长 沙 学 院

工程实训说明书

**题目 程序设计综合训练**

**院(部) 计算机工程与应用数学学院**

**专业(班级) 软件工程 2班**

**姓名 李佳骏**

**学号 B20190103224**

**指导教师 黄娟**

**起止日期 2022.02.21-2021.03.04**

工程实训任务书

**课程名称：**程序设计工程实训

**设计题目：**程序设计综合训练

**已知技术参数和设计要求**

**第一部分 必做题**

**1、水仙花数**

[http//lx.lanqiao.cn/problem.page?gpid=T775](http://lx.lanqiao.cn/problem.page?gpid=T775)

**问题描述**

　　求出所有的“水仙花数”。所谓的“水仙花数”，是指一个3位数，其各位数字的立方和等于该数本身。

**2、计数问题**

[http//lx.lanqiao.cn/problem.page?gpid=T772](http://lx.lanqiao.cn/problem.page?gpid=T772)

**问题描述**

　　试计算在区间 1 到 n 的所有整数中，数字 x（0 ≤ x ≤ 9）共出现了多少次？例如，在 1到 11 中，即在 1、2、3、4、5、6、7、8、9、10、11 中，数字 1 出现了 4 次。

**3、高精度减法**

<http://lx.lanqiao.cn/problem.page?gpid=T749>

**问题描述**

　　高精度减法

**4、分解质因数**

[http//lx.lanqiao.cn/problem.page?gpid=T748](http://lx.lanqiao.cn/problem.page?gpid=T748)

**问题描述**

　　给定一个正整数n，尝试对其分解质因数

**5、奇怪的数对**

<http://lx.lanqiao.cn/problem.page?gpid=T2631>

**问题描述**

　　给定N, A0, B0，求出所有数对(A, B)使得如果 A0 \* x + B0 \* y 能被N整除，那么 A \* x + B \* y 也能被N整除。

**6、字符串的操作**

[http//lx.lanqiao.cn/problem.page?gpid=T725](http://lx.lanqiao.cn/problem.page?gpid=T725)

**问题描述**

　　给出一个字符串S，然后给出q条指令，分别有4种

　　1. Append str

　　表示在S的最后追加一个字符串str。

　　例

　　原字符串ABCDE

　　执行 Append FGHIJ 后

　　字符串变为ABCDEFGHIJ

　　2. Insert x str

　　表示在位置x处插入一个字符串str。（输入保证0<x<=当前字符串长度）

　　例

　　原字符串ABCGHIJ

　　执行 Insert 4 DEF 后

　　字符串变为ABCDEFGHIJ

　　3. Swap a b c d

　　表示交换从第a位到第b位的字符串与从第c位到第d位的字符串。（输入保证0<a<b<c<d<=当前字符串长度）

　　例

　　原字符串ABGHIFCDEJ

　　执行 Swap 3 5 7 9后

　　字符串变为ABCDEFGHIJ

　　4. Reverse a b

　　表示将从第a位到第b位的字符串反转。（输入保证0<a<b<=当前字符串长度）

　　例

　　原字符串ABGFEDCHIJ

　　执行 Reverse 3 7 后

　　字符串变为ABCDEFGHIJ

　　最后输出按顺序执行完指令后的字符串。

**7、质数**

[http//lx.lanqiao.cn/problem.page?gpid=T723](http://lx.lanqiao.cn/problem.page?gpid=T723)

**问题描述**

　　给定一个正整数N，请你输出N以内（不包含N）的质数以及质数的个数。

**8、成绩排序**

[http//lx.lanqiao.cn/problem.page?gpid=T557](http://lx.lanqiao.cn/problem.page?gpid=T557)

**问题描述**

　　给出n个学生的成绩，将这些学生按成绩排序，

　　排序规则，优先考虑数学成绩，高的在前；数学相同，英语高的在前；数学英语都相同，语文高的在前；三门都相同，学号小的在前

**9、快速排序**

[http//lx.lanqiao.cn/problem.page?gpid=T571](http://lx.lanqiao.cn/problem.page?gpid=T571)

**问题描述**

　　用递归来实现快速排序（quick sort）算法。快速排序算法的基本思路是假设要对一个数组a进行排序，且a[0] = x。首先对数组中的元素进行调整，使x放在正确的位置上。同时，所有比x小的数都位于它的左边，所有比x大的数都位于它的右边。然后对于左、右两段区域，递归地调用快速排序算法来进行排序。

**10、成绩排名**

[http//lx.lanqiao.cn/problem.page?gpid=T551](http://lx.lanqiao.cn/problem.page?gpid=T551)

**问题描述**

　　小明刚经过了一次数学考试，老师由于忙碌忘记排名了，于是老师把这个光荣的任务交给了小明，小明则找到了聪明的你，希望你能帮他解决这个问题。

**11、GPA**

[http//lx.lanqiao.cn/problem.page?gpid=T539](http://lx.lanqiao.cn/problem.page?gpid=T539)

**问题描述**

　　输入A,B两人的学分获取情况，输出两人GPA之差。

**12、互质数个数**

<http://lx.lanqiao.cn/problem.page?gpid=T2618>

**问题描述**

　　已知正整数x，求1~x-1中，有多少与x互质的数。（互质是指两个数最大公约数为1）

**13、N皇后问题**

<http://lx.lanqiao.cn/problem.page?gpid=T2492>

**问题描述**

在N\*N的方格棋盘放置了N个皇后，使得它们不相互攻击（即任意2个皇后不允许处在同一排，同一列，也不允许处在与棋盘边框成45角的斜线上。你的任务是，对于给定的N，求出有多少种合法的放置方法。

**14、矩阵转置**

<http://lx.lanqiao.cn/problem.page?gpid=T587>

**问题描述**

　　给定一个n×m矩阵相乘，求它的转置。其中1≤n≤20，1≤m≤20，矩阵中的每个元素都在整数类型（4字节）的表示范围内。

**15、数组求和**

<http://lx.lanqiao.cn/problem.page?gpid=T582>

**问题描述**

输入n个数，围成一圈，求连续m（m<n）个数的和最大为多少？

**16、最大公约数和最小公倍数问题**

<http://lx.lanqiao.cn/problem.page?gpid=T2100>

**问题描述**

　　输入二个正整数x0,y0(2<=x0<=100000,2<=y0<=100000),求出满足下列条件的P,Q的个数

　　条件: 1.P,Q是正整数

　　2.要求P,Q以x0为最大公约数,以y0为最小公倍数.

　　试求:满足条件的所有可能的两个正整数的个数.

**17、数的计算**

<http://lx.lanqiao.cn/problem.page?gpid=T2099>

**问题描述**

　　我们要求找出具有下列性质数的个数(包含输入的自然数n):

　　先输入一个自然数n(n<=1000),然后对此自然数按照如下方法进行处理:

1. 不作任何处理;

2. 在它的左边加上一个自然数,但该自然数不能超过原数的一半;

3. 加上数后,继续按此规则进行处理,直到不能再加自然数为止.

**18、数字反转**

<http://lx.lanqiao.cn/problem.page?gpid=T2047>

**问题描述**

　　给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零（参见样例 2） 。

**第二部分 提高题**

**1、天天向上**

[http//lx.lanqiao.cn/problem.page?gpid=T753](http://lx.lanqiao.cn/problem.page?gpid=T753)

**问题描述**

　　A同学的学习成绩十分不稳定，于是老师对他说“只要你连续4天成绩有进步，那我就奖励给你一朵小红花。”可是这对于A同学太困难了。于是，老师对他放宽了要求“只要你有4天成绩是递增的，我就奖励你一朵小红花。”即只要对于第i、j、k、l四天，满足i<j<k<l并且对于成绩wi<wj<wk<wl，那么就可以得到一朵小红花的奖励。现让你求出，A同学可以得到多少朵小红花。

**2、欧拉函数**

[http//lx.lanqiao.cn/problem.page?gpid=T752](http://lx.lanqiao.cn/problem.page?gpid=T752)

**问题描述**

　　老师出了一道难题，小酱不会做，请你编个程序帮帮他，奖金一瓶酱油

　　从1—n中有多少个数与n互质?

**3、计算超阶乘**

[http//lx.lanqiao.cn/problem.page?gpid=T751](http://lx.lanqiao.cn/problem.page?gpid=T751)

**问题描述**

　　计算1\*(1+k)\*(1+2\*k)\*(1+3\*k)\*...\*(1+n\*k-k)的末尾有多少个0，最后一位非0位是多少。

**4、区间最大和**

<http://lx.lanqiao.cn/problem.page?gpid=T2624>

**问题描述**

给定一段长度为N的整数序列A，请从中选出一段连续的子序列（可以为0）使得这段的总和最大。

**5、高精度乘法**

[http//lx.lanqiao.cn/problem.page?gpid=T722](http://lx.lanqiao.cn/problem.page?gpid=T722)

**问题描述**

计算机真是最傻的东西；他都不能计算大于10^65-1的a\*b，请你帮这个“最傻”的东西过关，否则它就真的只认识1和0了。

**6、字符串匹配**

[http//lx.lanqiao.cn/problem.page?gpid=T656](http://lx.lanqiao.cn/problem.page?gpid=T656)

**问题描述**

　　给出一个字符串和多行文字，在这些文字中找到字符串出现的那些行。你的程序还需支持大小写敏感选项当选项打开时，表示同一个字母的大写和小写看作不同的字符；当选项关闭时，表示同一个字母的大写和小写看作相同的字符。

**7、递归输出**

<http://lx.lanqiao.cn/problem.page?gpid=T594>

**问题描述**

　　编写递归函数，将组成整数的所有数字逐个输出，每个数字后面加上一个减号“-”，例如对于整数123，该函数将输出1-2-3- 。编写主函数测试该递归函数。

**8、字符串压缩**

[http//lx.lanqiao.cn/problem.page?gpid=T578](http://lx.lanqiao.cn/problem.page?gpid=T578)

**问题描述**

　　编写一个程序，输入一个字符串，然后采用如下的规则对该字符串当中的每一个字符进行压缩

　　(1) 如果该字符是空格，则保留该字符；

　　(2) 如果该字符是第一次出现或第三次出现或第六次出现，则保留该字符；

　　(3) 否则，删除该字符。

　　例如，若用户输入“occurrence”，经过压缩后，字符c的第二次出现被删除，第一和第三次出现仍保留；字符r和e的第二次出现均被删除，因此最后的结果为“ocurenc”。

**9、 奥运会开幕式**

[http//lx.lanqiao.cn/problem.page?gpid=T570](http://lx.lanqiao.cn/problem.page?gpid=T570)

**问题描述**

　　学校给高一（三）班分配了一个名额，去参加奥运会的开幕式。每个人都争着要去，可是名额只有一个，怎么办？班长想出了一个办法，让班上的所有同学（共有n个同学）围成一圈，按照顺时针方向进行编号。然后随便选定一个数m，并且从1号同学开始按照顺时针方向依次报数，1, 2, …, m，凡报到m的同学，都要主动退出圈子。然后不停地按顺时针方向逐一让报出m者出圈，最后剩下的那个人就是去参加开幕式的人。

　　要求用环形链表的方法来求解。所谓环形链表，即对于链表尾结点，其next指针又指向了链表的首结点。基本思路是先创建一个环形链表，模拟众同学围成一圈的情形。然后进入循环淘汰环节，模拟从1到m报数，每次让一位同学（结点）退出圈子。

**10、双十一抢购**

[http//lx.lanqiao.cn/problem.page?gpid=T552](http://lx.lanqiao.cn/problem.page?gpid=T552)

**问题描述**

　　一年一度的双十一又来了，某网购网站又开始了半价销售的活动。

　　小G打算在今年的双十一里尽情地购物，以享受购买的极度快感，她已经列好了她想买的物品的列表。

　　当然小G并不是出身富贵家庭，所以她网银里的钱只是一个有限的整数S（单位元）。

　　这次抢购她打算遵循这三个原则选择每一个物品

　　1.先买能“赚”最多的；

　　2.在“赚”一样多的情况下，先买最便宜的（这样买的东西就可能更多了）；

　　3.在前两条里都判断不了购买顺序的话，先购买在列表里靠前的。

　　（由于网站里还是有一部分商品并没有打五折，所以2的情况（“赚”的钱数为0）是完全可能发生的）

　　现在，在双十一的这一天，你要帮小G编写一个程序，来看看她应该去买她列表里的哪些物品。（总价格不要超过S哦）

　　要是帮她写好这个程序的话，或许你能在光棍节这一天里赢得她的芳心哦~

**11、Monday-Saturday质因子**

[http//lx.lanqiao.cn/problem.page?gpid=T546](http://lx.lanqiao.cn/problem.page?gpid=T546)

**问题描述**

　　这个问题是个简单的与数论有关的题目，看起来似乎是“求正整数的所有质因子”，但实际上并不完全是这样。

　　本题中需要定义以下几个概念

1． Monday-Saturday数

　　对于一个正整数N，如果它除以7得到的余数是1或6，则可以写成N=7k+{1,6}的形式。更形象的，我们把这样的N称作“Monday-Saturday数”，简称“MS数”。

2． Monday-Saturday因子

　　如果对于两个MS数a,b，若存在一个MS数x，使得ax=b，那么就称a是b的一个“Monday-Saturday因子”，简称“MS因子”。

3． Monday-Saturday质数

　　如果对于MS数a，满足a>1且除了1和a之外a没有其他的MS因子，那么称a是一个“Monday-Saturday质数”，简称“MS质数”。

　　注对于传统意义上的质数，若它是一个MS数，则它一定是一个MS质数。但反之不必成立，例如27，它是一个MS质数但不是传统意义上的质数。

4． Monday-Saturday质因子

　　如果对于两个MS数a,b，若满足a是b的MS因子且a是一个MS质数，那么称a是b的一个“Monday-Saturday质因子”。

　　例如27是216的一个MS质因子（216=27\*8）。

　　问题就是，给定一个MS数N，求其所有的Monday-Saturday质因子。

**12、P0602**

<http://lx.lanqiao.cn/problem.page?gpid=T2627>

**问题描述**

　　编写一个程序，输入一个4位的自然数，将组成该数的各位数字重新排列，形成一个最大数和一个最小数，之后用最大数减去最小数，得到一个新的自然数，把这个数打印出来。然后对于这个新的自然数，重复上述步骤，直到该自然数的值不再发生变化。例如，假设用户输入的自然数为1001，那么由它所形成的最大数为1100，最小数为11，因此新的自然数为1089。对于1089，由它形成的最大数为9810，最小数为189，因此新的自然数为9621。9621的最大数为9621，最小数为1269，结果为8352,。8352的最大数为8532，最小数为2358，结果为6174。6174的最大数为7641，最小数为1467，结果仍为6174，因此程序结束。

**13、 矩阵乘法**

[http//lx.lanqiao.cn/problem.page?gpid=T530](http://lx.lanqiao.cn/problem.page?gpid=T530)

**问题描述**

　　小明最近刚刚学习了矩阵乘法，但是他计算的速度太慢，于是他希望你能帮他写一个矩阵乘法的运算器。

**14、JOE的算数**

[http//lx.lanqiao.cn/problem.page?gpid=T526](http://lx.lanqiao.cn/problem.page?gpid=T526)

**问题描述**

　　有一天，JOE终于不能忍受计算a^b%c这种平凡的运算了。所以他决定要求你写一个程序，计算a^b%c。

**15、P0102**

[http//lx.lanqiao.cn/problem.page?gpid=T427](http://lx.lanqiao.cn/problem.page?gpid=T427)

**问题描述**

　　用户输入三个字符，每个字符取值范围是0-9，A-F。然后程序会把这三个字符转化为相应的十六进制整数，并分别以十六进制，十进制，八进制输出，十六进制表示成3位，八进制表示成4位，若不够前面补0。（不考虑输入不合法的情况）

**16、三进制数位和**

[http//lx.lanqiao.cn/problem.page?gpid=T425](http://lx.lanqiao.cn/problem.page?gpid=T425)

**问题描述**

　　给定L和R，你需要对于每一个6位三进制数（允许前导零），计算其每一个数位上的数字和，设其在十进制下为S。

　　一个三进制数被判断为合法，当且仅当S为质数，或者S属于区间[L,R]。

　　你的任务是给出合法三进制数的个数。

**17、9-1九宫格**

[http//lx.lanqiao.cn/problem.page?gpid=T409](http://lx.lanqiao.cn/problem.page?gpid=T409)

**问题描述**

　　九宫格。输入1-9这9个数字的一种任意排序，构成3\*3二维数组。如果每行、每列以及对角线之和都相等，打印1。否则打印0。

**18、身份证号码升级**

[http//lx.lanqiao.cn/problem.page?gpid=T325](http://lx.lanqiao.cn/problem.page?gpid=T325)

**问题描述**

　　从1999年10月1日开始，公民身份证号码由15位数字增至18位。(18位身份证号码简介)。升级方法为

　　1、把15位身份证号码中的年份由2位(7,8位)改为四位。

　　2、最后添加一位验证码。验证码的计算方案

　　将前 17 位分别乘以对应系数 (7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2) 并相加，然后除以 11 取余数，0-10 分别对应 1 0 x 9 8 7 6 5 4 3 2。

　　请编写一个程序，用户输入15位身份证号码，程序生成18位身份证号码。假设所有要升级的身份证的四位年份都是19××年

**19、快乐司机**

[http//lx.lanqiao.cn/problem.page?gpid=T321](http://lx.lanqiao.cn/problem.page?gpid=T321)

**问题描述**

　　"嘟嘟嘟嘟嘟嘟

　　喇叭响

　　我是汽车小司机

　　我是小司机

　　我为祖国运输忙

　　运输忙"

　　这是儿歌“快乐的小司机”。话说现在当司机光有红心不行，还要多拉快跑。多拉不是超载，是要让所载货物价值最大，特别是在当前油价日新月异的时候。司机所拉货物为散货，如大米、面粉、沙石、泥土......

　　现在知道了汽车核载重量为w，可供选择的物品的数量n。每个物品的重量为gi,价值为pi。求汽车可装载的最大价值。（n<10000,w<10000,0<gi<=100,0<=pi<=100)

**20、数组移动**

<http://lx.lanqiao.cn/problem.page?gpid=T2672>

**问题描述**

初始数组A[N]中为1,2,..,N，N个数字，现要进行M次操作，每次操作给定一个数字i，记其在数组中的位置为Bi，将A[1]..A[Bi]移到数组末尾。

**21、接水问题**

<http://lx.lanqiao.cn/problem.page?gpid=T2028>

**问题描述**

　　学校里有一个水房，水房里一共装有m 个龙头可供同学们打开水，每个龙头每秒钟的 供水量相等，均为1。 现在有n 名同学准备接水，他们的初始接水顺序已经确定。将这些同学按接水顺序从1 到n 编号，i 号同学的接水量为wi。接水开始时，1 到m 号同学各占一个水龙头，并同时打 开水龙头接水。当其中某名同学j 完成其接水量要求wj 后，下一名排队等候接水的同学k 马上接替j 同学的位置开始接水。这个换人的过程是瞬间完成的，且没有任何水的浪费。即 j 同学第x 秒结束时完成接水，则k 同学第x+1 秒立刻开始接水。若当前接水人数n’不足m， 则只有n’个龙头供水，其它m−n’个龙头关闭。 现在给出n 名同学的接水量，按照上述接水规则，问所有同学都接完水需要多少秒。

**22、阴谋**

<http://lx.lanqiao.cn/problem.page?gpid=T2953>

**问题描述**

　　第四次圣杯战争开始了。 为了收集敌人的情报，言峰绮礼命令他的使魔Assassin将自己的灵体分成n份，分别监视教堂外的长直走道。

　　Assassin每份灵体的能力不同。 第i份灵体可以监视到的区域是闭区间[ai,bi]。

　　绮礼想知道，监控范围内的区域的总长度是多少。

　　比如，第一份灵体的视野是[−1,1]，第二份灵体的视野是[0,2]，第三份灵体的视野是[3,4]。 那么绮礼能获得的全部视野是[−1,2]∪[3,4]，长度为4。

**各阶段具体要求**

1、完成第一部分必做题共计18题的练习

能够进行一定算法分析设计，完成基本的算法训练的题目，并正确提交，并完成相应的文档总结。

2、选做完成第二部分提高题练习

能够综合运用所学知识，进行算法分析与设计，完成提高题的题目，并正确提交，并完成相应的文档总结。

**设计工作量**

（1）编码测试在系统中正确提交代码并得分。

（2）设计报告**要求撰写不少于18个题目的题解分析文档，要求①说明算法思路，②绘制流程图（至少5个），③源代码，④测试结果截图，⑤系统提交得分截图**

**工作计划**

解题进度安排如下，文档编写同步进行；

1. 必做题练习（16课时）
2. 提高题练习（16课时）
3. 撰写文档（4课时）
4. 答辩考核（4课时）

**注意事项**

* + **提交文档**
* 长沙学院课程设计任务书 （每学生1份）
* 长沙学院课程设计论文 （每学生1份）
* 长沙学院课程设计鉴定表 （每学生1份）

目录

**[一.必做题 19](#_Toc682872927)**

[1、水仙花数 19](#_Toc1682025686)

[2、计数问题 22](#_Toc1078349553)

[3、高精度减法 26](#_Toc1476553317)

[4、分解质因数 28](#_Toc765281386)

[5、奇怪的数对 32](#_Toc1697673153)

[6、字符串的操作 33](#_Toc1202653865)

[7、质数 35](#_Toc1420912455)

[8、成绩排序 40](#_Toc1815047803)

[9、快速排序 42](#_Toc1209363022)

[10、成绩排名 46](#_Toc2143219408)

[11、GPA 47](#_Toc979416376)

[12、互质数个数 50](#_Toc606439326)

[13、N皇后问题 51](#_Toc1917318659)

[14、矩阵转置 55](#_Toc129810965)

[15、数组求和 57](#_Toc2007097144)

[16、最大公约数和最小公倍数问题 59](#_Toc1161211946)

[17、数的计算 61](#_Toc410423984)

[18、数字反转 63](#_Toc1027374976)

**[二.选做题 65](#_Toc477275374)**

[1、天天向上 65](#_Toc1745802444)

[2、欧拉函数 69](#_Toc749829798)

[3、计算超阶乘 71](#_Toc961679275)

[4、区间最大和 73](#_Toc175989323)

[5、高精度乘法 75](#_Toc1121222444)

[6、字符串匹配 76](#_Toc554068363)

[7、递归输出 78](#_Toc1487848591)

[8、字符串压缩 80](#_Toc2094416255)

[9、奥运会开幕式 82](#_Toc1604195900)

[10、双十一抢购 84](#_Toc524703721)

[11、Monday-Saturday质因子 87](#_Toc506062926)

[12、P0602 89](#_Toc139585179)

[13、矩阵乘法 92](#_Toc59245759)

[14、JOE的算数 94](#_Toc1584412479)

[15、P0102 96](#_Toc1616138496)

[16、三进制数位和 98](#_Toc824527145)

[17、9-1九宫格 100](#_Toc1134601984)

[18、身份证号码升级 102](#_Toc671308713)

[19、快乐司机 104](#_Toc97955953)

[20、数组移动 106](#_Toc802166140)

[21、接水问题 109](#_Toc1880671736)

[22、阴谋 111](#_Toc93691713)

# 一.必做题

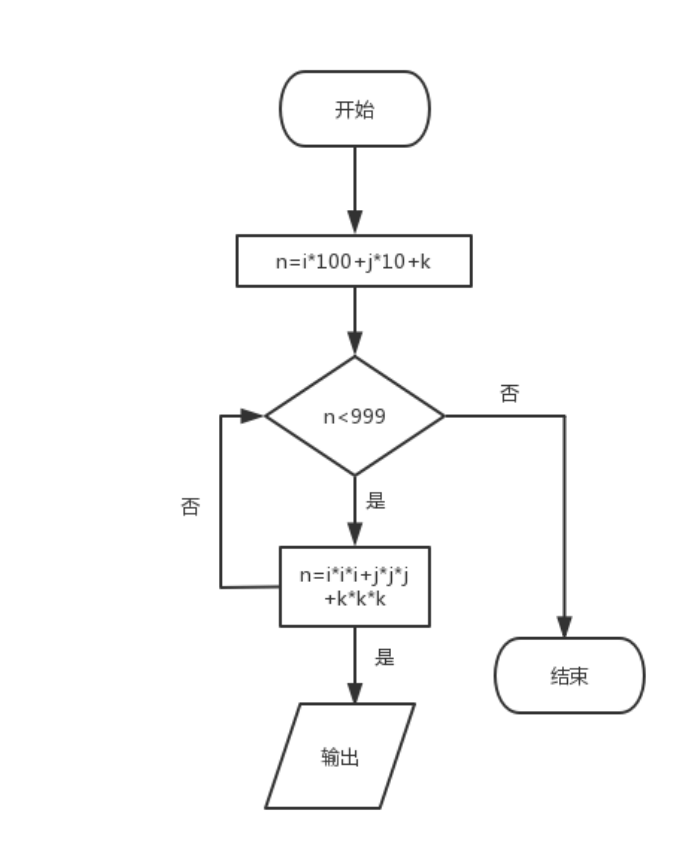
## 1、[水仙花数](http://lx.lanqiao.cn/problem.page?gpid=T775)

求出所有的“水仙花数”。所谓的“水仙花数”，是指一个3位数，其各位数字的立方和等于该数本身。

### 算法思路:

枚举100-999,符合要求输出即可

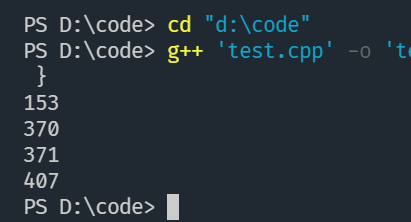
### 流程图:



### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
int n, m, t;  
void solve(){  
 for (int i = 100; i <= 999; i ++){  
 int t = i;  
 int a = t % 10; t /= 10;  
 int b = t % 10; t /= 10;  
 int c = t % 10; t /= 10;  
 if(inv(a,3) + inv(b,3) + inv(c,3) == i) cout << i << endl;  
 }  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



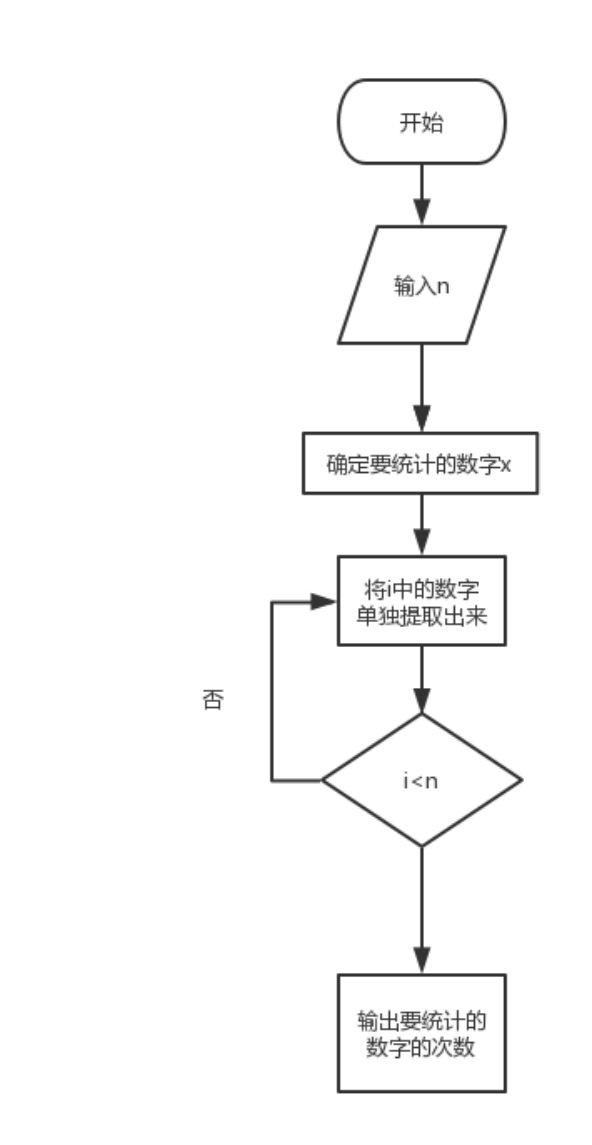
## 2、[计数问题](http://lx.lanqiao.cn/problem.page?gpid=T772)

试计算在区间 1 到 n 的所有整数中，数字 共出现了多少次？例如，在 1到 11 中，即在 1、2、3、4、5、6、7、8、9、10、11 中，数字 1 出现了 4 次。

### 算法思路:

将1到n之间的数依次进行%10和/10的操作将他们每个位上的数都分离开来，从而来计算所求数所出现的次数

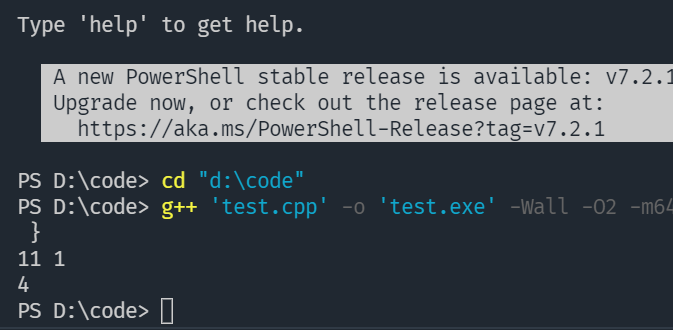
### 流程图:



### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
void solve(){  
 int n, x, cnt = 0;  
 cin >> n >> x;  
 for (int i = 1; i <= n; i++){  
 int t = i;  
 while(t){  
 if(x == t % 10) cnt++;  
 t /= 10;  
 }  
 }  
 cout << cnt << endl;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 3、[高精度减法](http://lx.lanqiao.cn/problem.page?gpid=T749)

高精度减法

### 算法思路:

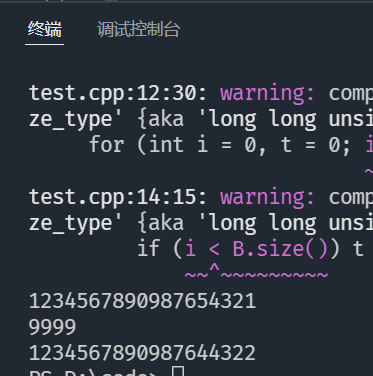
模拟减法过程，使用2个vector来存储大数，最后将结果以vector返回

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
vector<int> A, B, C;  
// C = A - B, 满足A >= B, A >= 0, B >= 0  
vector<int> sub(vector<int> &A, vector<int> &B){  
 vector<int> C;  
 for (int i = 0, t = 0; i < A.size(); i ++ ){  
 t = A[i] + t;  
 if (i < B.size()) t -= B[i];  
 C.push\_back((t + 10) % 10);  
 if (t < 0) t = -1;  
 else t = 0;  
 }  
  
 while (C.size() > 1 && C.back() == 0) C.pop\_back();  
 return C;  
}  
void solve(){  
 string a, b;  
 cin >> a >> b;  
 for(int i = a.size() - 1; i >= 0; i--) A.push\_back(a[i] - '0');   
 for(int i = b.size() - 1; i >= 0; i--) B.push\_back(b[i] - '0');   
  
 auto C = sub(A, B);  
 for(int i = C.size() - 1; i >= 0; i--) cout << C[i];  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 4、[分解质因数](http://lx.lanqiao.cn/problem.page?gpid=T748)

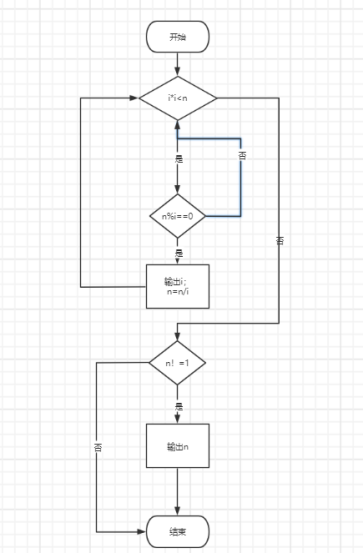
给定一个正整数n，尝试对其分解质因数

### 算法思路:

（1）输入一个数n首先利用for循环从2到n的开二次方遍历（为了防止运行超时），里面套用一个while循环当n%i==0时，输出i，并且n=n/i；

（2）由于i是从2到n的开二次方遍历，那么最后还会有一个质因数未输出，则在末尾还要输出最后一个质因数（与运算后的n相等）

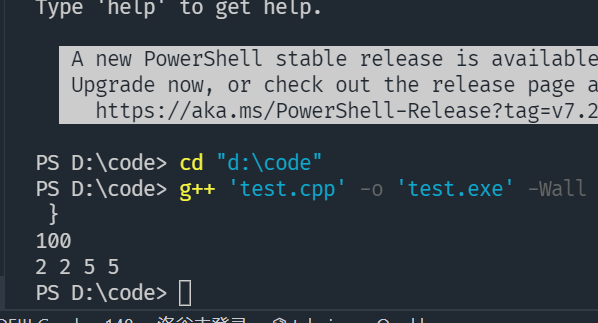
### 流程图:



### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
void solve(){  
 int n; cin >> n;  
 int N = n;  
 for (int i = 2; i \* i <= N; i ++){  
 while(n % i == 0){  
 cout << i << " ";  
 n /= i;  
 }  
 }  
 if(n > 1) cout << n;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 5、[奇怪的数对](http://lx.lanqiao.cn/problem.page?gpid=T2631)

给定，求出所有数对A0 \* x + B0 \* y A \* x + B \* y 也能被N整除。

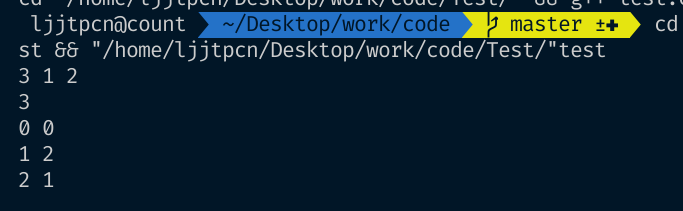
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
#define ll long long  
using namespace std;  
  
ll n, a, b;  
vector<pair<int, int> > ans;  
   
int gcd(int a, int b) {  
 if (!b) return a;  
 return gcd(b, a % b);  
}  
   
int lcm(int a, int b) {  
 return a / gcd(a, b) \* b;  
}  
   
int main() {  
 while (~scanf("%d", &n)) {  
 scanf("%d%d", &a, &b);  
 ans.clear();  
 a %= n, b %= n;  
 ll x = n / gcd(a, n);  
 ll y = n / gcd(b, n);  
 ll d = lcm(x, y);  
 for (int i = 0; i < d; i++)  
 ans.push\_back({a \* i % n, b \* i % n});  
 sort(ans.begin(), ans.end());  
  
 printf("%d\n", d);  
 for (int i = 0; i < d; i++)  
 printf("%d %d\n", ans[i].first, ans[i].second);  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 6、[字符串的操作](http://lx.lanqiao.cn/problem.page?gpid=T725)

给出一个字符串S，然后给出q条指令，分别有4种,最后输出按顺序执行完指令后的字符串。

### 算法思路:

由于这道题涉及到较多的字符串操作，可以采用C++库函数来较少代码的编辑。

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
int main() {  
 string s;  
 int n;  
 cin>>s>>n;  
 string ss[n];  
 for(int i=0;i<n;i++){  
 cin>>ss[i];  
 if(ss[i]=="Append"){  
 string str;  
 cin>>str;  
 s=s+str;  
 }else if(ss[i]=="Insert"){  
 int x;  
 string str;  
 cin>>x>>str;  
 string str1,str2;  
 str1=s.substr(0,x-1);  
 str2=s.substr(x-1,s.size()-x+1);  
 s=str1+str+str2;  
 }else if(ss[i]=="Swap"){  
 int a,b,c,d;  
 cin>>a>>b>>c>>d;  
 string str1,str2,str3,str4,str5;  
 str1=s.substr(0,a-1);  
 str2=s.substr(a-1,b-a+1);  
 str3=s.substr(b,c-b-1);  
 str4=s.substr(c-1,d-c+1);  
 str5=s.substr(d,s.size()-d);  
 s=str1+str4+str3+str2+str5;  
 }else if(ss[i]=="Reverse"){  
 int a,b;  
 cin>>a>>b;  
 string str;  
 string str1,str2;  
 str1=s.substr(0,a-1);  
 str2=s.substr(b,s.size()-b);  
 str=s.substr(a-1,b-a+1);  
 reverse(str.begin(),str.end());  
 s=str1+str+str2;  
 }  
 }  
 cout<<s;  
 return 0;  
}

### 运行截图:



### 提交记录:



## 7、[质数](http://lx.lanqiao.cn/problem.page?gpid=T723)

给定一个正整数N，请你输出N以内（不包含N）的质数以及质数的个数。

### 算法思路:

（1）输入给定的正整数n，定义sum为n以内的质数，先令其等于0；

（2）定义i从2到n-1循环；

（3）定义j从2到i-1循环，如果j<=i-1,则执行步骤（4），否则返回步骤（5）；

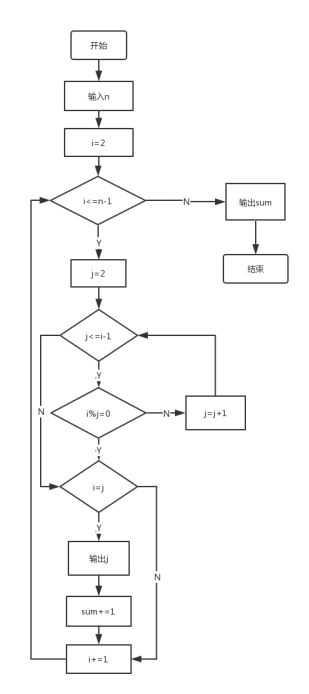
（4）如果i%j=0，则进行步骤（5），否则执行j=j+1，并返回步骤（3）；

（5）如果j=i，则输出j，sum=sum+1；

（6）i=i+1；

（7）如果i<=n-1,则执行步骤（3），否则输出sum；

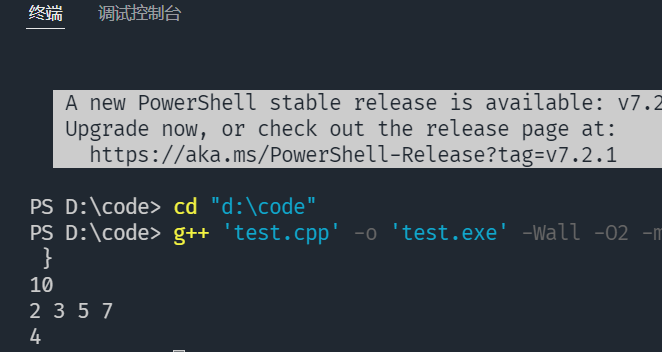
### 流程图:



### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
int st[1010];  
void solve(){  
 int n; cin >> n;  
 int cnt = 0;  
 for (int i = 2; i < n; i++){  
 if(!st[i]){  
 for (int j = i << 1; j < n; j += i){  
 st[j] = 1;  
 }  
 cout << i << " "; cnt ++;  
 }  
 }  
 cout << endl << cnt;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 8、[成绩排序](http://lx.lanqiao.cn/problem.page?gpid=T557)

给出n个学生的成绩，将这些学生按成绩排序，排序规则，优先考虑数学成绩，高的在前；数学相同，英语高的在前；数学英语都相同，语文高的在前；三门都相同，学号小的在前

### 算法思路:

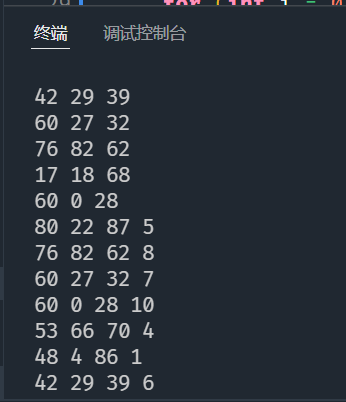
利用结构体排序自定义规则即可

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
struct node{  
 int id, math, english, chinese;  
}nd[110];  
  
void solve(){  
 int n; cin >> n;  
 for (int i = 0; i < n; i++){  
 int a, b, c; cin >> a >> b >> c;  
 nd[i] = {i+1, a, b, c};  
 }  
 sort(nd, nd + n, [](node a, node b){  
 if(a.math != b.math)  
 return a.math > b.math;  
 else if(a.english != b.english)  
 return a.english > b.english;  
 else if(a.chinese != b.chinese)  
 return a.chinese > b.chinese;  
 else  
 return a.id < b.id;  
 });  
 for (int i = 0; i < n; i++)  
 cout << nd[i].math << ' ' << nd[i].english << ' ' << nd[i].chinese << ' ' << nd[i].id << endl;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:

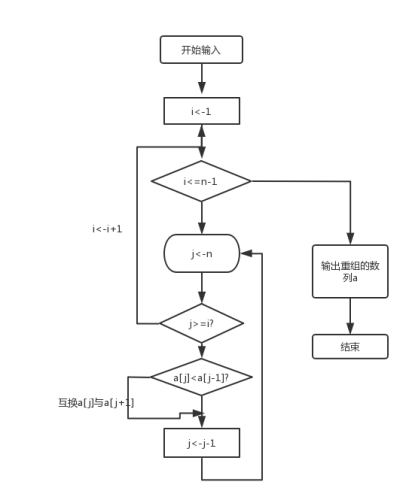


## 9、[快速排序](http://lx.lanqiao.cn/problem.page?gpid=T571)

用递归来实现快速排序（quick sort）算法。快速排序算法的基本思路是假设要对一个数组a进行排序，且a[0] = x。首先对数组中的元素进行调整，使x放在正确的位置上。同时，所有比x小的数都位于它的左边，所有比x大的数都位于它的右边。然后对于左、右两段区域，递归地调用快速排序算法来进行排序。

### 算法思路:

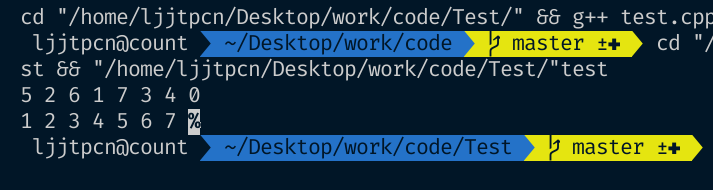
### 流程图:



### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 2010;  
  
vector<int> arr;  
void quick\_sort(vector<int> &arr, int l, int r)  
{  
 if(l >= r) return;  
  
 int i = l - 1, j = r + 1, mid = arr[l + r >> 1];  
 while(i < j){  
 do i++; while(arr[i] < mid);  
 do j--; while(arr[j] > mid);  
 if(i < j) swap(arr[i], arr[j]);  
 }  
 quick\_sort(arr, l ,j), quick\_sort(arr, j + 1, r);  
}  
  
void solve(){  
 int x;  
 while((cin >> x) && x != 0) {  
 arr.pb(x);  
 }  
   
 quick\_sort(arr, 0, arr.size() - 1);  
   
 for(int i = 0; i < arr.size(); i ++) cout << arr[i] << ' ';  
}  
  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
   
 return 0;  
}

### 运行截图:



### 提交记录:



## 10、[成绩排名](http://lx.lanqiao.cn/problem.page?gpid=T551)

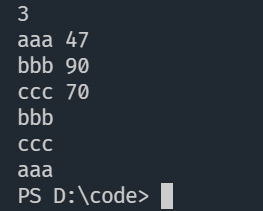
小明刚经过了一次数学考试，老师由于忙碌忘记排名了，于是老师把这个光荣的任务交给了小明，小明则找到了聪明的你，希望你能帮他解决这个问题。

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
typedef long long ll;  
const int mod = 998244353;  
const int N = 30010;  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
struct node{  
 string name;  
 int score;  
}nd[110];  
  
void solve(){  
 int n; cin >> n;  
 for (int i = 0; i < n; i++){  
 string name; int score;  
 cin >> name >> score;  
 nd[i] = {name,score};  
 }  
 sort(nd, nd + n, [](node a, node b){  
 if(a.score != b.score) return a.score > b.score;  
 else return a.name < b.name;  
 });  
 for (int i = 0; i < n; i++)  
 cout << nd[i].name << endl;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 11、[GPA](http://lx.lanqiao.cn/problem.page?gpid=T539)

输入A,B两人的学分获取情况，输出两人GPA之差。

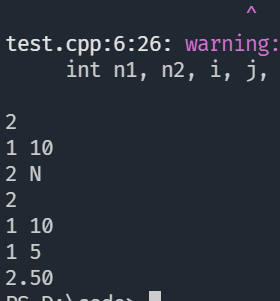
### 算法思路:

### 流程图:

### 源代码:

#include <iostream>  
#include <stdio.h>  
using namespace std;  
int main()  
{  
 int n1, n2, i, j, s, a[3];  
 double m1 = 0.0, m2 = 0.0, sum1 = 0.0, sum2 = 0.0;  
 char c[3];  
 cin >> n1;  
 while (n1--)  
 {  
 cin >> s >> c;  
 if (c[0] != 'P' && c[0] != 'N')  
 {  
 int aa = 0;  
 for (int i = 0; i < 3; i++)  
 {  
 if (c[i] >= '0' && c[i] <= '9')  
 {  
 int tt = c[i] - '0';  
 aa = aa \* 10 + tt;  
 }  
 }  
 sum1 += s \* aa;  
 m1 += s;  
 }  
 }  
 cin >> n2;  
 while (n2--)  
 {  
 cin >> s >> c;  
 if (c[0] != 'P' && c[0] != 'N')  
 {  
 int aa = 0;  
 for (int i = 0; i < 3; i++)  
 {  
 if (c[i] >= '0' && c[i] <= '9')  
 {  
 int tt = c[i] - '0';  
 aa = aa \* 10 + tt;  
 }  
 }  
 sum2 += s \* aa;  
 m2 += s;  
 }  
 }  
 if (sum1 / m1 - sum2 / m2 >= 0.01 || sum1 / m1 - sum2 / m2 <= -0.01)  
 printf("%.2lf\n", sum1 / m1 - sum2 / m2);  
 else  
 cout << "0.00" << endl;  
 return 0;  
}

### 运行截图:



### 提交记录:



## 12、[互质数个数](http://lx.lanqiao.cn/problem.page?gpid=T2618)

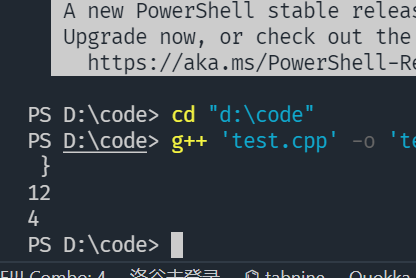
已知正整数x，求1~x-1中，有多少与x互质的数。（互质是指两个数最大公约数为1）

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, arr[N];  
  
void solve(){  
 cin >> n;  
 int N = n, ans = N, idx = 0;  
 for(int i = 2; i \* i <= N; i ++){  
 if(n % i == 0) arr[idx++] = i;  
 while(n % i == 0) n /= i;  
 }  
 if(n > 1) arr[idx++] = n;  
  
 for (int i = 0; i < idx; i++){  
 ans = ans - ans / arr[i];  
 }  
 cout << ans;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 13、[N皇后问题](http://lx.lanqiao.cn/problem.page?gpid=T2492)

在N\*N的方格棋盘放置了N个皇后，使得它们不相互攻击（即任意2个皇后不允许处在同一排，同一列，也不允许处在与棋盘边框成45角的斜线上。你的任务是，对于给定的N，求出有多少种合法的放置方法。

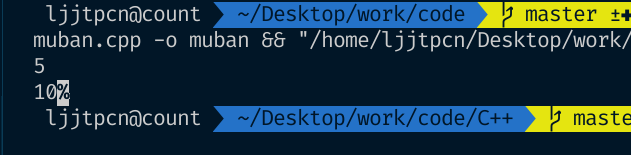
### 算法思路:

### 流程图:

### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(false); cin.tie(0); cout.tie(0)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 30010;  
  
int n, m, ans;  
int x[10], g[10][10];  
  
bool check(int row\_max)  
{  
 for(int i = 0; i < row\_max; i++){  
 if(x[row\_max] == x[i] || abs(x[row\_max] - x[i]) == abs(row\_max - i))  
 return false;   
 }  
 return true;  
}  
  
void dfs(int row)  
{  
 if(row >= n){  
 ans++;  
 return;  
 }  
  
 for(int i = 0; i < n; i++){  
 x[row] = i;  
 if(check(row)) dfs(row + 1);  
 }  
}  
  
  
void solve(){  
 cin >> n;  
 dfs(0);  
 cout << ans;  
}  
  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 14、[矩阵转置](http://lx.lanqiao.cn/problem.page?gpid=T587)

给定一个矩阵相乘，求它的转置。其中，，矩阵中的每个元素都在整数类型（4字节）的表示范围内。

### 算法思路:

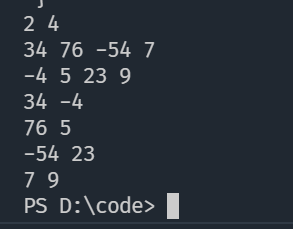
模拟

### 流程图:

### 源代码:

#include <iostream>  
using namespace std;  
int main()  
{  
 int m, n, i, j;  
 cin >> m >> n;  
 int a[m][n];  
 for (i = 0; i < m; i++)  
 {  
 for (j = 0; j < n; j++)  
 {  
 cin >> a[i][j];  
 }  
 }  
 for (i = 0; i < n; i++)  
 {  
 for (j = 0; j < m; j++)  
 {  
 cout << a[j][i] << " ";  
 }  
 cout << endl;  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 15、[数组求和](http://lx.lanqiao.cn/problem.page?gpid=T582)

输入个数，围成一圈，求连续个数的和最大为多少？

### 算法思路:

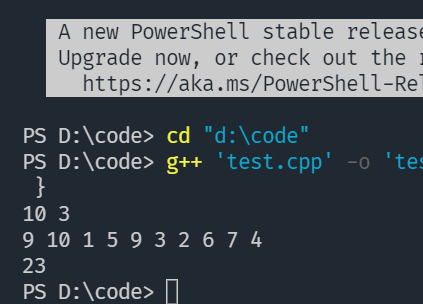
前缀和

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, m, arr[N], s[N];  
  
void solve(){  
 cin >> n >> m;  
 for (int i = 1; i <= n; i ++){  
 cin >> arr[i];  
 s[i] = arr[i] + s[i - 1];  
 }  
 int ans = 0;  
 for (int i = 1; i <= n; i ++){  
 if(i+m-1 > n){  
 int l = i, r = (i + m - 1) % n;  
 ans = max(ans, s[r] + s[n] - s[l - 1]);  
 }else{  
 int l = i, r = i + m - 1;  
 ans = max(ans, s[r] - s[l - 1]);  
 }  
 }  
 cout << ans;  
}  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 16、[最大公约数和最小公倍数问题](http://lx.lanqiao.cn/problem.page?gpid=T2100)

输入二个正整数,求出满足下列条件的P,Q的个数

条件:

试求:满足条件的所有可能的两个正整数的个数.

### 算法思路:

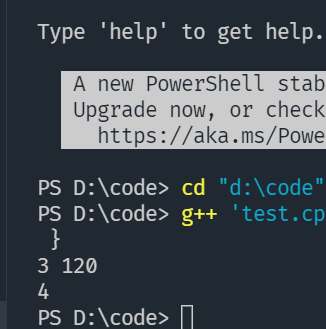
1. 两数的最小公倍数等于两数之积除以两数的最大公约数
2. 最大公约数和最小公倍数的乘积就是原两个数的积

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, m, arr[N], s[N];  
  
int gcd(int a, int b){  
 return b == 0 ? a : gcd(b, a % b);  
}  
  
void solve(){  
 int x0, y0;  
 cin >> x0 >> y0;  
  
 int cnt = 0;  
 for (int i = x0; i \* i <= x0 \* y0; i += x0){  
 if((x0\*y0) % i == 0 && gcd(i, x0\*y0 / i) == x0)  
 cnt += 2;  
 }  
 cout << cnt;  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 17、[数的计算](http://lx.lanqiao.cn/problem.page?gpid=T2099)

我们要求找出具有下列性质数的个数(包含输入的自然数n):

先输入一个自然数,然后对此自然数按照如下方法进行处理:

1. 不作任何处理;
2. 在它的左边加上一个自然数,但该自然数不能超过原数的一半;
3. 加上数后,继续按此规则进行处理,直到不能再加自然数为止.

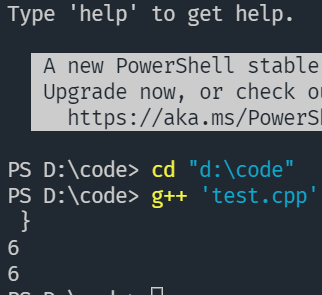
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, m, arr[N], s[N];  
  
int count(int n){  
 if(n == 1)  
 return 1;  
 int cnt = 1;  
 for (int i = 1; i <= n / 2; i++)  
 cnt += count(i);  
 return cnt;  
}  
  
void solve(){  
 int n;  
 cin >> n;  
 cout << count(n) << endl;  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 18、[数字反转](http://lx.lanqiao.cn/problem.page?gpid=T2047)

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零（参见样例 2） 。

### 算法思路:

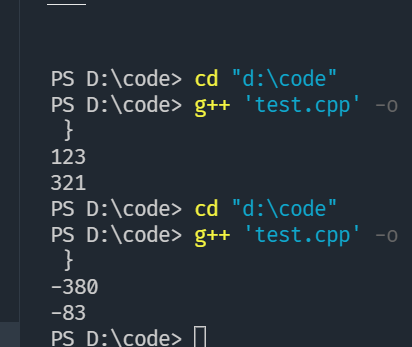
模拟数的反转过程

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, m, arr[N], s[N];  
  
void solve(){  
 string s;  
 cin >> s;  
 bool flag = false;  
 if(s[0] == '-') flag = true, cout << "-";  
  
 int i = s.size() - 1; while(s[i] == '0') i--;  
 for (; i >= (flag == true ? 1 : 0); i--) cout << s[i];  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



# 二.选做题

## 1、[天天向上](http://lx.lanqiao.cn/problem.page?gpid=T753)

A同学的学习成绩十分不稳定，于是老师对他说“只要你连续4天成绩有进步，那我就奖励给你一朵小红花。”可是这对于A同学太困难了。于是，老师对他放宽了要求“只要你有4天成绩是递增的，我就奖励你一朵小红花。”即只要对于第四天，满足并且对于成绩，那么就可以得到一朵小红花的奖励。现让你求出，A同学可以得到多少朵小红花。

### 算法思路:

求第四天的连续递增，就要求三天的连续递增，。。。一直到第一天

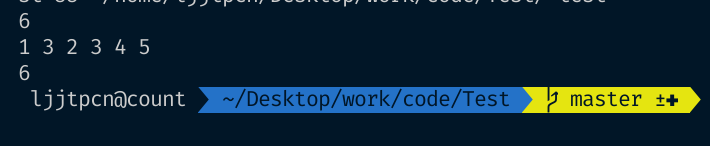
所以

### 流程图:

### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 2010;  
  
int n;  
ll f[N][N], arr[N];  
void solve(){  
 cin >> n;  
 for(int i = 1; i <= n; i++) {  
 cin >> arr[i];  
 f[i][1] = 1;  
 }   
  
 //f[i][j]表示以第i天结尾，递增j天的方案数  
 for(int j = 2; j <= 4; j ++){  
 for(int i = 1; i <= n; i ++){  
 for(int k = 1; k < i; k ++){  
 if(arr[i] > arr[k])  
 f[i][j] = f[i][j] + f[k][j - 1];  
 }  
 }  
 }  
   
 ll ans = 0;  
 for(int i = 1; i <= n; i ++){  
 ans += f[i][4];  
 }  
  
 cout << ans << endl;  
}  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 2、[欧拉函数](http://lx.lanqiao.cn/problem.page?gpid=T752)

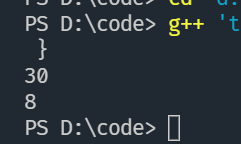
老师出了一道难题，小酱不会做，请你编个程序帮帮他，奖金一瓶酱油,从中有多少个数与互质?

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, m, arr[N], f[N];  
  
void solve(){  
 cin >> n;  
 int cnt = 0, N = n;  
 for (int i = 2; i \* i <= N; i++){  
 if(n % i == 0) arr[cnt++] = i;  
 while(n % i == 0) n /= i;  
 }  
 if(n > 1) arr[cnt++] = n;  
  
 int ans = N;  
 for(int i = 0; i < cnt; i++){  
 ans = ans - ans / arr[i];  
 }  
 cout << ans;  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 3、[计算超阶乘](http://lx.lanqiao.cn/problem.page?gpid=T751)

计算1\*(1+k)\*(1+2\*k)*(1+3*k)\*...\*(1+n\*k-k)的末尾有多少个0，最后一位非0位是多少。

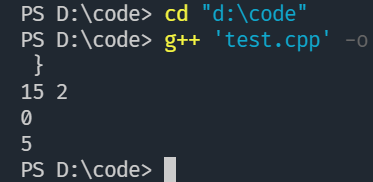
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 30010;  
int n, k, arr[N], f[N];  
  
void solve(){  
 cin >> n >> k;  
  
 long long sum = 1, cnt = 0;  
 for (int i = 1; i < n; i ++){  
 sum \*= (1 + i \* k);  
 while(sum % 10 == 0){  
 sum /= 10;  
 cnt++;  
 }  
 sum %= 1000000000;  
 }  
 cout << cnt << endl << sum % 10;  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 4、[区间最大和](http://lx.lanqiao.cn/problem.page?gpid=T2624)

给定一段长度为N的整数序列A，请从中选出一段连续的子序列（可以为0）使得这段的总和最大。

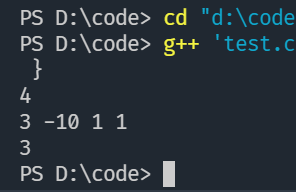
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 100010;  
typedef long long ll;  
ll n, k, arr[N], f[N];  
  
void solve(){  
 cin >> n;  
 for (int i = 1; i <= n; i++) cin >> arr[i];  
  
 ll ans = 0;  
 for(int i = 1; i <= n; i ++){  
 f[i] = max(arr[i], f[i-1] + arr[i]);  
 ans = max(ans, f[i]);  
 }  
 cout << ans << endl;  
}  
  
int main(){  
 int \_\_ = 1;  
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 5、[高精度乘法]( http://lx.lanqiao.cn/problem.page?gpid=T722)

计算机真是最傻的东西；他都不能计算大于10^65-1的a\*b，请你帮这个“最傻”的东西过关，否则它就真的只认识1和0了。

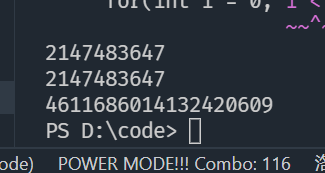
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e6 + 10;  
//高精乘高精度  
vector<int> mul(vector<int> &A, vector<int> &B) {  
 vector<int> C(A.size() + B.size(), 0); // 初始化为 0，且999\*99最多 5 位  
  
 for (int i = 0; i < A.size(); i++)  
 for (int j = 0; j < B.size(); j++)  
 C[i + j] += A[i] \* B[j];  
  
 int t = 0;  
 for (int i = 0; i < C.size(); i++) { // i = C.size() - 1时 t 一定小于 10  
 t += C[i];  
 C[i] = t % 10;  
 t /= 10;  
 }  
  
 while (C.size() > 1 && C.back() == 0) C.pop\_back(); // 必须要去前导 0，因为最高位很可能是 0  
 reverse(C.begin(), C.end());  
 return C;  
}  
  
vector<int> a, b;  
void solve(){  
 string s1, s2;  
 cin >> s1 >> s2;  
 for(int i = s1.size() - 1; i >= 0; i--) a.push\_back(s1[i] - '0');  
 for(int i = s2.size() - 1; i >= 0; i--) b.push\_back(s2[i] - '0');  
  
 vector<int> c = mul(a, b);  
 for(int i = 0; i < c.size(); i++) cout << c[i];  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 6、[字符串匹配](http://lx.lanqiao.cn/problem.page?gpid=T656)

给出一个字符串和多行文字，在这些文字中找到字符串出现的那些行。你的程序还需支持大小写敏感选项当选项打开时，表示同一个字母的大写和小写看作不同的字符；当选项关闭时，表示同一个字母的大写和小写看作相同的字符。

### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e6 + 10;  
  
string s, sub;  
int flag, n;  
  
void solve(){  
 cin >> sub >> flag >> n;  
 if(!flag) transform(sub.begin(), sub.end(), sub.begin(), ::tolower);  
 for(int i = 0; i < n; i++){  
 cin >> s;  
 string copy = s;  
 if(!flag) transform(copy.begin(), copy.end(), copy.begin(), ::tolower);  
 if(copy.find(sub) != string::npos) cout << s << endl;  
 }  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 7、[递归输出](http://lx.lanqiao.cn/problem.page?gpid=T594)

编写递归函数，将组成整数的所有数字逐个输出，每个数字后面加上一个减号“-”，例如对于整数123，该函数将输出1-2-3- 。编写主函数测试该递归函数。

### 算法思路:

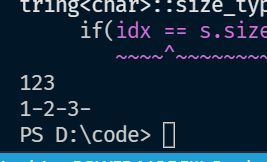
递归

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e6 + 10;  
  
string s;  
void dfs(int idx){  
 if(idx == s.size()) return;  
 cout << s[idx] << '-';  
 dfs(idx + 1);  
}  
void solve(){  
 cin >> s;  
 dfs(0);  
}  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 8、[字符串压缩](http://lx.lanqiao.cn/problem.page?gpid=T578)

编写一个程序，输入一个字符串，然后采用如下的规则对该字符串当中的每一个字符进行压缩

(1) 如果该字符是空格，则保留该字符；

(2) 如果该字符是第一次出现或第三次出现或第六次出现，则保留该字符；

(3) 否则，删除该字符。

例如，若用户输入“occurrence”，经过压缩

后，字符c的第二次出现被删除，第一和第三次出现仍保留；字符r和e的第二次出现均被删除，因此最后的结果为“ocurenc”。

### 算法思路:

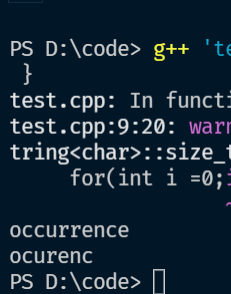
模拟题意

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e6 + 10;  
  
map<char, int> mp;  
void solve(){  
 string s;  
 getline(cin, s);  
 for(int i =0;i < s.size(); i ++){  
 if(s[i] == ' ') cout << s[i];  
 else{  
 if(mp[s[i]] == 0 || mp[s[i]] == 2 || mp[s[i]] == 5){  
 cout << s[i];  
 }  
 mp[s[i]]++;  
 }  
 }  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 9、[奥运会开幕式](http://lx.lanqiao.cn/problem.page?gpid=T570)

学校给高一（三）班分配了一个名额，去参加奥运会的开幕式。每个人都争着要去，可是名额只有一个，怎么办？班长想出了一个办法，让班上的所有同学（共有n个同学）围成一圈，按照顺时针方向进行编号。然后随便选定一个数m，并且从1号同学开始按照顺时针方向依次报数，1, 2, …, m，凡报到m的同学，都要主动退出圈子。然后不停地按顺时针方向逐一让报出m者出圈，最后剩下的那个人就是去参加开幕式的人。

要求用环形链表的方法来求解。所谓环形链表，即对于链表尾结点，其next指针又指向了链表的首结点。基本思路是先创建一个环形链表，模拟众同学围成一圈的情形。然后进入循环淘汰环节，模拟从1到m报数，每次让一位同学（结点）退出圈子。

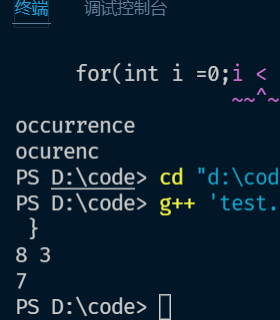
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e6 + 10;  
  
int n, m;  
deque<int> q;  
void solve(){  
 cin >> n >> m;  
 for (int i = 1; i <= n; i ++) q.push\_back(i);  
 while(q.size() > 1){  
 for (int i = 0; i < m; i ++) {  
 int x = q.front();  
 q.pop\_front();  
 q.push\_back(x);  
 }  
 q.pop\_back();  
 }  
 cout << q.front() << endl;  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 10、[双十一抢购](http://lx.lanqiao.cn/problem.page?gpid=T552)

一年一度的双十一又来了，某网购网站又开始了半价销售的活动。

小G打算在今年的双十一里尽情地购物，以享受购买的极度快感，她已经列好了她想买的物品的列表。

当然小G并不是出身富贵家庭，所以她网银里的钱只是一个有限的整数S（单位元）。

这次抢购她打算遵循这三个原则选择每一个物品

1.先买能“赚”最多的；

2.在“赚”一样多的情况下，先买最便宜的（这样买的东西就可能更多了）；

3.在前两条里都判断不了购买顺序的话，先购买在列表里靠前的。

（由于网站里还是有一部分商品并没有打五折，所以2的情况（“赚”的钱数为0）是完全可能发生的）

现在，在双十一的这一天，你要帮小G编写一个程序，来看看她应该去买她列表里的哪些物品。（总价格不要超过S哦）

要是帮她写好这个程序的话，或许你能在光棍节这一天里赢得她的芳心哦~

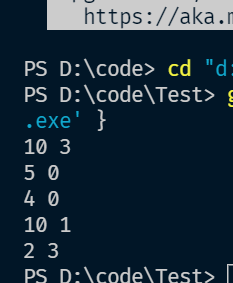
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e3 + 10;  
  
struct node{  
 double price;  
 int base, id;  
} st[N];  
bool cmp(const node& a, const node& b){  
 double p1 = a.base == 0 ? 0 : a.price / 2.0;  
 double p2 = b.base == 0 ? 0 : b.price / 2.0;  
 if(p1 != p2) return p1 > p2;  
 else if(a.price != b.price) return a.price < b.price;  
 else return a.id < b.id;  
}  
  
int n, m;  
double s;  
void solve(){  
 cin >> s >> n;  
 for (int i = 1; i <= n; i++){  
 double a;  
 int b;  
 cin >> a >> b;  
 st[i].price = a, st[i].base = b, st[i].id = i;  
 }  
 sort(st + 1, st + 1 + n, cmp);  
  
 vector<int> out;  
 for(int i = 1; i <= n; i++){  
 double t = st[i].base == 1 ? st[i].price / 2.0 : st[i].price;  
 if(s < t) continue;  
 out.push\_back(st[i].id);  
 s -= t;  
 }  
   
 sort(out.begin(), out.end());  
 if(out.size() == 0) cout << 0 << endl;  
 else for(int i = 0; i < out.size(); i ++) cout << out[i] << ' ';  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:fig:

## 11、[Monday-Saturday质因子](http://lx.lanqiao.cn/problem.page?gpid=T546)

这个问题是个简单的与数论有关的题目，看起来似乎是“求正整数的所有质因子”，但实际上并不完全是这样。

本题中需要定义以下几个概念

1． Monday-Saturday数

对于一个正整数N，如果它除以7得到的余数是1或6，则可以写成N=7k+{1,6}的形式。更形象的，我们把这样的N称作“Monday-Saturday数”，简称“MS数”。

2． Monday-Saturday因子

如果对于两个MS数a,b，若存在一个MS数x，使得ax=b，那么就称a是b的一个“Monday-Saturday因子”，简称“MS因子”。

3． Monday-Saturday质数

如果对于MS数a，满足a>1且除了1和a之外a没有其他的MS因子，那么称a是一个“Monday-Saturday质数”，简称“MS质数”。

注对于传统意义上的质数，若它是一个MS数，则它一定是一个MS质数。但反之不必成立，例如27，它是一个MS质数但不是传统意义上的质数。

4． Monday-Saturday质因子

如果对于两个MS数a,b，若满足a是b的MS因子且a是一个MS质数，那么称a是b的一个“Monday-Saturday质因子”。

例如27是216的一个MS质因子（216=27\*8）。

问题就是，给定一个MS数N，求其所有的Monday-Saturday质因子。

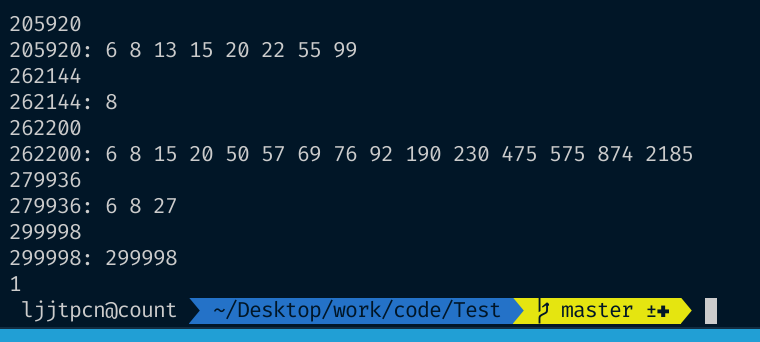
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
  
bool sign[300010] = {false};  
vector<int>v;  
bool check(int x)  
{  
 if(!sign[x])  
 return false;  
 for(int i=2;i<=x/2;i++){  
 if(x%i==0&&sign[i])  
 return false;  
 }  
 return true;  
}  
int main()  
{  
 sign[1] = true; sign[6] = true;  
 for(int i=1;i<=300000;i++)  
 if(sign[i])  
 sign[i+7] = true;  
 int n;  
 while(~scanf("%d",&n) && n!=1){  
 if(check(n))  
 cout<<n<<": "<<n<<endl;  
 else{  
 v.clear();  
 for(int i=2;i<=n/2;i++)  
 if(n%i==0)  
 v.push\_back(i);  
 cout<<n<<":";  
 for(int i=0;i<v.size();i++)  
 {  
 if(check(v[i]))  
 cout<<" "<<v[i];  
 }  
 cout<<endl;  
 }  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 12、[P0602](http://lx.lanqiao.cn/problem.page?gpid=T2627)

编写一个程序，输入一个4位的自然数，将组成该数的各位数字重新排列，形成一个最大数和一个最小数，之后用最大数减去最小数，得到一个新的自然数，把这个数打印出来。然后对于这个新的自然数，重复上述步骤，直到该自然数的值不再发生变化。例如，假设用户输入的自然数为1001，那么由它所形成的最大数为1100，最小数为11，因此新的自然数为1089。对于1089，由它形成的最大数为9810，最小数为189，因此新的自然数为9621。9621的最大数为9621，最小数为1269，结果为8352,。8352的最大数为8532，最小数为2358，结果为6174。6174的最大数为7641，最小数为1467，结果仍为6174，因此程序结束。

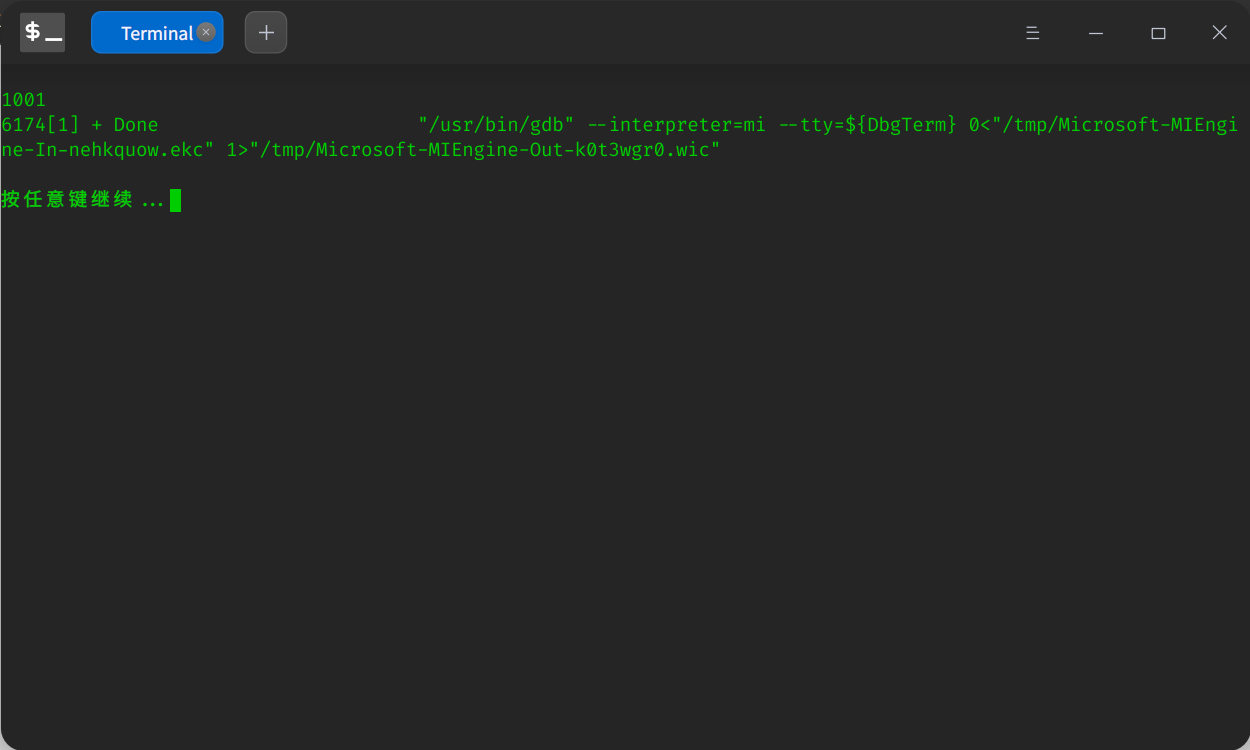
### 算法思路:

### 流程图:

### 源代码:

#include <bits/stdc++.h>  
using namespace std;  
  
const int N = 1e5+10;  
long long m,n,x,y;//全局声明  
string ss,a,b;  
  
void solve(string s){//处理4位数字符串  
 a=s;  
 sort(a.begin(),a.end());//每个位置从小到大  
 b=a;//b字符串是最小的数  
 reverse(a.begin(),a.end());//反转a，得到最大的数a  
 n=0;  
 m=0;  
  
 for (int i = 0; i < 4; i ++ ){//将4位数字符串转化为int  
 if(i==3){   
 n+=a[i]-48;  
 m+=b[i]-48;  
 }  
 else {  
 n+=(a[i]-48)\*pow(10,3-i);  
 m+=(b[i]-48)\*pow(10,3-i);  
 }  
 }  
}  
  
int main()  
{  
 cin>>ss;  
 solve(ss);  
  
 x=n-m;  
 y=0;  
 while(x!=y){//找到两次相邻相同的结果就终止  
  
 y=x;//上一次的结果  
 for (int i = 3; i >= 0; i -- ){//将int结果还原为字符串ss  
 ss[i]=48+x%10;  
 x/=10;  
 }  
 solve(ss);//重新得到得到最大数n，和最小数m  
 x=n-m;//当前结果  
 }  
 cout<<x;//输出结果  
 return 0;  
}

### 运行截图:



### 提交记录:



## 13、[矩阵乘法](http://lx.lanqiao.cn/problem.page?gpid=T530)

小明最近刚刚学习了矩阵乘法，但是他计算的速度太慢，于是他希望你能帮他写一个矩阵乘法的运算器。

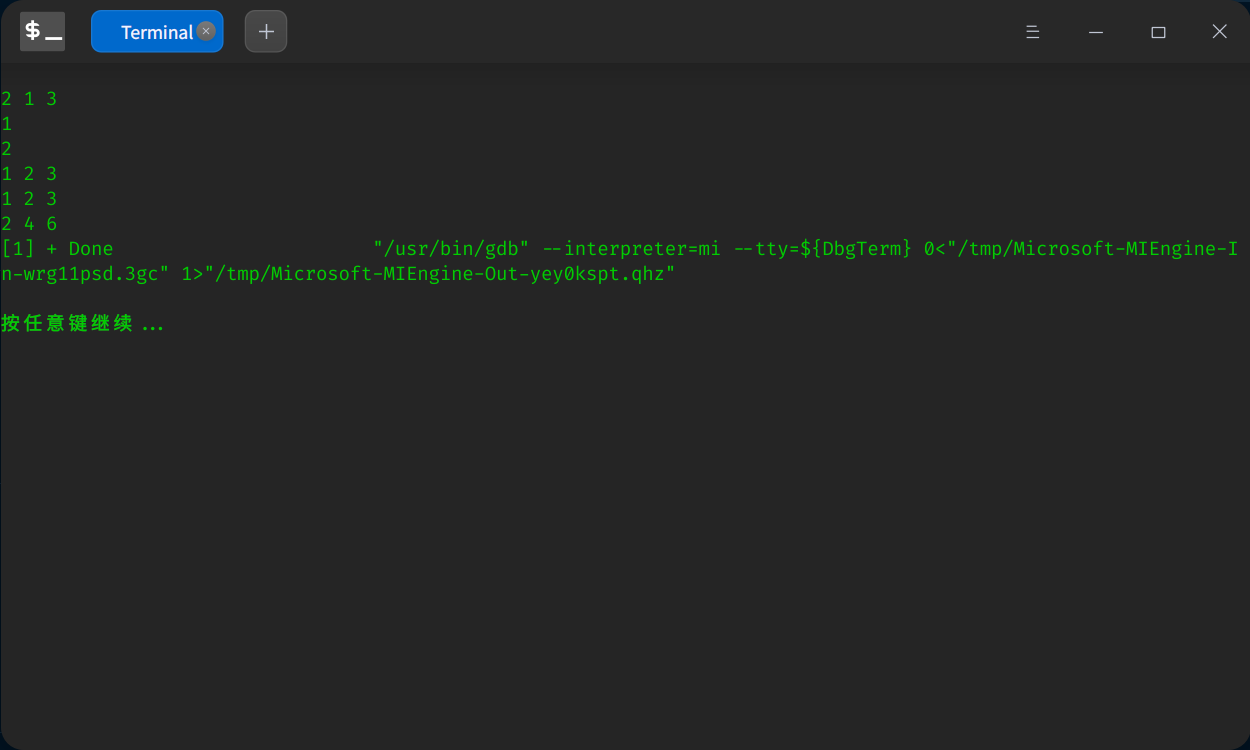
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
const int N = 1e3 + 10;  
  
int n, m, k, arr1[N][N], arr2[N][N];  
void solve(){  
 cin >> n >> m >> k;  
 for(int i = 1; i <= n; i++)  
 for(int j = 1; j <= m; j++)  
 cin >> arr1[i][j];  
 for(int i = 1; i <= m; i++)  
 for(int j = 1; j <= k; j++)  
 cin >> arr2[i][j];  
  
 int ans[N][1010];  
 for(int i = 1; i <= n; i++){  
 for(int j = 1; j <= k; j++){  
 ans[i][j] = 0;  
 for(int l = 1; l <= m; l++)   
 ans[i][j] += arr1[i][l] \* arr2[l][j];  
 cout << ans[i][j] << ' ';  
 }  
 cout << endl;  
 }  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 14、[JOE的算数](http://lx.lanqiao.cn/problem.page?gpid=T526)

有一天，JOE终于不能忍受计算a^b%c这种平凡的运算了。所以他决定要求你写一个程序，计算a^b%c。

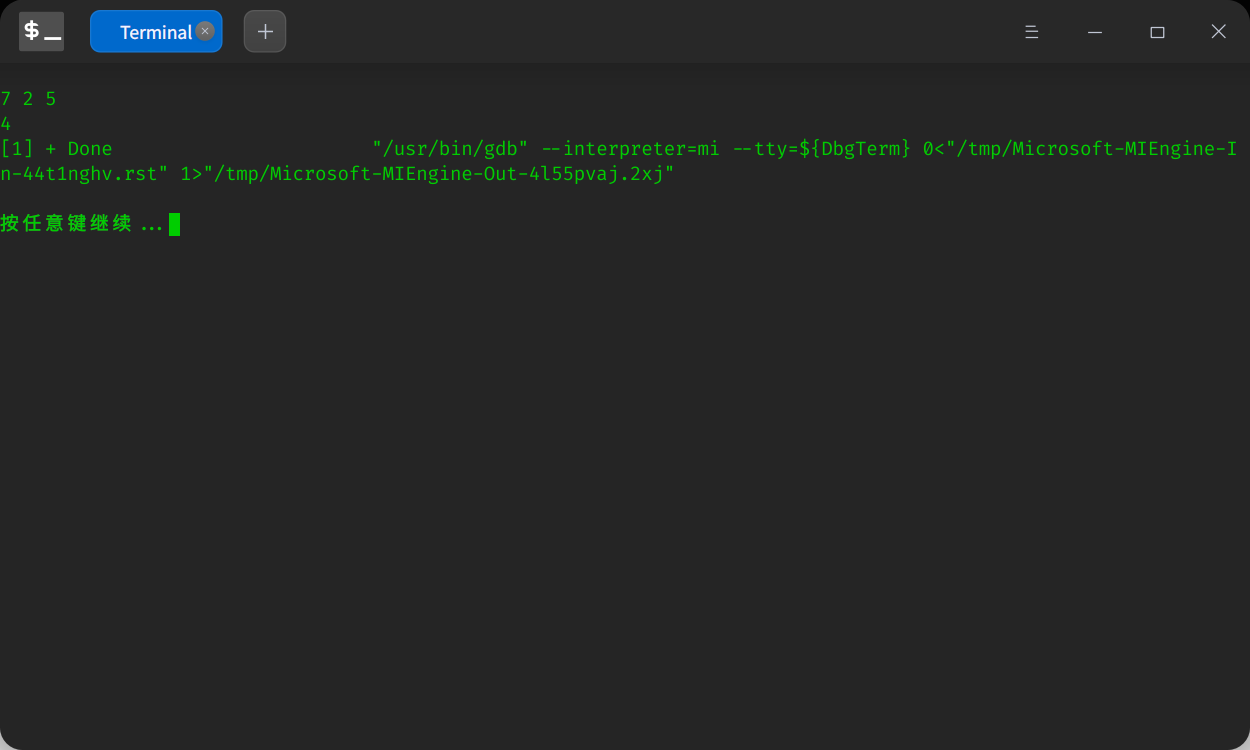
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
#define ll long long  
using namespace std;  
  
const int N = 1e3 + 10;  
  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b, int mod) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
  
ll a, b, mod;  
void solve(){  
 cin >> a >> b >> mod;  
 cout << inv(a, b, mod) << endl;  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 15、[P0102](http://lx.lanqiao.cn/problem.page?gpid=T427)

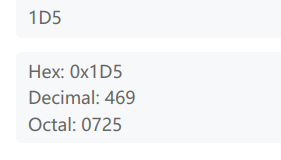
用户输入三个字符，每个字符取值范围是0-9，A-F。然后程序会把这三个字符转化为相应的十六进制整数，并分别以十六进制，十进制，八进制输出，十六进制表示成3位，八进制表示成4位，若不够前面补0。（不考虑输入不合法的情况）

### 流程图:

### 源代码:

#include<stdio.h>  
#include<math.h>  
void SixTwo(char st,int \*arr,int k){//将十六进制数转换成二进制数  
 int temp[4],num;  
 if(st<58)  
 num=st-48;  
 else  
 num=10+st-65;  
 for(int i=3;i>=0;i--){  
 temp[i]=num%2;  
 num/=2;   
 }  
 for(int i=0;i<4;i++)  
 arr[i+4\*k]=temp[i];  
}  
void TwoTen(int two[],int \*ten){//将二进制数转换成十进制数  
 for(int i=11;i>=0;i--){  
 \*ten+=pow(2,11-i)\*two[i];  
 }  
}  
void TwoSeven(int two[],int \*seven){//将二进制数转换成八进制数  
 int k=0,sum;  
 for(int i=0;i<12;i+=3){  
 sum=0;  
 for(int j=i;j<i+3;j++){  
 sum+=pow(2,i+2-j)\*two[j];  
 }  
 seven[k++]=sum;  
 }  
}  
int main(){  
 char str[3];  
 int two[12],ten=0,seven[4];  
 gets (str);  
 for(int i=0;i<3;i++){  
 SixTwo(str[i],two,i);  
 }  
 TwoTen(two,&ten);  
 TwoSeven(two,seven);  
 printf("Hex: 0x%s\n",str);  
 printf("Decimal: %d\nOctal: ",ten);  
 for(int i=0;i<4;i++)  
 printf("%d",seven[i]);  
 return 0;  
}

### 运行截图:



### 提交记录:



## 16、[三进制数位和](http://lx.lanqiao.cn/problem.page?gpid=T425)

给定L和R，你需要对于每一个6位三进制数（允许前导零），计算其每一个数位上的数字和，设其在十进制下为S。

一个三进制数被判断为合法，当且仅当S为质数，或者S属于区间[L,R]。

你的任务是给出合法三进制数的个数。

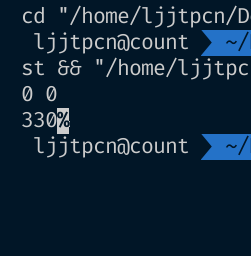
### 算法思路:

### 流程图:

### 源代码:

#include <bits/stdc++.h>  
#define ll long long  
using namespace std;  
const int N = 1e3 + 10;  
  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b, int mod)  
{  
 ll ans = 1;  
 while (b)  
 {  
 if (b & 1)  
 ans = ans \* a % mod;  
 a = a \* a % mod;  
 b >>= 1LL;  
 }  
 return ans % mod;  
}  
  
int judge(int x) //判断质数  
{  
 int i;  
 if (x < 2)  
 return 0;  
 for (i = 2; i <= sqrt(x); i++)  
 if (x % i == 0)  
 return 0;  
 return 1;  
}  
  
vector<int> a;  
int R, L, ans;  
void solve()  
{  
 cin >> L >> R;  
 int i, j, k;  
 for (i = 0; i < 6; i++)  
 a.push\_back(0);  
 int sum1;  
 for (i = 0; i <= 728; i++) //模拟6位3进制+  
 {  
 sum1 = 0;  
 for (j = 5; j >= 0; j--)  
 {  
 if (a[j] > 2)  
 {  
 a[j - 1]++;  
 a[j] = 0;  
 }  
 }  
  
 for (k = 0; k < 6; k++) //求数位和  
 sum1 += a[k];  
 if ((sum1 >= L && sum1 <= R) || (judge(sum1)) == 1) //判断满足条件  
 ans++;  
 a[5]++;  
 }  
 cout << ans;  
}  
  
int main()  
{  
 int \_\_ = 1;  
 while (\_\_--)  
 {  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 17、[9-1九宫格](http://lx.lanqiao.cn/problem.page?gpid=T409)

九宫格。输入1-9这9个数字的一种任意排序，构成3\*3二维数组。如果每行、每列以及对角线之和都相等，打印1。否则打印0。

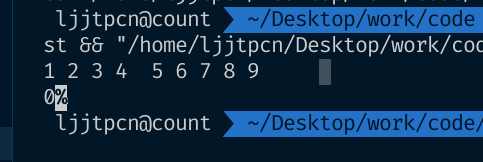
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
int main(){  
 int a[3][3],b[9],k=0;  
 for(int i=0;i<3;i++){  
 for(int j=0;j<3;j++){  
 cin>>a[i][j];  
 }  
 }  
 b[0]=a[0][0]+a[0][1]+a[0][2];  
 b[1]=a[1][0]+a[1][1]+a[1][2];  
 b[2]=a[2][0]+a[2][1]+a[2][2];  
 b[3]=a[0][0]+a[1][0]+a[2][0];  
 b[4]=a[0][1]+a[1][1]+a[2][1];  
 b[5]=a[0][2]+a[1][2]+a[2][2];  
 b[6]=a[0][0]+a[1][1]+a[2][2];  
 b[7]=a[0][2]+a[1][1]+a[2][0];  
 for(int i=0;i<7;i++){  
 if(b[i]!=b[i+1]){  
 k=1;  
 break;  
 }  
 }  
 if(k==1){  
 cout<<"0";  
 } else{  
 cout<<"1";  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 18、[身份证号码升级](http://lx.lanqiao.cn/problem.page?gpid=T325)

从1999年10月1日开始，公民身份证号码由15位数字增至18位。(18位身份证号码简介)。升级方法为

1、把15位身份证号码中的年份由2位(7,8位)改为四位。

2、最后添加一位验证码。验证码的计算方案

将前 17 位分别乘以对应系数 (7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2) 并相加，然后除以 11 取余数，0-10 分别对应 1 0 x 9 8 7 6 5 4 3 2。

请编写一个程序，用户输入15位身份证号码，程序生成18位身份证号码。假设所有要升级的身份证的四位年份都是19××年

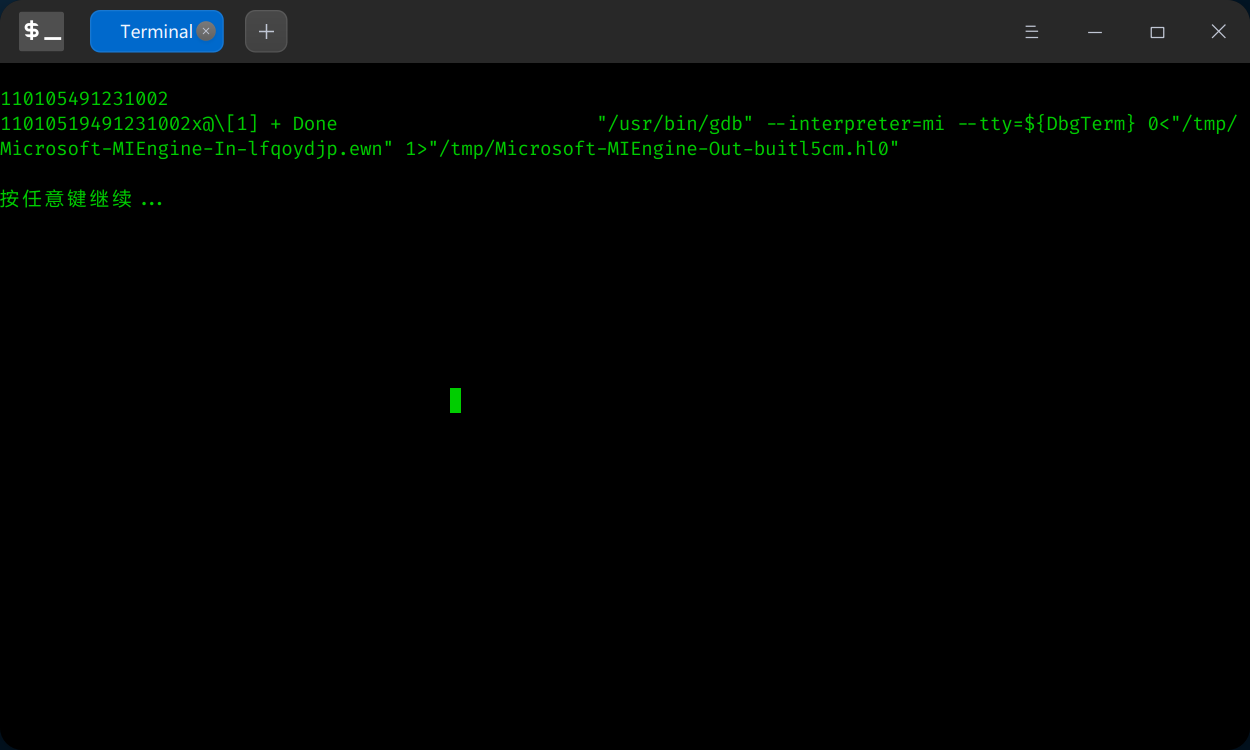
### 算法思路:

### 流程图:

### 源代码:

#include <bits/stdc++.h>  
using namespace std;  
int main()  
{  
 char k, b[18] = {0};  
 int j = 0, c[17] = {7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2}, sum = 0;  
 string a = "10x98765432";  
 cin >> b;  
 for (int i = 14; i > 5; i--)  
 {  
 b[16 - j] = b[i];  
 j++;  
 }  
 b[6] = '1';  
 b[7] = '9';  
 for (j = 0; j < 17; j++)  
 {  
 sum = sum + (b[j] - '0') \* c[j];  
 }  
 b[17] = a[sum % 11];  
 cout << b;  
   
 return 0;  
}

### 运行截图:



### 提交记录:



## 19、[快乐司机](http://lx.lanqiao.cn/problem.page?gpid=T321)

"嘟嘟嘟嘟嘟嘟

喇叭响

我是汽车小司机

我是小司机

我为祖国运输忙

运输忙"

这是儿歌“快乐的小司机”。话说现在当司机光有红心不行，还要多拉快跑。多拉不是超载，是要让所载货物价值最大，特别是在当前油价日新月异的时候。司机所拉货物为散货，如大米、面粉、沙石、泥土......

现在知道了汽车核载重量为w，可供选择的物品的数量n。每个物品的重量为gi,价值为pi。求汽车可装载的最大价值。（n<10000,w<10000,0<gi<=100,0<=pi<=100)

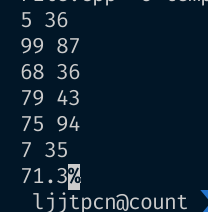
### 算法思路:

### 流程图:

### 源代码:

#include<bits/stdc++.h>  
using namespace std;  
  
struct goods{  
 int weight;  
 int value;  
 float perValue;  
};  
  
bool cmp(goods temp1, goods temp2){  
 return temp1.perValue > temp2.perValue;  
}  
  
int main(){  
 int n,w;  
 cin >> n >> w;  
 vector<goods> goodsList;  
 for(int i = 0;i < n; i++){  
 goods tempGoods;  
 cin >> tempGoods.weight >> tempGoods.value;  
 tempGoods.perValue = tempGoods.value \* 1.00 / tempGoods.weight;  
 goodsList.push\_back(tempGoods);  
 }  
 sort(goodsList.begin(), goodsList.end(), cmp);  
 int countWeight = 0;  
 float maxValue = 0;  
 for(int j = 0;j < n; j++){  
 if(goodsList[j].weight <= w - countWeight){  
 countWeight += goodsList[j].weight;  
 maxValue += goodsList[j].value;  
 }else{  
 maxValue += (w - countWeight) \* goodsList[j].perValue;  
 countWeight += w - countWeight;  
 }  
 if(countWeight == w){  
 break;  
 }  
 }  
 printf("%.1f", maxValue);  
 return 0;  
}

### 运行截图:



### 提交记录:



## 20、[数组移动](http://lx.lanqiao.cn/problem.page?gpid=T2672)

初始数组A[N]中为1,2,..,N，N个数字，现要进行M次操作，每次操作给定一个数字i，记其在数组中的位置为Bi，将A[1]..A[Bi]移到数组末尾。

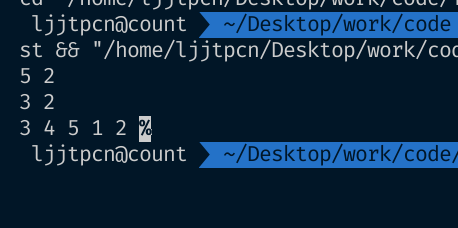
### 算法思路:

### 流程图:

### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(false); cin.tie(nullptr); cout.tie(nullptr)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 30010;  
  
int n, m, t;  
vector<int> q;  
void solve(){  
 cin >> n >> t;  
 for(int i = 1; i <= n; i ++) q.push\_back(i);  
 int k;  
 for(int i = 0; i < t; i ++) cin >> k;  
 for(int i = 0; i < n; i ++){  
 if(k >= n){  
 k = 0;  
 }  
 cout << k + 1 << ' ';  
 k++;  
 }  
}  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 21、[接水问题](http://lx.lanqiao.cn/problem.page?gpid=T2028)

学校里有一个水房，水房里一共装有m 个龙头可供同学们打开水，每个龙头每秒钟的 供水量相等，均为1。 现在有n 名同学准备接水，他们的初始接水顺序已经确定。将这些同学按接水顺序从1 到n 编号，i 号同学的接水量为wi。接水开始时，1 到m 号同学各占一个水龙头，并同时打 开水龙头接水。当其中某名同学j 完成其接水量要求wj 后，下一名排队等候接水的同学k 马上接替j 同学的位置开始接水。这个换人的过程是瞬间完成的，且没有任何水的浪费。即 j 同学第x 秒结束时完成接水，则k 同学第x+1 秒立刻开始接水。若当前接水人数n’不足m， 则只有n’个龙头供水，其它m−n’个龙头关闭。 现在给出n 名同学的接水量，按照上述接水规则，问所有同学都接完水需要多少秒。

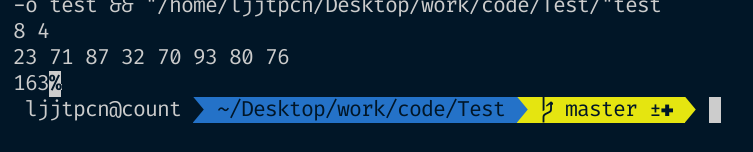
### 算法思路:

### 流程图:

### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 10010;  
  
int n, m, ans, w[N];  
priority\_queue<int, vector<int>, greater<int>> q;  
void solve(){  
 cin >> n >> m;  
 for(int i = 0; i < n; i ++) cin >> w[i];  
 for(int i = 0; i < m; i ++) q.push(w[i]), ans = max(ans, w[i]);  
  
 for(int i = m; i < n; i++){  
 int t = q.top();  
 q.pop();  
 q.push(w[i] + t);  
 ans = max(ans, t + w[i]);  
 }  
 cout << ans;  
}  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



### 提交记录:



## 22、[阴谋](http://lx.lanqiao.cn/problem.page?gpid=T2953)

第四次圣杯战争开始了。 为了收集敌人的情报，言峰绮礼命令他的使魔Assassin将自己的灵体分成n份，分别监视教堂外的长直走道。

Assassin每份灵体的能力不同。 第i份灵体可以监视到的区域是闭区间[ai,bi]。

绮礼想知道，监控范围内的区域的总长度是多少。

比如，第一份灵体的视野是[−1,1]，第二份灵体的视野是[0,2]，第三份灵体的视野是[3,4]。 那么绮礼能获得的全部视野是[−1,2]∪[3,4]，长度为4。

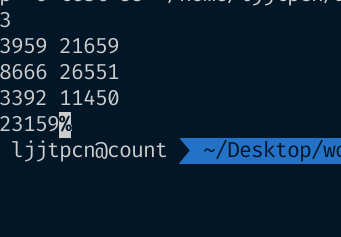
### 算法思路:

### 流程图:

### 源代码:

#define debug(a) cout << #a << " = " << a << endl;  
#define debug2(a,b) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<endl;  
#define debug3(a,b,c) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<endl;  
#define debug4(a,b,c,d) cout<<#a<<" = "<<a<<' '<<#b<<" = "<<b<<' '<<#c<<" = "<<c<<' '<<#d<<" = "<<d<<endl;  
#define debugx(a, idx) cout << #a << "[" << idx << "] = " << a[idx] << endl;  
#define debugarr(a, x) cout<<#a<<": "; for(int i = 0; i < x; i++) cout << a[i] << ' '; cout << endl;  
#define debugvec(a) cout<<#a<<": "; for(int i = 0; i < a.size(); i++) cout << a[i] << ' '; cout << endl;  
#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0)  
#define cin1(a) cin >> a;  
#define cin2(a, b) cin >> a >> b;  
#define cin3(a, b, c) cin >> a >> b >> c;  
#include<bits/stdc++.h>  
using namespace std;  
#define pb push\_back  
#define eb emplace\_back  
#define fir first  
#define sec second  
#define ll long long  
#define ull unsigned long long  
#define inf 0x3f3f3f3f  
#define INF 0x3f3f3f3f3f3f3f3f  
#define endl '\n'  
typedef pair<int,int> pii;  
const int mod = 1000000007;  
struct node{int x, y, cnt;};struct cmp{bool operator()(node a, node b){return a.cnt > b.cnt;}};  
//快速幂, 逆元 求逆元前提mod质数  
ll inv(ll a, ll b) {ll ans = 1;while(b) {if(b & 1) ans = ans \* a % mod;a = a \* a % mod;b >>= 1LL;}return ans % mod;}  
const int N = 10010;  
  
// 将所有存在交集的区间合并  
void merge(vector<pii> &segs)  
{  
 vector<pii> res;  
  
 sort(segs.begin(), segs.end());  
  
 int st = -2e9, ed = -2e9;  
 for (int i = 0; i < segs.size(); i ++){  
 pii seg = segs[i];  
 if (ed < seg.first){  
 if (st != -2e9) res.push\_back({st, ed});  
 st = seg.first, ed = seg.second;  
 }  
 else ed = max(ed, seg.second);  
 }  
 if (st != -2e9) res.push\_back({st, ed});   
  
 segs = res;  
}  
  
int n;  
vector<pii> segs;  
void solve(){  
 for(cin >> n; n --;){  
 int a, b;  
 cin >>a >> b;  
 segs.pb({a, b});  
 }  
   
 merge(segs);  
  
 ll ans = 0;  
 for(int i = 0; i < segs.size(); i ++){  
 pii t = segs[i];  
 ans += (t.sec - t.fir);  
 }  
   
 cout << ans;  
}  
int main(){  
 IOS;  
 int \_\_ = 1;   
 while(\_\_--){  
 solve();  
 }  
 return 0;  
}

### 运行截图:



提交记录:

