       2. 依次根据数字的个位、十位、百位。。。对数据进行处理
       3. 根据每一位的数字，将之放在对应编号的桶里
       4. 将桶中的数据依次取出，进行下一次处理，同时清空桶内的数据
    2. 代码：  
         
       
15. 给定一个数组，数组中包含n个数字，返回能够组成的最大数字
    1. 思路
       1. 建立10个桶，分别0~9编号
       2. 循环取出每个数值，并获取每个数值的最高位的数字
       3. 依次放入桶中
       4. 如果桶中数字个数超过1个，那么将这个桶中的数据从第二步再一次
       5. 直到所有的桶都剩一个元素为止
       6. 倒序将桶中的数据取出并拼接 组成最大的数字
    2. 代码：  
         
         
       
16. 抽奖算法
    1. 思路：
       1. 首先计算总概率
       2. 循环所有奖品，从概率最低的奖品开始
       3. 生成随机数，由0~当前总概率
       4. 判断当前随机数是否小于当前奖品的概率，如果小于，返回中奖信息
       5. 否则，使用当前总概率减去当前奖品的概率，进入下一次循环
       6. 直至中奖为止
    2. 代码：  
       
17. 红包算法：
    1. 思路：
       1. 获取总金额p以及红包个数n
       2. 根据公式(p/n)\*2 = max,单个红包的最大值
       3. 计算最大金额：
          1. p - ((n - 1) \* 0.01) < max = nmax : nmax
          2. P - ((n - 1) \* 0.01) > max = nmax : nmax = max
       4. 计算最小金额：
          1. P - ((n - 1) \* 4) > 0 = min : min
          2. P - ((n - 1) \* 4) < 0 = min : min = 0
       5. 根据最大值与最小值生成随机数为红包的金额，直至循环结束
    2. 代码：  
       