

# 信息奥赛一本通 New Help

# Table of Contents

1. 信息学奥赛一本通.....	2
1.1 语言及算法基础篇.....	16
1.1.1 第一部分：C++语言.....	16
1.1.2 第二部分 基础算法.....	237
1.1.3 第三部分 数据结构.....	536
1.2 算法提高篇.....	699
1.2.1 第一部分：基础算法.....	699

# 1. 信息学奥赛一本通

【信息学奥赛一本通】题解目录

OJ 网站:

## 【语言及算法基础篇】

### 第一部分：C++语言

#### 第一章：C++语言入门

Hello, World! (信息学奥赛一本通-T1001) : \_  
输出第二个整数 (信息学奥赛一本通-T1002) : \_  
对齐输出 (信息学奥赛一本通-T1003) : \_  
字符三角形 (信息学奥赛一本通-T1004) : \_  
地球人口承载力估计 (信息学奥赛一本通-T1005) : \_

#### 第二章：顺序结构程序设计

##### 第一节 运算符和表达式

A+B 问题 (信息学奥赛一本通-T1006) : [?](#)  
计算 $(a+b)*c$  的值 (信息学奥赛一本通-T1007) : \_  
计算 $(a+b)/c$  的值 (信息学奥赛一本通-T1008) : \_  
带余除法 (信息学奥赛一本通-T1009) : \_  
计算分数的浮点数值 (信息学奥赛一本通-T1010) : \_

##### 第二节 常量和变量

甲流疫情死亡率 (信息学奥赛一本通-T1011) : \_  
计算多项式的值 (信息学奥赛一本通-T1012) : \_  
温度表达转化 (信息学奥赛一本通-T1013) : \_  
与圆相关的计算 (信息学奥赛一本通-T1014) : \_  
计算并联电阻的阻值 (信息学奥赛一本通-T1015) : \_

##### 第三节 标准数据类型

整型数据类型存储空间大小（信息学奥赛一本通-T1016）：  
浮点型数据类型存储空间大小（信息学奥赛一本通-T1017）：  
其他数据类型存储空间大小（信息学奥赛一本通-T1018）：  
浮点数向零舍入（信息学奥赛一本通-T1019）：  
打印 ASCII 码（信息学奥赛一本通-T1020）：  
打印字符（信息学奥赛一本通-T1021）：  
整型与布尔型的转换（信息学奥赛一本通-T1022）：  
Hello, World! 的大小（信息学奥赛一本通-T1023）：

#### 第四节 数据输入输出

保留 3 位小数的浮点数（信息学奥赛一本通-T1024）：  
保留 12 位小数的浮点数（信息学奥赛一本通-T1025）：  
空格分隔输出（信息学奥赛一本通-T1026）：  
输出浮点数（信息学奥赛一本通-T1027）：  
字符菱形（信息学奥赛一本通-T1028）：

#### 第五节 顺序结构实例

计算浮点数相除的余（信息学奥赛一本通-T1029）：  
计算球的体积（信息学奥赛一本通-T1030）：  
反向输出一个三位数（信息学奥赛一本通-T1031）：  
大象喝水（信息学奥赛一本通-T1032）：  
计算线段长度（信息学奥赛一本通-T1033）：  
计算三角形面积（信息学奥赛一本通-T1034）：  
等差数列末项计算（信息学奥赛一本通-T1035）：  
A\*B 问题（信息学奥赛一本通-T1036）：  
计算 2 的幂（信息学奥赛一本通-T1037）：  
苹果和虫子（信息学奥赛一本通-T1038）：

### 第三章 程序的控制结构

#### 第一节 if 选择结构

判断数正负（信息学奥赛一本通-T1039）：  
输出绝对值（信息学奥赛一本通-T1040）：  
奇偶数判断（信息学奥赛一本通-T1041）：  
奇偶 ASCII 值判断（信息学奥赛一本通-T1042）：  
整数大小比较（信息学奥赛一本通-T1043）：  
判断是否为两位数（信息学奥赛一本通-T1044）：  
收集瓶盖赢大奖（信息学奥赛一本通-T1045）：  
判断一个数能否同时被 3 和 5 整除（信息学奥赛一本通-T1046）：  
判断能否被 3, 5, 7 整除（信息学奥赛一本通-T1047）：  
有一门课不及格的学生（信息学奥赛一本通-T1048）：

## 第二节 switch 语句

晶晶赴约会（信息学奥赛一本通-T1049）：  
 骑车与走路（信息学奥赛一本通-T1050）：  
 分段函数（信息学奥赛一本通-T1051）：  
 计算邮资（信息学奥赛一本通-T1052）：  
 最大数输出（信息学奥赛一本通-T1053）：  
 三角形判断（信息学奥赛一本通-T1054）：  
 判断闰年（信息学奥赛一本通-T1055）：  
 点和正方形的关系（信息学奥赛一本通-T1056）：  
 简单计算器（信息学奥赛一本通-T1057）：  
 求一元二次方程（信息学奥赛一本通-T1058）：

## 第四章 循环结构的程序设计

### 第一节 for 语句

求平均年龄（信息学奥赛一本通-T1059）：  
 均值（信息学奥赛一本通-T1060）：  
 求整数的和与均值（信息学奥赛一本通-T1061）：  
 最高的分数（信息学奥赛一本通-T1062）：  
 最大跨度值（信息学奥赛一本通-T1063）：  
 奥运奖牌计数（信息学奥赛一本通-T1064）：  
 奇数求和（信息学奥赛一本通-T1065）：  
 满足条件的数累加（信息学奥赛一本通-T1066）：  
 整数的个数（信息学奥赛一本通-T1067）：  
 与指定数字相同的数的个数（信息学奥赛一本通-T1068）：  
 乘方计算（信息学奥赛一本通-T1069）：  
 人口增长（信息学奥赛一本通-T1070）：  
 菲波那契数（信息学奥赛一本通-T1071）：  
 鸡尾酒疗法（信息学奥赛一本通-T1072）：  
 救援（信息学奥赛一本通-T1073）：  
 津津的储蓄计划（信息学奥赛一本通-T1074）：  
 药房管理（信息学奥赛一本通-T1075）：  
 正常血压（信息学奥赛一本通-T1075）：  
 统计满足条件的 4 位数（信息学奥赛一本通-T1077）：  
 求分数序列和（信息学奥赛一本通-T1078）：  
 计算分数加减表达式的值（信息学奥赛一本通-T1079）：  
 余数相同问题（信息学奥赛一本通-T1080）：  
 分苹果（信息学奥赛一本通-T1081）：  
 求小数的某一位（信息学奥赛一本通-T1082）：  
 计算星期几（信息学奥赛一本通-T1083）：  
 幂的末尾（信息学奥赛一本通-T1084）：

## 第二节 while 与 do-while 语句

球弹跳高度的计算（信息学奥赛一本通-T1085）：  
 角谷猜想（信息学奥赛一本通-T1086）：  
 级数求和（信息学奥赛一本通-T1087）：  
 分离整数的各个数（信息学奥赛一本通-T1088）：  
 数字反转（信息学奥赛一本通-T1089）：  
 含 k 个 3 的数（信息学奥赛一本通-T1090）：

## 第三节 循环嵌套

求阶乘的和（信息学奥赛一本通-T1091）：  
 求出 e 的值（信息学奥赛一本通-T1092）：  
 计算多项式的值（信息学奥赛一本通-T1093）：  
 与 7 无关的数（信息学奥赛一本通-T1094）：  
 数 1 的个数（信息学奥赛一本通-T1095）：  
 数字统计（信息学奥赛一本通-T1096）：  
 画矩形（信息学奥赛一本通-T1097）：  
 质因数分解（信息学奥赛一本通-T1098）：  
 第 n 小的质数（信息学奥赛一本通-T1099）：  
 金币（信息学奥赛一本通-T1110）：  
 不定方程求解（信息学奥赛一本通-T1101）：

# 第五章 数组

## 第一节 一维数组

与指定数字相同的数的个数（信息学奥赛一本通-T1102）：  
 陶陶摘苹果（信息学奥赛一本通-T1103）：  
 计算书费（信息学奥赛一本通-T1104）：  
 数组逆序重存放（信息学奥赛一本通-T1105）：  
 年龄与疾病（信息学奥赛一本通-T1106）：  
 校门外的树（信息学奥赛一本通-T1107）：  
 向量点积计算（信息学奥赛一本通-T1108）：  
 开关灯（信息学奥赛一本通-T1109）：  
 查找特定的值（信息学奥赛一本通-T1110）：  
 不高兴的津津（信息学奥赛一本通-T1111）：  
 最大值和最小值的差（信息学奥赛一本通-T1112）：  
 不与最大数相同的数字之和（信息学奥赛一本通-T1113）：  
 白细胞计数（信息学奥赛一本通-T1114）：  
 直方图（信息学奥赛一本通-T1115）：  
 最长平台（信息学奥赛一本通-T1116）：

整数去重（信息学奥赛一本通-T1117）：  
铺地毯（信息学奥赛一本通-T1118）：

## 第二节 二维数组

矩阵交换行（信息学奥赛一本通-T1119）：  
同行列对角线的格（信息学奥赛一本通-T1120）：  
计算矩阵边缘元素之和（信息学奥赛一本通-T1121）：  
计算鞍点（信息学奥赛一本通-T1122）：  
图像相似度（信息学奥赛一本通-T1123）：  
矩阵加法（信息学奥赛一本通-T1124）：  
矩阵乘法（信息学奥赛一本通-T1125）：  
矩阵转置（信息学奥赛一本通-T1126）：  
图像旋转（信息学奥赛一本通-T1127）：  
图像模糊处理（信息学奥赛一本通-T1128）：

## 第三节 字符类型和字符数组

统计数字字符个数（信息学奥赛一本通-T1129）：  
找第一个只出现一次的字符（信息学奥赛一本通-T1130）：  
基因相关性（信息学奥赛一本通-T1131）：  
石头剪子布（信息学奥赛一本通-T1132）：  
输出亲朋字符串（信息学奥赛一本通-T1133）：  
合法 C 标识符（信息学奥赛一本通-T1134）：  
配对碱基链（信息学奥赛一本通-T1135）：  
密码翻译（信息学奥赛一本通-T1136）：  
加密的病历单（信息学奥赛一本通-T1137）：  
将字符串中的小写字母转换成大写字母（信息学奥赛一本通-T1138）：  
整理药名（信息学奥赛一本通-T1139）：  
验证子串（信息学奥赛一本通-T1140）：  
删除单词后缀（信息学奥赛一本通-T1141）：  
单词的长度（信息学奥赛一本通-T1142）：  
最长最短单词（信息学奥赛一本通-T1143）：  
单词翻转（信息学奥赛一本通-T1144）：  
字符串 p 型编码（信息学奥赛一本通-T1145）：  
判断字符串是否为回文（信息学奥赛一本通-T1146）：  
最高分数的学生姓名（信息学奥赛一本通-T1147）：  
连续出现的字符（信息学奥赛一本通-T1148）：  
最长单词（信息学奥赛一本通-T1149）：

## 第六章 函数

## 第一节 函数

求正整数 2 和  $n$  之间的完全数（信息学奥赛一本通-T1150）：  
素数个数（信息学奥赛一本通-T1151）：  
最大数  $\max(x, y, z)$ （信息学奥赛一本通-T1152）：  
绝对素数（信息学奥赛一本通-T1153）：  
亲和数（信息学奥赛一本通-T1154）：  
回文三位数（信息学奥赛一本通-T1155）：  
求  $\pi$  的值（信息学奥赛一本通-T1156）：  
哥德巴赫猜想（信息学奥赛一本通-T1157）：  
简单算术表达式求值（信息学奥赛一本通-T1397）：  
短信计费（信息学奥赛一本通-T1398）：  
甲流病人初筛（信息学奥赛一本通-T1399）：  
统计单词数（信息学奥赛一本通-T1400）：  
机器翻译（信息学奥赛一本通-T1401）：  
Vigenère 密码（信息学奥赛一本通-T1402）：  
素数对（信息学奥赛一本通-T1403）：  
我家的门牌号（信息学奥赛一本通-T1404）：  
质数的和与积（信息学奥赛一本通-T1405）：  
单词替换（信息学奥赛一本通-T1406）：  
笨小猴（信息学奥赛一本通-T1407）：  
素数回文数的个数（信息学奥赛一本通-T1408）：  
判决素数个数（信息学奥赛一本通-T1409）：  
最大质因子序列（信息学奥赛一本通-T1410）：  
区间内的真素数（信息学奥赛一本通-T1411）：  
二进制分类（信息学奥赛一本通-T1412）：  
确定进制（信息学奥赛一本通-T1413）：

## 第二节 递归算法

求  $1+2+3+\dots$ （信息学奥赛一本通-T1158）：  
斐波那契数列（信息学奥赛一本通-T1159）：  
倒序数（信息学奥赛一本通-T1160）：  
转进制（信息学奥赛一本通-T1161）：  
字符串逆序（信息学奥赛一本通-T1162）：  
阿克曼(Ackmann)函数（信息学奥赛一本通-T1163）：  
digit 函数（信息学奥赛一本通-T1164）：  
Hermite 多项式（信息学奥赛一本通-T1165）：  
求  $f(x, n)$ （信息学奥赛一本通-T1166）：  
再求  $f(x, n)$ （信息学奥赛一本通-T1167）：

# 第二部分 基础算法



## 第一章 高精度计算

高精度乘法（信息学奥赛一本通-T1307）：  
高精除（信息学奥赛一本通-T1308）：  
回文数（信息学奥赛一本通-T1309）：  
大整数加法（信息学奥赛一本通-T1168）：  
大整数减法（信息学奥赛一本通-T1169）：  
计算 2 的 N 次方（信息学奥赛一本通-T1170）：  
大整数的因子（信息学奥赛一本通-T1171）：  
求 10000 以内 n 的阶乘（信息学奥赛一本通-T1172）：  
阶乘和（信息学奥赛一本通-T1173）：  
大整数乘法（信息学奥赛一本通-T1174）：  
除以 13（信息学奥赛一本通-T1175）：

### 第二章 数据排序

车厢重组（信息学奥赛一本通-T1310）：  
求逆序对（信息学奥赛一本通-T1311）：  
谁考了第 k 名（信息学奥赛一本通-T1176）：  
奇数单增序列（信息学奥赛一本通-T1177）：  
成绩排序（信息学奥赛一本通-T1178）：  
奖学金（信息学奥赛一本通-T1179）：  
分数线划定（信息学奥赛一本通-T1180）：  
整数奇偶排序（信息学奥赛一本通-T1181）：  
合影效果（信息学奥赛一本通-T1182）：  
病人排队（信息学奥赛一本通-T1183）：  
明明的随机数（信息学奥赛一本通-T1184）：  
单词排序（信息学奥赛一本通-T1185）：  
出现次数超过一半的数（信息学奥赛一本通-T1186）：  
统计字符数（信息学奥赛一本通-T1187）：

## 第三章 递推算法

昆虫繁殖（信息学奥赛一本通-T1312）：  
位数问题（信息学奥赛一本通-T1313）：  
过河卒（信息学奥赛一本通-T1314）：  
菲波那契数列（信息学奥赛一本通-T1188）：  
Pell 数列（信息学奥赛一本通-T1189）：  
上台阶（信息学奥赛一本通-T1190）：  
流感传染（信息学奥赛一本通-T1191）：  
放苹果（信息学奥赛一本通-T1192）：  
吃糖果（信息学奥赛一本通-T1193）：

移动路线（信息学奥赛一本通-T1194）：  
 判断整除（信息学奥赛一本通-T1195）：  
 踩方格（信息学奥赛一本通-T1196）：  
 山区建小学（信息学奥赛一本通-T1197）：  
 第四章 递归算法  
 集合的划分（信息学奥赛一本通-T1315）：  
 数的计数（信息学奥赛一本通-T1316）：  
 逆波兰表达式（信息学奥赛一本通-T1198）：  
 全排列（信息学奥赛一本通-T1199）：  
 分解因数（信息学奥赛一本通-T1200）：  
 菲波那契数列（信息学奥赛一本通-T1201）：  
 Pell 数列（信息学奥赛一本通-T1202）：  
 扩号匹配问题（信息学奥赛一本通-T1203）：  
 爬楼梯（信息学奥赛一本通-T1204）：  
 汉诺塔问题（信息学奥赛一本通-T1205）：  
 放苹果（信息学奥赛一本通-T1206）：  
 求最大公约数问题（信息学奥赛一本通-T1207）：  
 2 的幂次方表示（信息学奥赛一本通-T1208）：  
 分数求和（信息学奥赛一本通-T1209）：  
 因子分解（信息学奥赛一本通-T1210）：  
 判断元素是否存在（信息学奥赛一本通-T1211）：  
 第五章 搜索与回溯算法（DFS）  
 组合的输出（信息学奥赛一本通-T1317）：  
 自然数的拆分（信息学奥赛一本通-T1318）：  
 LETTERS（信息学奥赛一本通-T1212）：  
 八皇后问题（信息学奥赛一本通-T1213）：  
 八皇后（信息学奥赛一本通-T1214）：  
 迷宫（信息学奥赛一本通-T1215）：  
 红与黑（信息学奥赛一本通-T1216）：  
 棋盘问题（信息学奥赛一本通-T1217）：  
 取石子游戏（信息学奥赛一本通-T1218）：  
 马走日（信息学奥赛一本通-T1219）：  
 单词接龙（信息学奥赛一本通-T1220）：  
 分成互质组（信息学奥赛一本通-T1221）：  
 放苹果（信息学奥赛一本通-T1222）：

## 第六章 贪心算法

排队接水（信息学奥赛一本通-T1319）：  
 均分纸牌（信息学奥赛一本通-T1320）：  
 删数问题（信息学奥赛一本通-T1321）：  
 拦截导弹问题（信息学奥赛一本通-T1322）：  
 活动选择（信息学奥赛一本通-T1323）：

整数区间（信息学奥赛一本通-T1324）：  
An Easy Problem（信息学奥赛一本通-T1223）：  
最大子矩阵（信息学奥赛一本通-T1224）：  
金银岛（信息学奥赛一本通-T1225）：  
装箱问题（信息学奥赛一本通-T1226）：  
Ride to Office（信息学奥赛一本通-T1227）：  
书架（信息学奥赛一本通-T1228）：  
电池的寿命（信息学奥赛一本通-T1229）：  
寻找平面上的极大点（信息学奥赛一本通-T1230）：  
最小新整数（信息学奥赛一本通-T1231）：  
Crossing River（信息学奥赛一本通-T1232）：  
接水问题（信息学奥赛一本通-T1233）：

## 第七章 分治算法

循环比赛日程表（信息学奥赛一本通-T1325）：  
取余运算（信息学奥赛一本通-T1326）：  
黑白棋子的移动（信息学奥赛一本通-T1327）：  
光荣的梦想（信息学奥赛一本通-T1328）：  
2011（信息学奥赛一本通-T1234）：  
输出前 k 大的数（信息学奥赛一本通-T1235）：  
区间合并（信息学奥赛一本通-T1236）：  
求排列的逆序数（信息学奥赛一本通-T1237）：  
一元三次方程求解（信息学奥赛一本通-T1238）：  
统计数字（信息学奥赛一本通-T1239）：  
查找最接近的元素（信息学奥赛一本通-T1240）：  
二分法求函数的零点（信息学奥赛一本通-T1241）：  
网线主管（信息学奥赛一本通-T1242）：  
月度开销（信息学奥赛一本通-T1243）：  
和为给定数（信息学奥赛一本通-T1244）：  
不重复地输出数（信息学奥赛一本通-T1245）：  
膨胀的木棍（信息学奥赛一本通-T1246）：  
河中跳房子（信息学奥赛一本通-T1247）：

## 第八章 广度优先搜索（BFS）

细胞（信息学奥赛一本通-T1329）：  
最少步数（信息学奥赛一本通-T1330）：  
Dungeon Master（信息学奥赛一本通-T1248）：  
Lake Counting（信息学奥赛一本通-T1249）：  
The Castle（信息学奥赛一本通-T1250）：

仙岛求药（信息学奥赛一本通-T1251）：  
走迷宫（信息学奥赛一本通-T1252）：  
抓住那头牛（信息学奥赛一本通-T1253）：  
走出迷宫（信息学奥赛一本通-T1254）：  
迷宫问题（信息学奥赛一本通-T1255）：  
献给阿尔吉侬的花束（信息学奥赛一本通-T1256）：  
Knight Moves（信息学奥赛一本通-T1257）：

## 第九章 动态规划

### 第一节 动态规划的基本模型

T1258 数字金字塔  
T1259 求最长不下降序列  
T1260 拦截导弹  
T1261 城市交通路网  
T1262 挖地雷  
T1263 友好城市  
T1264 合唱队形  
T1265 最长公共子序列  
T1266 机器分配  
T1281 最长上升子序列  
T1282 最大子矩阵  
T1283 登山  
T1284 摘花生  
T1285 最大上升子序列和  
T1286 怪盗基德的滑翔翼  
T1287 最低通行费  
T1288 三角形最佳路径问题  
T1289 拦截导弹

### 第二节 背包问题

T1267 01 背包问题  
T1268 完全背包问题  
T1269 庆功会  
T1270 混合背包  
T1271 潜水员  
T1272 分组背包  
T1273 货币系统  
T1290 采药  
T1291 数字组合  
T1292 宠物小精灵之收服  
T1293 买书

T1294 Charm Bracelet

T1295 装箱问题

T1296 开餐馆

### 第三节 动态规划经典问题

T1274 合并石子

T1275 乘积最大

T1276 编辑距离

T1277 方格取数

T1278 复制书稿

T1279 橱窗布置

T1280 滑雪

T1297 公共子序列

T1298 计算字符串距离

T1299 糖果

T1300 鸡蛋的硬度

T1301 大盗阿福

T1302 股票买卖

T1303 鸣人的影分身

T1304 数的划分

T1305 Maximum sum

T1306 最长公共子上升序列

## 第三部分 数据结构

### 第一章 栈

T1331 后缀表达式的值

T1353 表达式括号匹配

T1354 括弧匹配检验

T1355 字符串匹配问题

T1356 计算

T1357 车厢调度

T1358 中缀表达式值

### 第二章 队列

T1332 周末舞会

T1333 Blah 数集

- T1334 围圈报数
- T1335 连通块
- T1359 围成面积
- T1360 奇怪的电梯
- T1361 产生数
- T1362 家庭问题

## 第三章 树与堆

### 第一节 树与二叉树

- T1336 找树根和孩子
- T1337 单词查找树
- T1338 医院设置
- T1339 求后序遍历
- T1340 扩展二叉树
- T1363 小球
- T1364 二叉树遍历
- T1365 FBI 树
- T1366 二叉树输出
- T1367 查找二叉树
- T1368 对称二叉树

### 第二节 堆及其应用

- T1369 合并果子
- T1370 最小函数值
- T1371 看病
- T1372 小明的账单
- T1373 鱼塘钓鱼

## 第四章 图论算法

### 第一节 图的遍历

- T1341 一笔画问题
- T1374 铲雪车
- T1375 骑马修栅栏

### 第二节 最短路径算法

- T1342 最短路径问题
- T1343 牛的旅行
- T1344 最小花费
- T1345 香甜的黄油
- T1376 信使
- T1377 最优乘车
- T1378 最短路径
- T1379 热浪
- T1380 分糖果
- T1381 城市路
- T1382 最短路

### 第三节 图的连通性问题

- T1383 刻录光盘
- T1384 珍珠

### 第四节 并查集

- T1346 亲戚
- T1347 格子游戏
- T1385 团伙
- T1386 打击犯罪
- T1387 搭配购买
- T1388 家谱
- T1389 亲戚
- T1390 食物链

### 第五节 最小生成树

- T1348 城市公交网建设问题
- T1349 最优布线问题
- T1350 最短网络
- T1351 家谱树
- T1391 局域网
- T1392 繁忙的都市
- T1393 联络员
- T1394 连接格点

### 第六节 拓扑排序与关键路径

- T1352 奖金
- T1395 烦人的幻灯片
- T1396 病毒

# 【算法提高篇】

## 第一部分：基础算法

### 第一章：贪心算法

活动安排（信息学奥赛一本通-T1422）：  
种树（信息学奥赛一本通-T1423）：  
喷水装置（信息学奥赛一本通-T1424）：  
加工生产调度（信息学奥赛一本通-T1425）：  
智力大冲浪（信息学奥赛一本通-T1426）：  
数列极差（信息学奥赛一本通-T1427）：  
数列分段（信息学奥赛一本通-T1428）：  
线段（信息学奥赛一本通-T1429）：  
家庭作业（信息学奥赛一本通-T1430）：  
钓鱼（信息学奥赛一本通-T1431）：  
糖果传递（信息学奥赛一本通-T1432）：

### 第二章：二分与三分

愤怒的牛（信息学奥赛一本通-T1433）：  
Best Cow Fences（信息学奥赛一本通-T1434）：  
曲线（信息学奥赛一本通-T1435）：  
数列分段 II（信息学奥赛一本通-T1436）：  
扩散（信息学奥赛一本通-T1437）：  
灯泡（信息学奥赛一本通-T1438）：  
传送带（信息学奥赛一本通-T1439）：

### 第三章：深搜的剪枝技巧

数的划分（信息学奥赛一本通-T1440）：  
生日蛋糕（信息学奥赛一本通-T1441）：  
小木棍（信息学奥赛一本通-T1442）：  
Addition Chains（信息学奥赛一本通-T1443）：  
埃及分数（信息学奥赛一本通-T1444）：  
平板涂色（信息学奥赛一本通-T1445）：



素数方阵（信息学奥赛一本通-T1446）：

靶形数独（信息学奥赛一本通-T1447）：

## 1.1 语言及算法基础篇

### 1.1.1 第一部分：C++语言

#### 1.1.1.1 第一章：C++语言入门

##### 1.1.1.1.1 Hello,World!（信息学奥赛一本通-T1001）

## Hello,World!（信息学奥赛一本通-T1001）

原创 Alex\_McAvoy 发布于 2018-01-30 19:05:15 阅读数 12850 收藏  
展开

#### 【题目描述】

编写一个能够输出“Hello,World!”的程序，这个程序常常作为一个初学者接触一门新的编程语言所写的第一个程序，也经常用来测试开发、编译环境是否能够正常工作。

提示：“Hello,World!”中间没空格。

#### 【输入】

无

#### 【输出】

Hello,World!

#### 【输入样例】

（无）

#### 【输出样例】

Hello,World!

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Hello,World!"<<endl;
    return 0;
}
```

## 1.1.1.1.2 输出第二个整数（信息学奥赛一本通-T1002）

# 输出第二个整数（信息学奥赛一本通-T1002）

原创 Alex\_McAvoy 发布于 2018-01-30 19:05:28 阅读数 7032 收藏  
展开

**【题目描述】**

输入三个整数，整数之间由一个空格分隔，整数是 32 位有符号整数。把第二个输入的整数输出。

**【输入】**

只有一行，共三个整数，整数之间由一个空格分隔。整数是 32 位有符号整数。

**【输出】**

只有一行，一个整数，即输入的第二个整数。

**【输入样例】**

123 456 789

**【输出样例】**

456

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
```

```

    cout<<b<<endl;
    return 0;
}

```

### 1.1.1.1.3 对齐输出（信息学奥赛一本通-T1003）

## 对齐输出（信息学奥赛一本通-T1003）

原创 Alex\_McAvoy 发布于 2018-01-30 19:05:39 阅读数 8701 收藏  
展开

#### 【题目描述】

读入三个整数，按每个整数占 8 个字符的宽度，右对齐输出它们，按照格式要求依次输出三个整数，之间以一个空格分开。

#### 【输入】

只有一行，按照格式要求依次输出三个整数，之间以一个空格分开。

#### 【输出】

只有一行，一个整数，即输入的第二个整数。

#### 【输入样例】

123456789 0 -1

#### 【输出样例】

123456789      0      -1

#### 【源程序】

```

#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    printf("%8d %8d %8d",a,b,c);
    return 0;
}

```

#### 1.1.1.1.4 字符三角形（信息学奥赛一本通-T1004）

## 字符三角形（信息学奥赛一本通-T1004）

原创 Alex\_McAvoy 发布于 2018-01-30 19:05:47 阅读数 7304 收藏  
展开

### 【题目描述】

给定一个字符，用它构造一个底边长 5 个字符，高 3 个字符的等腰字符三角形。

### 【输入】

只有一行，包含一个字符。

### 【输出】

该字符构成的等腰三角形，底边长 5 个字符，高 3 个字符。

### 【输入样例】

\*

### 【输出样例】

\*  
\*\*\*  
\*\*\*\*\*

### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    char ch;
    cin>>ch;
    cout<<" "<<" "<<ch<<endl;
    cout<<" "<<ch<<ch<<ch<<endl;
    cout<<ch<<ch<<ch<<ch<<ch<<endl;
    return 0;
}
```

### 1.1.1.1.5 地球人口承载力估计（信息学奥赛一本通-T1005）

## 地球人口承载力估计（信息学奥赛一本通-T1005）

原创 Alex\_McAvoy 发布于 2018-01-30 19:06:07 阅读数 9360 收藏  
展开

#### 【题目描述】

假设地球上的新生资源按恒定速度增长。照此测算，地球上现有资源加上新生资源可供  $x$  亿人生活  $a$  年，或供  $y$  亿人生活  $b$  年。

为了能够实现可持续发展，避免资源枯竭，地球最多能够养活多少亿人？

#### 【输入】

输入只有一行，包括四个正整数  $x$ ， $a$ ， $y$ ， $b$ ，两个整数之间用单个空格隔开。 $x > y$ ， $a < b$ ， $ax < by$ ，各整数均不大于 10000。

#### 【输出】

一个实数  $z$ ，表示地球最多养活  $z$  亿人，舍入到小数点后两位。

#### 【输入样例】

110 90 90 210

#### 【输出样例】

75.00

#### 【源程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int x,y,a,b;
    double z;
    cin>>x>>a>>y>>b;
    z=(y*b-x*a)/(b-a);//计算最多人数
    cout<<setiosflags(ios::fixed)<<setprecision(2);//保留两位小数
    cout<<z<<endl;
```

```
    return 0;  
}
```

## 1.1.1.2 第二章：顺序结构程序设计

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第一部分：C++语言](#) >  
第二章：顺序结构程序设计

### 1.1.1.2.1 第一节 运算符和表达式

#### 1.1.1.2.1.1 A+B 问题（信息学奥赛一本通-T1006）

## A+B 问题（信息学奥赛一本通-T1006）

原创 Alex\_McAvoy 发布于 2018-01-30 19:06:47 阅读数 5476 收藏  
展开

### 【题目描述】

求两个整数的和。

### 【输入】

只有一行，两个用空格隔开的整数。

### 【输出】

两个整数的和。

### 【输入样例】

1 2

### 【输出样例】

3

### 【源程序】

```
#include<iostream>  
using namespace std;  
int main()  
{  
    int a,b;  
    int sum;  
    cin>>a>>b;
```

```

sum=a+b;
cout<<sum<<endl;
return 0;
}

```

#### 1.1.1.2.1.2 计算 $(a+b)*c$ 的值（信息学奥赛一本通-T1007）

## 计算 $(a+b)*c$ 的值（信息学奥赛一本通-T1007）

原创 Alex\_McAvoy 发布于 2018-01-30 19:06:59 阅读数 3357 收藏  
展开

### 【题目描述】

给定 3 个整数  $a$ 、 $b$ 、 $c$ ，计算表达式 $(a+b)*c$  的值。

### 【输入】

只有一行，包括三个整数  $a$ 、 $b$ 、 $c$ ，数与数之间以一个空格分开。（ $-10,000 < a, b, c < 10,000$ ）

### 【输出】

只有一行，即表达式的值。

### 【输入样例】

2 3 5

### 【输出样例】

25

### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    cout<<(a+b)*c<<endl;
    return 0;
}

```

#### 1.1.1.2.1.3 计算 $(a+b)/c$ 的值（信息学奥赛一本通-T1008）

## 计算 $(a+b)/c$ 的值（信息学奥赛一本通-T1008）

原创 Alex\_McAvoy 发布于 2018-01-30 19:07:06 阅读数 2572 收藏  
展开

### 【题目描述】

给定 3 个整数  $a$ 、 $b$ 、 $c$ ，计算表达式 $(a+b)/c$  的的值。

### 【输入】

只有一行，包括三个整数  $a$ 、 $b$ 、 $c$ ，数与数之间以一个空格分开。（ $-10,000 < a, b, c < 10,000$ ）

### 【输出】

只有一行，即表达式的值。

### 【输入样例】

1 1 3

### 【输出样例】

0

### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    cout<<(a+b)/c<<endl;
    return 0;
}
```

#### 1.1.1.2.1.4 带余除法（信息学奥赛一本通-T1009）

## 带余除法（信息学奥赛一本通-T1009）

原创 Alex\_McAvoy 发布于 2018-01-30 19:07:13 阅读数 4536 收藏  
展开



**【题目描述】**

给定被除数和除数，求整数商及余数。此题中请使用默认的整除和取余运算，无需对结果进行任何特殊处理。

**【输入】**

一行，包含两个整数，依次为被除数和除数（除数非零），中间用一个空格隔开。

**【输出】**

一行，包含两个整数，依次为整数商和余数，中间用一个空格隔开。

**【输入样例】**

10 3

**【输出样例】**

3 1

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    cin>>a>>b;
    cout<<a/b<<" "<<a%b<<endl;
    return 0;
}
```

点赞 1

### 1.1.1.2.1.5 计算分数的浮点数值（信息学奥赛一本通-T1010）

## 计算分数的浮点数值（信息学奥赛一本通-T1010）

原创 Alex\_McAvoy 发布于 2018-01-30 19:07:20 阅读数 5508 收藏  
展开

**【题目描述】**

两个整数  $a$  和  $b$  分别作为分子和分母，既分数  $a/b$ ，求它的浮点数值(双精度浮点数，保留小数点后 9 位)。

**【输入】**

输入仅一行，包括两个整数  $a$  和  $b$ 。

**【输出】**

输出也仅一行，分数  $a/b$  的浮点数值（双精度浮点数，保留小数点后 9 位）。

**【输入样例】**

5 7

**【输出样例】**

0.714285714

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int a,b;
    double result;
    cin>>a>>b;
    result=1.0*a/b;
    cout<<setiosflags(ios::fixed)<<setprecision(9);
    cout<<result<<endl;
    return 0;
}
```

点赞 2

## 1.1.1.2.2 第二节 常量和变量

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第一部分：C++语言](#) > [第二章：顺序结构程序设计](#) > 第二节 常量和变量

### 1.1.1.2.2.1 甲流疫情死亡率（信息学奥赛一本通-T1011）

## 甲流疫情死亡率（信息学奥赛一本通-T1011）

原创 Alex\_McAvoy 发布于 2018-01-31 19:35:48 阅读数 6932 收藏  
展开

### 【题目描述】

甲流并不可怕，在中国，它的死亡率并不是很高。请根据截止 2009 年 12 月 22 日各省报告的甲流确诊数和死亡数，计算甲流在各省的死亡率。

### 【输入】

输入仅一行，有两个整数，第一个为确诊数，第二个为死亡数。

### 【输出】

输出仅一行，甲流死亡率，以百分数形式输出，精确到小数点后 3 位。

### 【输入样例】

10433 60

### 【输出样例】

0.575%

### 【源程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double a,b,num;
    cin>>a>>b;
    num=100*b/a;
    cout<<setiosflags(ios::fixed)<<setprecision(3);
    cout<<num<<"%"<<endl;
    return 0;
}
```

## 1.1.1.2.2.2 计算多项式的值（信息学奥赛一本通-T1012）

# 计算多项式的值（信息学奥赛一本通-T1012）

原创 Alex\_McAvoy 发布于 2018-01-31 19:36:06 阅读数 6057 收藏  
展开

**【题目描述】**

对于多项式  $f(x)=ax^3+bx^2+cx+d$  和给定的  $a,b,c,d,x$ ，计算  $f(x)$  的值，保留到小数点后 7 位。

**【输入】**

输入仅一行，包含 5 个实数，分别是  $x$ ，及参数  $a$ 、 $b$ 、 $c$ 、 $d$  的值，每个数都是绝对值不超过 100 的双精度浮点数。数与数之间以一个空格分开。

**【输出】**

输出一个实数，即  $f(x)$  的值，保留到小数点后 7 位。

**【输入样例】**

2.31 1.2 2 2 3

**【输出样例】**

33.0838692

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double x,a,b,c,d,num;
    cin>>x>>a>>b>>c>>d;
    num=a*x*x*x+b*x*x+c*x+d;
    cout<<setiosflags(ios::fixed)<<setprecision(7);
    cout<<num<<endl;
    return 0;
}
```

### 1.1.1.2.2.3 温度表达转化（信息学奥赛一本通-T1013）

## 温度表达转化（信息学奥赛一本通-T1013）

原创 Alex\_McAvoy 发布于 2018-01-31 19:36:17 阅读数 4134 收藏  
展开

**【题目描述】**

利用公式  $C = 5*(F-32)/9$  (其中  $C$  表示摄氏温度， $F$  表示华氏温度) 进行计算转化，输入华氏温度  $f$ ，输出摄氏温度  $c$ ，要求精确到小数点后 5 位。

**【输入】**

输入一行，包含一个实数  $f$ ，表示华氏温度。（ $f \geq -459.67$ ）

**【输出】**

输出一行，包含一个实数，表示对应的摄氏温度，要求精确到小数点后 5 位。

**【输入样例】**

41

**【输出样例】**

5.00000

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double c,f;
    cin>>f;
    c=5.0*(f-32)/9;
    cout<<setiosflags(ios::fixed)<<setprecision(5);
    cout<<c<<endl;
    return 0;
}
```

#### 1.1.1.2.2.4 与圆相关的计算（信息学奥赛一本通-T1014）

## 与圆相关的计算（信息学奥赛一本通-T1014）

原创 Alex\_McAvoy 发布于 2018-01-31 19:36:28 阅读数 6771 收藏  
展开

**【题目描述】**

给出圆的半径，求圆的直径、周长和面积。输入圆的半径实数  $r$ ，输出圆的直径、周长、面积，每个数保留小数点后 4 位。圆周率取值为 3.14159。

**【输入】**

输入包含一个实数  $r$  ( $0 < r \leq 10,000$ )，表示圆的半径。

**【输出】**

输出一行，包含三个数，分别表示圆的直径、周长、面积，数与数之间以一个空格分开，每个数保留小数点后 4 位。

**【输入样例】**

3.0

**【输出样例】**

6.0000 18.8495 28.2743

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double r,d,c,s,pi;
    pi=3.14159;
    cin>>r;
    d=2*r;
    c=2*pi*r;
    s=pi*r*r;
    cout<<setiosflags(ios::fixed)<<setprecision(4);
    cout<<d<<" "<<c<<" "<<s<<endl;
    return 0;
}
```

### 1.1.1.2.2.5 计算并联电阻的阻值（信息学奥赛一本通-T1015）

## 计算并联电阻的阻值（信息学奥赛一本通-T1015）

原创 Alex\_McAvoy 发布于 2018-01-31 19:36:37 阅读数 4367 收藏  
展开

**【题目描述】**

对于阻值为  $r_1$  和  $r_2$  的电阻，其并联电阻阻值公式计算见下，输入两个电阻阻抗大小，浮点型。输出并联之后的阻抗大小，结果保留小数点后 2 位。

**【输入】**

两个电阻阻抗大小，浮点型，以一个空格分开。

**【输出】**

并联之后的阻抗大小，结果保留小数点后 2 位。

**【输入样例】**

1 2

**【输出样例】**

0.67

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double r1,r2,R;
    cin>>r1>>r2;
    R=1.0/((1/r1)+(1/r2));
    cout<<setiosflags(ios::fixed)<<setprecision(2);
    cout<<R<<endl;
    return 0;
}
```

### 1.1.1.2.3 第三节 标准数据类型

#### 1.1.1.2.3.1 整型数据类型存储空间大小（信息学奥赛一本通-T1016）

## 整型数据类型存储空间大小（信息学奥赛一本通-T1016）

原创 Alex\_McAvoy 发布于 2018-01-31 19:37:04 阅读数 4679 收藏  
展开

**【题目描述】**

分别定义 `int`, `short` 类型的变量各一个，并依次输出它们的存储空间大小(单位: 字节)。

**【输入】**

(无)

**【输出】**

一行，两个整数，分别是两个变量的存储空间大小，用一个空格隔开。

**【输入样例】**

(无)

**【输出样例】**

(无)

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    cout<<sizeof(int)<<" "<<sizeof(short)<<endl;
    return 0;
}
```

### 1.1.1.2.3.2 浮点型数据类型存储空间大小（信息学奥赛一本通-T1017）

## 浮点型数据类型存储空间大小（信息学奥赛一本通-T1017）

原创 Alex\_McAvoy 发布于 2018-01-31 19:37:15 阅读数 3391 收藏  
展开

**【题目描述】**

分别定义 `float`, `double` 类型的变量各一个，并依次输出它们的存储空间大小(单位: 字节)。

**【输入】**

(无)

**【输出】**

一行，两个整数，分别是两个变量的存储空间大小，用一个空格隔开。



**【输入样例】**

(无)

**【输出样例】**

(无)

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    cout<<sizeof(float)<<" "<<sizeof(double)<<endl;
    return 0;
}
```

点赞

### 1.1.1.2.3.3 其他数据类型存储空间大小（信息学奥赛一本通-T1018）

## 其他数据类型存储空间大小（信息学奥赛一本通-T1018）

原创 Alex\_McAvoy 发布于 2018-01-31 19:37:31 阅读数 2773 收藏  
展开

**【题目描述】**

分别定义 `bool`，`char` 类型的变量各一个，并依次输出它们的存储空间大小(单位：字节)。

**【输入】**

(无)

**【输出】**

一行，两个整数，分别是两个变量的存储空间大小，用一个空格隔开。

**【输入样例】**

(无)

**【输出样例】**

(无)

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    cout<<sizeof(bool)<<" "<<sizeof(char)<<endl;
    return 0;
}
```

点赞

## 1.1.1.2.3.4 浮点数向零舍入（信息学奥赛一本通-T1019）

## 浮点数向零舍入（信息学奥赛一本通-T1019）

原创 Alex\_McAvoy 发布于 2018-01-31 19:38:33 阅读数 5276 收藏  
展开

**【题目描述】**

输入一个单精度浮点数，将其向零舍入到整数。说明：向零舍入的含义是，正数向下舍入，负数向上舍入。提示：可以使用强制类型转换来实现。

**【输入】**

一个单精度浮点数。

**【输出】**

一个整数，即向零舍入到整数的结果。

**【输入样例】**

2.3

**【输出样例】**

2

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    double n;
    cin>>n;
```

```

    cout<<int(n)<<endl;
    return 0;
}

```

### 1.1.1.2.3.5 打印 ASCII 码（信息学奥赛一本通-T1020）

## 打印 ASCII 码（信息学奥赛一本通-T1020）

原创 Alex\_McAvoy 发布于 2018-01-31 19:38:42 阅读数 4474 收藏  
展开

#### 【题目描述】

输入一个除空格以外的可见字符（保证在函数 `scanf` 中可使用格式说明符 `%c` 读入），输出其 ASCII 码。

#### 【输入】

一个除空格以外的可见字符。

#### 【输出】

一个十进制整数，即该字符的 ASCII 码。

#### 【输入样例】

A

#### 【输出样例】

65

#### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    char ch;
    cin>>ch;
    cout<<int(ch)<<endl;
    return 0;
}

```

点赞

### 1.1.1.2.3.6 打印字符（信息学奥赛一本通-T1021）

## 打印字符（信息学奥赛一本通-T1021）

原创 Alex\_McAvoy 发布于 2018-02-01 10:43:18 阅读数 4651 收藏  
展开

#### 【题目描述】

输入一个整数，即字符的 ASCII 码，保证存在对应的可见字符。输出相对应的字符。

#### 【输入】

一个整数，即字符的 ASCII 码，保证存在对应的可见字符。

#### 【输出】

一行，包含相应的字符。

#### 【输入样例】

65

#### 【输出样例】

A

#### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    cout<<char(n)<<endl;
    return 0;
}
```

### 1.1.1.2.3.7 整型与布尔型的转换（信息学奥赛一本通-T1022）

## 整型与布尔型的转换（信息学奥赛一本通-T1022）

原创 Alex\_McAvoy 发布于 2018-02-01 10:43:28 阅读数 4358 收藏  
展开

**【题目描述】**

将一个整型变量的值赋给一个布尔型变量，再将这个布尔型变量的值赋给一个整型变量，得到的值是多少？

**【输入】**

一个整型范围内的整数，即初始时整型变量的值。

**【输出】**

一个整数，经过上述过程后得到的结果。

**【输入样例】**

3

**【输出样例】**

1

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    cout<<bool(n)<<endl;
    return 0;
}
```

### 1.1.1.2.3.8 Hello,World!的大小（信息学奥赛一本通-T1023）

## Hello,World!的大小（信息学奥赛一本通-T1023）

原创 Alex\_McAvoy 发布于 2018-02-01 10:43:37 阅读数 6993 收藏  
展开

**【题目描述】**

我们曾经输出过的“Hello, World!”吗？它虽然不是本章所涉及的基本数据类型的数据，但我们同样可以用 `sizeof` 函数获得它所占用的空间大小。

**【输入】**

(无)

**【输出】**

一个整数，即“Hello, World!”的大小。

**【输入样例】**

(无)

**【输出样例】**

(无)

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    cout<<sizeof("Hello, World!")<<endl;
    return 0;
}
```

## 1.1.1.2.4 第四节 数据输入输出

### 1.1.1.2.4.1 保留 3 位小数的浮点数（信息学奥赛一本通-T1024）

## 保留 3 位小数的浮点数（信息学奥赛一本通-T1024）

原创 Alex\_McAvoy 发布于 2018-02-01 10:43:47 阅读数 4077 收藏  
展开

**【题目描述】**

读入一个单精度浮点数，保留 3 位小数输出这个浮点数。

**【输入】**

只有一行，一个单精度浮点数。

**【输出】**

也只有一行，读入的单精度浮点数。

**【输入样例】**

12.34521

**【输出样例】**

12.345

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    float n;
    cin>>n;
    cout<<setiosflags(ios::fixed)<<setprecision(3);
    cout<<n<<endl;
    return 0;
}
```

### 1.1.1.2.4.2 保留 12 位小数的浮点数（信息学奥赛一本通-T1025）

## 保留 12 位小数的浮点数（信息学奥赛一本通-T1025）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:00 阅读数 2895 收藏  
展开

**【题目描述】**

读入一个双精度浮点数，保留 12 位小数输出这个浮点数。

**【输入】**

只有一行，一个双精度浮点数。

**【输出】**

也只有一行，读入的双精度浮点数。

**【输入样例】**

3.1415926535798932

**【输出样例】**

3.141592653580

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double n;
    cin>>n;
    cout<<setiosflags(ios::fixed)<<setprecision(12);
    cout<<n<<endl;
    return 0;
}
```

## 1.1.1.2.4.3 空格分隔输出（信息学奥赛一本通-T1026）

## 空格分隔输出（信息学奥赛一本通-T1026）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:15 阅读数 3875 收藏  
展开

**【题目描述】**

读入一个字符，一个整数，一个单精度浮点数，一个双精度浮点数，然后按顺序输出它们，并且要求在他们之间用一个空格分隔。输出浮点数时保留 6 位小数。

**【输入】**

第一行是一个字符；

第二行是一个整数；

第三行是一个单精度浮点数；

第四行是一个双精度浮点数；

**【输出】**

输出字符、整数、单精度浮点数和双精度浮点数，之间用空格分隔。

**【输入样例】**

```
a
12
2.3
3.2
```



**【输出样例】**

a 12 2.300000 3.200000

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    char ch;
    int a;
    float b;
    double c;
    cin>>ch>>a>>b>>c;
    cout<<ch<<" "<<a<<" ";
    cout<<setiosflags(ios::fixed)<<setprecision(6);
    cout<<b<<" "<<c<<endl;
    return 0;
}
```

## 1.1.1.2.4.4 输出浮点数（信息学奥赛一本通-T1027）

## 输出浮点数（信息学奥赛一本通-T1027）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:29 阅读数 5048 收藏  
展开

**【题目描述】**

读入一个双精度浮点数，分别按输出格式“%f”，“%f”保留 5 位小数，“%e”和“%g”的形式输出这个整数，每次在单独一行上输出。

**【输入】**

一个双精度浮点数。

**【输出】**

第一行是按“%f”输出的双精度浮点数；

第二行是按“%f”保留 5 位小数输出的双精度浮点数；

第三行是按“%e”输出的双精度浮点数；

第四行是按“%g”输出的双精度浮点数。

**【输入样例】**

12.3456789

**【输出样例】**

12.345679  
 12.34568  
 1.234568e+001  
 12.3457

**【源程序】**

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    double n;
    cin>>n;
    printf("%f\n",n);
    printf("%.5f\n",n);
    printf("%e\n",n);
    printf("%g\n",n);
    return 0;
}
```

点赞 3  
 收藏

## 1.1.1.2.4.5 字符菱形（信息学奥赛一本通-T1028）

## 字符菱形（信息学奥赛一本通-T1028）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:39 阅读数 3979 收藏  
 展开

**【题目描述】**

给定一个字符，用它构造一个对角线长 5 个字符，倾斜放置的菱形。

**【输入】**

输入只有一行， 包含一个字符。

**【输出】**

该字符构成的菱形。

**【输入样例】**

\*

**【输出样例】**

\*

```
***
*****
***
*
```

## 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    char ch;
    cin>>ch;
    cout<<" "<<ch<<endl;
    cout<<" "<<ch<<ch<<ch<<endl;
    cout<<ch<<ch<<ch<<ch<<ch<<endl;
    cout<<" "<<ch<<ch<<ch<<endl;
    cout<<" "<<ch<<endl;
    return 0;
}
```

点赞

### 1.1.1.2.5 第五节 顺序结构实例

#### 1.1.1.2.5.1 计算浮点数相除的余（信息学奥赛一本通-T1029）

## 计算浮点数相除的余（信息学奥赛一本通-T1029）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:48 阅读数 7452 收藏  
展开

## 【题目描述】

计算两个双精度浮点数  $a$  和  $b$  的相除的余数， $a$  和  $b$  都是双精度浮点数。这里余数 ( $r$ ) 的定义是： $a=k*b+r$ ，其中  $k$  是整数， $0\leq r<b$ 。

## 【输入】

输入仅一行，包括两个双精度浮点数  $a$  和  $b$ 。

**【输出】**

输出也仅一行， $a \div b$  的余数。

**【输入样例】**

73.263 0.9973

**【输出样例】**

0.4601

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    double a, b;
    cin>>a>>b;
    double r=a-int(a/b)*b;//根据所给公式计算，其中 k=int(a/b)
    if(r<0)
        r+=b;
    cout<<r<<endl;
}
```

### 1.1.1.2.5.2 计算球的体积（信息学奥赛一本通-T1030）

## 计算球的体积（信息学奥赛一本通-T1030）

原创 Alex\_McAvoy 发布于 2018-02-01 10:44:57 阅读数 4746 收藏  
展开

**【题目描述】**

对于半径为  $r$  的球，其体积的计算公式为  $v = \frac{4}{3} \pi r^3$  这里取  $\pi = 3.14$ 。现给定  $r$ ，即球半径，类型为 `double`，求球的体积  $V$ ，保留到小数点后 2 位。

**【输入】**

输入为一个不超过 100 的非负实数，即球半径，类型为 `double`。

**【输出】**

输出一个实数，即球的体积，保留到小数点后 2 位。

**【输入样例】**

4

**【输出样例】**

267.95

**【源程序】**

```

#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double r,v;
    cin>>r;
    v=(4*3.14*r*r*r)/3.0;
    cout<<setiosflags(ios::fixed)<<setprecision(2);
    cout<<v<<endl;
    return 0;
}

```

### 1.1.1.2.5.3 反向输出一个三位数（信息学奥赛一本通-T1031）

## 反向输出一个三位数（信息学奥赛一本通-T1031）

原创 Alex\_McAvoy 发布于 2018-02-02 14:12:57 阅读数 6787 收藏  
展开

**【题目描述】**

将一个三位数反向输出，例如输入 358，反向输出 853。

**【输入】**

一个三位数 n。

**【输出】**

反向输出 n。

**【输入样例】**

100

**【输出样例】**

001

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int a,b,c;
    cin>>n;
    a=n/100;
    b=n%100/10;
    c=n%10;
    cout<<c<<b<<a<<endl;
    return 0;
}
```

## 1.1.1.2.5.4 大象喝水（信息学奥赛一本通-T1032）

## 大象喝水（信息学奥赛一本通-T1032）

原创 Alex\_McAvoy 发布于 2018-02-02 14:13:10 阅读数 6874 收藏  
展开

**【题目描述】**

一只大象口渴了，要喝 20 升水才能解渴，但现在只有一个深  $h$  厘米，底面半径为  $r$  厘米的小圆桶( $h$  和  $r$  都是整数)。问大象至少要喝多少桶水才会解渴。

**【输入】**

输入有一行：包行两个整数，以一个空格分开，分别表示小圆桶的深  $h$  和底面半径  $r$ ，单位都是厘米。

**【输出】**

输出一行，包含一个整数，表示大象至少要喝水的桶数。

**【输入样例】**

23 11

**【输出样例】**

3

**【源程序】**

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
```

```

{
    double r,c,h,pi,v,water;
    pi=3.14159;
    cin>>h>>r;
    v=pi*r*r*h/1000.0;
    water=20.0/v;
    c=ceil(water);//ceil 作用：返回大于或者等于指定表达式的最小整数
    cout<<c<<endl;
    return 0;
}

```

#### 1.1.1.2.5.5 计算线段长度（信息学奥赛一本通-T1033）

## 计算线段长度（信息学奥赛一本通-T1033）

原创 Alex\_McAvoy 发布于 2018-02-02 14:13:33 阅读数 5878 收藏  
展开

### 【题目描述】

已知线段的两个端点的坐标  $A(X_a, Y_a), B(X_b, Y_b)$ ，求线段  $AB$  的长度，保留到小数点后 3 位。

### 【输入】

第一行是两个实数  $X_a, Y_a$ ，即  $A$  的坐标。

第二行是两个实数  $X_b, Y_b$ ，即  $B$  的坐标。

输入中所有实数的绝对值均不超过 10000。

### 【输出】

一个实数，即线段  $AB$  的长度，保留到小数点后 3 位。

### 【输入样例】

```
1 1 2 2
```

### 【输出样例】

```
1.414
```

### 【源程序】

```

#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;

```

```

int main()
{
    double Xa,Xb,Ya,Yb;
    double len;
    cin>>Xa>>Ya;
    cin>>Xb>>Yb;
    len=sqrt( (Xa-Xb)*(Xa-Xb) + (Ya-Yb)*(Ya-Yb) );
    cout<<setiosflags(ios::fixed)<<setprecision(3);
    cout<<len<<endl;
    return 0;
}

```

#### 1.1.1.2.5.6 计算三角形面积（信息学奥赛一本通- T1034）

## 计算三角形面积（信息学奥赛一本通-T1034）

原创 Alex\_McAvoy 发布于 2018-02-02 14:20:26 阅读数 6854 收藏  
展开

### 【题目描述】

平面上有一个三角形，它的三个顶点坐标分别为 $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ ，那么请问这个三角形的面积是多少，精确到小数点后两位。

### 【输入】

输入仅一行，包括 6 个单精度浮点数，分别对应  $x_1, y_1, x_2, y_2, x_3, y_3$ 。

### 【输出】

输出也是一行，输出三角形的面积，精确到小数点后两位。

### 【输入样例】

0 0 4 0 0 3

### 【输出样例】

6.00

### 【源程序】

```

#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;
int main()
{

```



```
double x1,x2,x3,y1,y2,y3,a,b,c,area,p;
cin>>x1>>y1>>x2>>y2>>x3>>y3;
    a=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));//计算边长 a
    b=sqrt((x3-x1)*(x3-x1)+(y3-y1)*(y3-y1));//计算边长 b
    c=sqrt((x2-x3)*(x2-x3)+(y2-y3)*(y2-y3));//计算边长 c
    p=(a+b+c)/2.0;//计算半周长
    area=sqrt(p*(p-a)*(p-b)*(p-c));//海伦公式求面积
    cout<<setiosflags(ios::fixed)<<setprecision(2)<<area<<endl;
    return 0;
}
```

#### 1.1.1.2.5.7 等差数列末项计算（信息学奥赛一本通-T1035）

## 等差数列末项计算（信息学奥赛一本通-T1035）

原创 Alex\_McAvoy 发布于 2018-02-02 14:27:54 阅读数 3492 收藏  
展开

### 【题目描述】

给出一个等差数列的前两项  $a_1$ ,  $a_2$ , 求第  $n$  项是多少。。

### 【输入】

一行, 包含三个整数  $a_1$ ,  $a_2$ ,  $n$ 。  $-100 \leq a_1, a_2 \leq 100$ ,  $0 < n \leq 1000$ 。

### 【输出】

一个整数, 即第  $n$  项的值。。

### 【输入样例】

1 4 100

### 【输出样例】

298

### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int a_1,a_2,n;
    int a_n,d;
    cin>>a_1>>a_2>>n;
```

```

d=a_2-a_1;//求公差
a_n=a_1+(n-1)*d;//等差数列通项公式
cout<<a_n<<endl;
return 0;
}

```

点赞

#### 1.1.1.2.5.8 A\*B 问题（信息学奥赛一本通-T1036）

## A\*B 问题（信息学奥赛一本通-T1036）

原创 Alex\_McAvoy 发布于 2018-02-02 14:31:42 阅读数 4410 收藏  
展开

### 【题目描述】

输入两个正整数 A 和 B，求 A\*B 的值。注意乘积的范围和数据类型的选择。

### 【输入】

一行，包含两个正整数 A 和 B，中间用单个空格隔开。 $1 \leq A, B \leq 50000$ 。

### 【输出】

两个整数的积。

### 【输入样例】

3 4

### 【输出样例】

12

### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    long long a,b,product;//由题意知变量应定义为超长整型变量
    cin>>a>>b;
    product=a*b;//两数之积
    cout<<product<<endl;
    return 0;
}

```

点赞 2

### 1.1.1.2.5.9 计算 2 的幂（信息学奥赛一本通-T1037）

## 计算 2 的幂（信息学奥赛一本通-T1037）

原创 Alex\_McAvoy 发布于 2018-02-02 15:01:03 阅读数 5159 收藏  
展开

#### 【题目描述】

非负整数  $n$ ，求  $2^n$ ，即 2 的  $n$  次方。。

#### 【输入】

一个整数  $n$ 。  $0 \leq n < 31$ 。

#### 【输出】

一个实数，即线段 AB 的长度，保留到小数点后 3 位。

#### 【输入样例】

2

#### 【输出样例】

8

#### 【源程序】

```
#include<iostream>
#include<cmath>//pow()函数需调用<cmath>
using namespace std;
int main()
{
    int n,result;
    cin>>n;
    result=pow(2,n);//power(x,y)，求  $x^y$ ，其中  $x$  为底数， $y$  为指数
    cout<<result<<endl;
    return 0;
}
```

点赞

## 1.1.1.2.5.10 苹果和虫子（信息学奥赛一本通-T1038）

## 苹果和虫子（信息学奥赛一本通-T1038）

原创 Alex\_McAvoy 发布于 2018-02-02 15:06:46 阅读数 5964 收藏  
展开

**【题目描述】**

你买了一箱  $n$  个苹果，很不幸的是买完时箱子里混进了一条虫子。虫子每  $x$  小时能吃掉一个苹果，假设虫子在吃完一个苹果之前不会吃另一个，那么经过  $y$  小时你还有多少个完整的苹果？

**【输入】**

输入仅一行，包括  $n$ ， $x$  和  $y$ （均为整数）。

**【输出】**

输出也仅一行，剩下的苹果个数。

**【输入样例】**

10 4 9

**【输出样例】**

7

**【源程序】**

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    double n,x,y,c;
    cin>>n>>x>>y;
    c=n-ceil(y*1.0/x);//ceil 作用：返回大于或者等于指定表达式的最小整数
    if(c<0)
        c=0;//需考虑 c<0 的情况
    cout<<c<<endl;
    return 0;
}
```

## 1.1.1.2.5.11 新建项目

## 1.1.1.3 第三章 程序的控制结构

### 1.1.1.3.1 第一节 if 选择结构

#### 1.1.1.3.1.1 判断数正负（信息学奥赛一本通-T1039）

### 判断数正负（信息学奥赛一本通-T1039）

原创 Alex\_McAvoy 发布于 2018-02-04 13:42:08 阅读数 6316 收藏  
展开

#### 【题目描述】

给定一个整数  $N$ ，判断其正负。如果  $N > 0$ ，输出 `positive`；如果  $N = 0$ ，输出 `zero`；如果  $N < 0$ ，输出 `negative`。

#### 【输入】

输入一个整数  $N$  ( $-10^9 \leq N \leq 10^9$ )。

#### 【输出】

如果  $N > 0$ ，输出 `positive`

如果  $N = 0$ ，输出 `zero`

如果  $N < 0$ ，输出 `negative`

#### 【输入样例】

1

#### 【输出样例】

`positive`

#### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n;
    (n == 0) ? (cout << "zero") : (n > 0 ? (cout << "positive") : (cout << "negative"));
    cout << endl;
```

```
    return 0;
}
```

点赞

### 1.1.1.3.1.2 输出绝对值（信息学奥赛一本通-T1040）

## 输出绝对值（信息学奥赛一本通-T1040）

原创 Alex\_McAvoy 发布于 2018-02-04 14:18:56 阅读数 5158 收藏  
展开

#### 【题目描述】

输入一个浮点数，输出这个浮点数的绝对值，保留到小数点后两位。

#### 【输入】

输入一个浮点数，其绝对值不超过 10000。

#### 【输出】

输出这个浮点数的绝对值，保留到小数点后两位。

#### 【输入样例】

-3.14

#### 【输出样例】

3.14

#### 【源程序】

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    double N;
    cin>>N;
    cout<<setiosflags(ios::fixed)<<setprecision(2);
    if(N>=0) cout<<N<<endl;
    else cout<<-N<<endl;
    return 0;
}
```

### 1.1.1.3.1.3 奇偶数判断（信息学奥赛一本通-T1041）

#### 奇偶数判断（信息学奥赛一本通-T1041）

原创 Alex\_McAvoy 发布于 2018-02-04 14:19:43 阅读数 3802 收藏  
展开

##### 【题目描述】

给定一个整数，判断该数是奇数还是偶数。如果  $n$  是奇数，输出 `odd`；如果  $n$  是偶数，输出 `even`。

##### 【输入】

输入仅一行，一个大于零的正整数  $n$ 。

##### 【输出】

输出仅一行，如果  $n$  是奇数，输出 `odd`；如果  $n$  是偶数，输出 `even`。

##### 【输入样例】

5

##### 【输出样例】

odd

##### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int N;
    cin>>N;
    if(N%2) cout<<"odd"<<endl;
    else cout<<"even"<<endl;
    return 0;
}
```

### 1.1.1.3.1.4 奇偶 ASCII 值判断（信息学奥赛一本通-T1042）

#### 奇偶 ASCII 值判断（信息学奥赛一本通-T1042）

原创 Alex\_McAvoy 发布于 2018-02-05 02:05:47 阅读数 4757 收藏  
展开

**【题目描述】**

任意输入一个字符，判断其 ASCII 是否是奇数，若是，输出 YES，否则，输出 NO。

**【输入】**

输入一个字符。

**【输出】**

如果其 ASCII 值为奇数，则输出 YES，否则，输出 NO。

**【输入样例】**

A

**【输出样例】**

YES

**【源程序】**

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    char ch;
    scanf("%c",&ch);
    if (ch%2 == 0) cout << "NO" << endl;
    else cout << "YES" << endl;
    return 0;
}
```

点赞

### 1.1.1.3.1.5 整数大小比较（信息学奥赛一本通-T1043）

#### 整数大小比较（信息学奥赛一本通-T1043）

原创 Alex\_McAvoy 发布于 2018-02-05 02:05:59 阅读数 4926 收藏  
展开

**【题目描述】**

输入两个整数，比较它们的大小。若  $x > y$ ，输出  $>$ ；若  $x = y$ ，输出  $=$ ；若  $x < y$ ，输出  $<$ 。

**【输入】**

一行，包含两个整数  $x$  和  $y$ ，中间用单个空格隔开。 $0 \leq x < 2^{32}$ ,  $-2^{31} \leq y < 2^{31}$ 。



**【输出】**

$x > y$ ，输出  $>$ ；若  $x = y$ ，输出  $=$ ；若  $x < y$ ，输出  $<$ ；

**【输入样例】**

1000 100

**【输出样例】**

$>$

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cin>>x>>y;
    if(x>y) cout<<">"<<endl;
    else if(x<y) cout<<"<"<<endl;
    else if(x==y) cout<<"="<<endl;
    return 0;
}
```

### 1.1.1.3.1.6 判断是否为两位数（信息学奥赛一本通-T1044）

#### 判断是否为两位数（信息学奥赛一本通-T1044）

原创 Alex\_McAvoy 发布于 2018-02-05 02:06:05 阅读数 4276 收藏  
展开

**【题目描述】**

判断一个正整数是否是两位数(即大于等于 10 且小于等于 99)。若该正整数是两位数，输出 1，否则输出 0。

**【输入】**

一个正整数，不超过 1000。

**【输出】**

一行。若该正整数是两位数，输出 1，否则输出 0。

**【输入样例】**

54

**【输出样例】**

1

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if(n>=10&&n<=99) cout<<1<<endl;
    else cout<<0<<endl;
    return 0;
}
```

### 1.1.1.3.1.7 收集瓶盖赢大奖（信息学奥赛一本通-T1045）

#### 收集瓶盖赢大奖（信息学奥赛一本通-T1045）

原创 Alex\_McAvoy 发布于 2018-02-05 02:06:16 阅读数 4885 收藏  
展开

**【题目描述】**

某饮料公司最近推出了一个“收集瓶盖赢大奖”的活动：如果你拥有 10 个印有“幸运”、或 20 个印有“鼓励”的瓶盖，就可以兑换一个神秘大奖。现分别给出你拥有的印有“幸运”和“鼓励”的瓶盖数，判断是否可以去兑换大奖。若可以兑换大奖，输出 1，否则输出 0。

**【输入】**

一行，包含两个整数，分别是印有“幸运”和“鼓励”的瓶盖数，用一个空格隔开。

**【输出】**

一行。若可以兑换大奖，输出 1，否则输出 0。

**【输入样例】**

11 19

**【输出样例】**

1

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cin>>x>>y;
```

```

    if(x>=10||y>=20) cout<<1<<endl;
    else cout<<0<<endl;
    return 0;
}

```

### 1.1.1.3.1.8 判断一个数能否同时被 3 和 5 整除（信息学奥赛一本通-T1046）

#### 判断一个数能否同时被 3 和 5 整除（信息学奥赛一本通-T1046）

原创 Alex\_McAvoy 发布于 2018-02-05 02:06:33 阅读数 4503 收藏  
展开

##### 【题目描述】

判断一个数  $n$  能否同时被 3 和 5 整除，如果能同时被 3 和 5 整除输出 YES，否则输出 NO。

##### 【输入】

输入一行，包含一个整数  $n$ 。（ $-1,000,000 < n < 1,000,000$ ）

##### 【输出】

输出一行，如果能同时被 3 和 5 整除输出 YES，否则输出 NO。

##### 【输入样例】

15

##### 【输出样例】

YES

##### 【源程序】

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if((n%3==0)&&(n%5==0)) cout<<"YES"<<endl;
    else cout<<"NO"<<endl;
    return 0;
}

```

点赞 1

### 1.1.1.3.1.9 判断能否被 3，5，7 整除（信息学奥赛一本通-T1047）

#### 判断能否被 3，5，7 整除（信息学奥赛一本通-T1047）

原创 Alex\_McAvoy 发布于 2018-02-05 02:09:47 阅读数 8787 收藏  
展开

##### 【题目描述】

给定一个整数，判断它能否被 3，5，7 整除，并输出以下信息：

- 1、能同时被 3，5，7 整除（直接输出 3 5 7，每个数中间一个空格）；
- 2、只能被其中两个数整除（输出两个数，小的在前，大的在后。例如：3 5 或者 3 7 或者 5 7，中间用空格分隔）；
- 3、只能被其中一个数整除（输出这个除数）；
- 4、不能被任何数整除，输出小写字符'n'，不包括单引号。

##### 【输入】

输入一行，包括一个整数。

##### 【输出】

输出一行，按照描述要求给出整数被 3，5，7 整除的情况。

##### 【输入样例】

105

##### 【输出样例】

3 5 7

##### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int n,t;
    cin>>n;
    t=0;
    if(n%3==0)
        t=1;//能被 3 整除
    if(n%5==0)
        t=2;//能被 5 整除
```

```

if(n%7==0)
    t=3;//能被 7 整除
if((n%3==0)&&(n%5==0))
    t=4;//能被 3、5 整除
if((n%3==0)&&(n%7==0))
    t=5;//能被 3、7 整除
if((n%5==0)&&(n%7==0))
    t=6;//能被 5、7 整除
if((n%3==0)&&(n%5==0)&&(n%7==0))
    t=7;//能被 3、5、7 整除
switch (t)
{
    case 1:
        cout<<"3"<<endl;
        break;
    case 2:
        cout<<"5"<<endl;
        break;
    case 3:
        cout<<"7"<<endl;
        break;
    case 4:
        cout<<"3 5"<<endl;
        break;
    case 5:
        cout<<"3 7"<<endl;
        break;
    case 6:
        cout<<"5 7"<<endl;
        break;
    case 7:
        cout<<"3 5 7"<<endl;
        break;
    default:
        cout<<"n"<<endl;
}
return 0;
}

```

### 1.1.1.3.1.10 有一门课不及格的学生（信息学奥赛一本通-T1048）

有一门课不及格的学生（信息学奥赛一本通-T1048）

原创 Alex\_McAvoy 发布于 2018-02-05 02:10:09 阅读数 7104 收藏  
展开

#### 【题目描述】

给出一名学生的语文和数学成绩，判断他是否恰好有一门课不及格(成绩小于 60 分)。若该生恰好有一门课不及格，输出 1；否则输出 0。

#### 【输入】

一行，包含两个在 0 到 100 之间的整数，分别是该生的语文成绩和数学成绩。

#### 【输出】

若该生恰好有一门课不及格，输出 1；否则输出 0。

#### 【输入样例】

50 80

#### 【输出样例】

1

#### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int chinese,math;
    cin>>chinese>>math;
    if( (chinese>=60&&math<60) || (math>=60&&chinese<60) ) cout<<1<<endl;
    else cout<<0<<endl;
    return 0;
}
```

## 1.1.1.3.2 第二节 switch 语句

### 1.1.1.3.2.1 晶晶赴约会（信息学奥赛一本通-T1049）

#### 晶晶赴约会（信息学奥赛一本通-T1049）

原创 Alex\_McAvoy 发布于 2018-02-05 14:00:39 阅读数 5216 收藏  
展开

**【题目描述】**

晶晶的朋友贝贝约晶晶下周一起去看展览，但晶晶每周的 1、3、5 有课必须上课，请帮晶晶判断她能否接受贝贝的邀请，如果能输出 YES；如果不能则输出 NO。注意 YES 和 NO 都是大写字母！

**【输入】**

输入有一行，贝贝邀请晶晶去看展览的日期，用数字 1 到 7 表示从星期一到星期日。

**【输出】**

输出有一行，如果晶晶可以接受贝贝的邀请，输出 YES，否则，输出 NO。注意 YES 和 NO 都是大写字母！

**【输入样例】**

2

**【输出样例】**

YES

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if((n==1)||n==3||n==5) cout<<"NO"<<endl;
    else cout<<"YES"<<endl;
    return 0;
}
```

### 1.1.1.3.2.2 骑车与走路（信息学奥赛一本通-T1050）

#### 骑车与走路（信息学奥赛一本通-T1050）

原创 Alex\_McAvoy 发布于 2018-02-05 14:01:02 阅读数 5797 收藏  
展开

**【题目描述】**

你买了一箱  $n$  个苹果，很不幸的是买完时箱子里混进了一条虫子。虫子每  $x$  小时能吃掉一个苹果，假设虫子在吃完一个苹果之前不会吃另一个，那么经过  $y$  小时你还有多少个完整的苹果？

在清华校园里，没有自行车，上课办事会很不方便。但实际上。并非去办任何事情都是骑车快,因为骑车总要找车、开锁、停车、锁车等,这要耽误一些时间。假设找到自

行车，开锁并车上自行车的时间为 27 秒；停车锁车的时间为 23 秒；步行每秒行走 1.2 米，骑车每秒行走 3.0 米。请判断走不同的距离去办事，是骑车快还是走路快。如果骑车快，输出一行"Bike"；如果走路快，输出一行"Walk"；如果一样快，输出一行"All"。

#### 【输入】

输入仅一行，包括 n，x 和 y（均为整数）。

输入一行，包含一个整数，表示一次办事要行走的距离,单位为米。

#### 【输出】

输出也仅一行，剩下的苹果个数。

输出一行,如果骑车快,输出一行"Bike";如果走路快,输出一行"Walk";如果一样快,输出一行"All"。

#### 【输入样例】

120

#### 【输出样例】

Bike

#### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int s;
    double bike,walk;
    cin>>s;
    bike=s/3.0+50;
    walk=s/1.2;
    if(bike==walk) cout<<"All"<<endl;
    else if(bike>walk) cout<<"Walk"<<endl;
    else cout<<"Bike"<<endl;
    return 0;
}
```

### 1.1.1.3.2.3 分段函数（信息学奥赛一本通-T1051）

#### 分段函数（信息学奥赛一本通-T1051）

原创 Alex\_McAvoy 发布于 2018-02-06 14:47:31 阅读数 5228 收藏  
展开



**【题目描述】**

编写程序，计算下列分段函数  $y=f(x)$  的值。

$$y=-x+2.5; 0 \leq x < 5$$

$$y=2-1.5(x-3)(x-3); 5 \leq x < 10$$

$$y=x/2-1.5; 10 \leq x < 20$$

**【输入】**

一个浮点数  $N$ ， $0 \leq N < 20$ 。

**【输出】**

输出  $N$  对应的分段函数值： $f(N)$ 。结果保留到小数点后三位。

**【输入样例】**

1.0

**【输出样例】**

1.500

**【源程序】**

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    double x,y,t;
    cin>>x;
    t=int(x)/5.0;
    switch(int (t))
    {
        case 0:
            y=(-x)+2.5;
            break;
        case 1:
            y=2-1.5*(x-3)*(x-3);
            break;
        case 2:
        case 3:
            y=x/2-1.5;
            break;
    }
    cout<<setiosflags(ios::fixed)<<setprecision(3)<<y<<endl;
    return 0;
}
```

### 1.1.1.3.2.4 计算邮资（信息学奥赛一本通-T1052）

#### 计算邮资（信息学奥赛一本通-T1052）

原创 Alex\_McAvoy 发布于 2018-02-06 14:47:38 阅读数 9266 收藏  
展开

##### 【题目描述】

根据邮件的重量和用户是否选择加急计算邮费。计算规则：重量在 1000 克以内(包括 1000 克)，基本费 8 元。超过 1000 克的部分，每 500 克加收超重费 4 元，不足 500 克部分按 500 克计算；如果用户选择加急，多收 5 元。

##### 【输入】

输入一行，包含整数和一个字符，以一个空格分开，分别表示重量（单位为克）和是否加急。如果字符是 y，说明选择加急；如果字符是 n，说明不加急。

##### 【输出】

输出一行，包含一个整数，表示邮费。

##### 【输入样例】

1200 y

##### 【输出样例】

17

##### 【源程序】

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int m,money;
    char ch;
    cin>>m>>ch;
    if(m>1000) money=8+ceil((m-1000)/500.0)*4;
    else money=8;
    if(ch=='y') money=money+5;
    cout<<money<<endl;
    return 0;
}
```

### 1.1.1.3.2.5 最大数输出（信息学奥赛一本通-T1053）

#### 最大数输出（信息学奥赛一本通-T1053）

原创 Alex\_McAvoy 发布于 2018-02-06 14:48:16 阅读数 4593 收藏  
展开

##### 【题目描述】

输入三个整数，数与数之间以一个空格分开。 输出一个整数，即最大的整数。

##### 【输入】

输入为一行，包含三个整数，数与数之间以一个空格分开。

##### 【输出】

输出一行，包含一个整数，即最大的整数。

##### 【输入样例】

10 20 56

##### 【输出样例】

56

##### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    int max;
    cin>>a>>b>>c;
    if(a>b) max=a;
    else max=b;
    if(c>max) max=c;
    cout<<max<<endl;
    return 0;
}
```

点赞 1

收藏

### 1.1.1.3.2.6 三角形判断（信息学奥赛一本通-T1054）

#### 三角形判断（信息学奥赛一本通-T1054）

原创 Alex\_McAvoy 发布于 2018-02-06 14:48:59 阅读数 4799 收藏  
展开

#### 【题目描述】

给定三个正整数，分别表示三条线段的长度，判断这三条线段能否构成一个三角形。如果能构成三角形，则输出“yes”，否则输出“no”。

#### 【输入】

输入共一行，包含三个正整数，分别表示三条线段的长度，数与数之间以一个空格分开。

#### 【输出】

如果能构成三角形，则输出“yes”，否则输出“no”。

#### 【输入样例】

3 4 5

#### 【输出样例】

yes

#### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    if((a+b>c)&&(a+c>b)&&(b+c>a)) cout<<"yes"<<endl;
    else cout<<"no"<<endl;
    return 0;
}
```

### 1.1.1.3.2.7 判断闰年（信息学奥赛一本通-T1055）

#### 判断闰年（信息学奥赛一本通-T1055）

原创 Alex\_McAvoy 发布于 2018-02-06 14:49:36 阅读数 5764 收藏  
展开

#### 【题目描述】

判断某年是否是闰年。如果公元  $a$  年是闰年输出 Y，否则输出 N。

#### 【输入】

输入只有一行，包含一个整数  $a(0 < a < 3000)$ 。

**【输出】**

一行，如果公元  $a$  年是闰年输出 Y，否则输出 N。

**【输入样例】**

2006

**【输出样例】**

N

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin>>a;
    if(a%4==0)
    {
        if(a%100==0)
        {
            if(a%400) cout<<"N"<<endl;
            else cout<<"Y"<<endl;
        }
        else cout<<"Y"<<endl;
    }
    else cout<<"N"<<endl;
    return 0;
}
```

### 1.1.1.3.2.8 点和正方形的关系（信息学奥赛一本通-T1056）

#### 点和正方形的关系（信息学奥赛一本通-T1056）

原创 Alex\_McAvoy 发布于 2018-02-06 14:49:46 阅读数 5206 收藏  
展开

**【题目描述】**

有一个正方形，四个角的坐标  $(x,y)$  分别是  $(1, -1)$ ,  $(1, 1)$ ,  $(-1, -1)$ ,  $(-1, 1)$ ,  $x$  是横轴， $y$  是纵轴。写一个程序，判断一个给定的点是否在这个正方形内(包括正方形边界)。如果点在正方形内，则输出 yes，否则输出 no。

**【输入】**

输入一行，包括两个整数  $x$ 、 $y$ ，以一个空格分开，表示坐标  $(x,y)$ 。

**【输出】**

输出一行，如果点在正方形内，则输出 **yes**，否则输出 **no**。

**【输入样例】**

1 1

**【输出样例】**

yes

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    double x,y;
    cin>>x>>y;
    if( (x>=-1&& x<=1) && (y>=-1&& y<=1) ) cout<<"yes"<<endl;
    else cout<<"no"<<endl;
    return 0;
}
```

### 1.1.1.3.2.9 简单计算器（信息学奥赛一本通-T1057）

#### 简单计算器（信息学奥赛一本通-T1057）

原创 Alex\_McAvoy 发布于 2018-02-06 14:50:02 阅读数 7762 收藏  
展开

**【题目描述】**

一个最简单的计算器，支持 $+$ 、 $-$ 、 $*$ 、 $/$  四种运算。仅需考虑输入输出为整数的情况，数据和运算结果不会超过 `int` 表示的范围。然而：

1. 如果出现除数为 0 的情况，则输出：Divided by zero!
2. 如果出现无效的操作符(即不为  $+$ 、 $-$ 、 $*$ 、 $/$  之一)，则输出：Invalid operator!

**【输入】**

输入只有一行，共有三个参数，其中第 1、2 个参数为整数，第 3 个参数为操作符( $+$ 、 $-$ 、 $*$ 、 $/$ )。

**【输出】**

输出只有一行，一个整数，为运算结果。然而：

- 1.如果出现除数为 0 的情况，则输出：Divided by zero!
- 2.如果出现无效的操作符(即不为  $+$ 、 $-$ 、 $*$ 、 $/$  之一)，则输出：Invalid operator!

## 【输入样例】

1 2 +

## 【输出样例】

3

## 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    char ch;
    cin>>a>>b>>ch;
    if((ch=='+')||(ch=='-')||(ch=='*')||(ch=='/'))
    {
        switch (ch)
        {
            case '+':
                cout<<a+b<<endl;
                break;
            case '-':
                cout<<a-b<<endl;
                break;
            case '*':
                cout<<a*b<<endl;
                break;
            case '/':
                {
                    if(b==0) cout<<"Divided by zero!"<<endl;
                    else cout<<a/b<<endl;
                }
            }
        }
    }
    else cout<<"Invalid operator!"<<endl;
    return 0;
}
```

点赞 2

### 1.1.1.3.2.10 求一元二次方程（信息学奥赛一本通-T1058）

#### 求一元二次方程（信息学奥赛一本通-T1058）

原创 Alex\_McAvoy 发布于 2018-02-06 14:52:27 阅读数 12034 收藏  
展开

### 【题目描述】

求一元二次方程  $ax^2+bx+c=0$  的根，其中  $a$  不等于 0。结果要求精确到小数点后 5 位。

### 【输入】

输入一行，包含三个浮点数  $a, b, c$ （它们之间以一个空格分开），分别表示方程  $ax^2+bx+c=0$  的系数。

### 【输出】

输出一行，表示方程的解。

若两个实根相等，则输出形式为：“ $x_1=x_2=\dots x_1=x_2=\dots$ ”；

若两个实根不等，在满足根小者在前的原则，则输出形式为：“ $x_1=\dots; x_2=\dots x_1=\dots; x_2=\dots$ ”；

若无实根输出“No answer!”。

所有输出部分要求精确到小数点后 5 位，数字、符号之间没有空格。

### 【输入样例】

-15.97 19.69 12.02

### 【输出样例】

$x_1=-0.44781; x_2=1.68075$

### 【源程序】

```
#include <iostream>
```

```
#include <cstdio>
```

```
#include <cmath>
```

```
#define precision_1 1e-12//自己定一精度，用于判断浮点数大小
```

```
#define precision_2 1e-6//自己定一精度，用于判断浮点数大小
```

```
/*
```

双精度浮点数表示法：1 bit 符号位，11bit 指数位(用阶码表示)，52bit 小数部分(尾数)。

因此一个规格化的单精度浮点数  $x$  的真值为  $x=(-1)^S(1.M)(2^{(E-127)})$

显然， $x$  永远也不可能为绝对 0。

针对上面的描述，当阶码  $E$  为全 0 且尾数  $M$  也全 0 时，可以认为表示的真值  $x$  为计算机中的绝对 0 值，

再结合符号位  $S$ ，有正 0 和负 0 之分；

```
*/
```

```
/*
```

$x_1$  和  $x_2$  是两浮点数，precision 是自设的精度。

可以用  $\text{fabs}(x_1-x_2)\leq\text{precision}$  来判断  $x_1$  和  $x_2$  是否相等。



如果要求更高的精度，把 `precision` 定得更小即可

```

*/
using namespace std;
int main()
{
    double a,b,c,x1,x2,delta;
    cin>>a>>b>>c;
    delta=b*b-4*a*c;//判别式
    if(delta<0&&fabs(delta)>precision_1)//当判别式小于给定的精度范围内，即
    delta<0 时，无实根
        printf("No answer!\n");
    else if(fabs(delta)<precision_1)//当判别式小于给定的精度但>0，此时 delta≈0
    {
        x1=-b/(2*a);
        if(fabs(x1)<precision_2) //超出所给的最小范围，此时，视为 delta=0，直接输出
        0

        printf("x1=x2=%.5lf\n",0);
        else//未超出所给范围时，结果仍在小数点后 5 位有效数字以内，输出结果
            printf("x1=x2=%.5lf\n",x1);
    }
    else//delta>0 时
    {
        x1=(-b+sqrt(delta))/(2*a);
        x2=(-b-sqrt(delta))/(2*a);
        if(fabs(x1)<precision_2)
            x1=fabs(x1);//小于所给的最高精度，编译系统视为-0，求绝对值
        if(fabs(x2)<precision_2)
            x2=fabs(x2);//小于所给的最高精度，编译系统视为-0，求绝对值
        if(x1<x2)//x1、x2 中小的数在前输出
            printf("x1=%.5lf;x2=%.5lf",x1,x2);
        else
            printf("x1=%.5lf;x2=%.5lf",x2,x1);
    }
    return 0;
}

```

## 1.1.1.4 第四章 循环结构的程序设计

### 1.1.1.4.1 第一节 for 语句

#### 1.1.1.4.1.1 求平均年龄（信息学奥赛一本通-T1059）

## 求平均年龄（信息学奥赛一本通-T1059）

原创 Alex\_McAvoy 发布于 2018-02-10 01:49:05 阅读数 10150 收藏  
展开

### 【题目描述】

班上有学生若干名，给出每名学生的年龄（整数），求班上所有学生的平均年龄，保留到小数点后两位。

### 【输入】

第一行有一个整数  $n$  ( $1 \leq n \leq 100$ )，表示学生的人数。其后  $n$  行每行有 1 个整数，表示每个学生的年龄，取值为 15 到 25。

### 【输出】

输出一行，该行包含一个浮点数，为要求的平均年龄，保留到小数点后两位。

### 【输入样例】

```
2
18
17
```

### 【输出样例】

```
17.50
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int n,age[101];
    double avrrage,sum=0;
    int i;

    cin>>n;//人数
    for(i=0; i<n; i++)
    {
        cin>>age[i];//年龄
        sum+=age[i];//年龄和
    }
    avrrage=sum/n;//平均值
```

```
printf("%.2lf\n",avrrage);
return 0;
}
```

#### 1.1.1.4.1.2 均值（信息学奥赛一本通-T1060）

## 均值（信息学奥赛一本通-T1060）

原创 Alex\_McAvoy 发布于 2018-02-10 01:51:09 阅读数 7081 收藏  
展开

### 【题目描述】

给出一组样本数据，包含  $n$  个浮点数，计算其均值，精确到小数点后 4 位。

### 【输入】

输入有两行，第一行包含一个整数  $n$  ( $n$  小于 100)，代表样本容量；第二行包含  $n$  个绝对值不超过 1000 的浮点数，代表各个样本数据。

### 【输出】

输出一行，包含一个浮点数，表示均值，精确到小数点后 4 位。

### 【输入样例】

```
2
1.0 3.0
```

### 【输出样例】

```
2.0000
```

### 【源程序】

```
#include <iostream>
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    int num,i;
    double n,arr,sum=0;
    cin>>num;
    for(i=1; i<=num; i++)
    {
        cin>>n;//输入数据
        sum=sum+n;//数据和
    }
}
```

```

arr=sum*1.0/num;//均值
cout<<setprecision(4)<<setiosflags(ios::fixed);
cout<<arr<<endl;
return 0;
}

```

### 1.1.1.4.1.3 求整数的和与均值（信息学奥赛一本通-T1061）

#### 求整数的和与均值（信息学奥赛一本通-T1061）

原创 Alex\_McAvoy 发布于 2018-02-12 18:44:04 阅读数 5764 收藏  
展开

##### 【题目描述】

读入  $n(1 \leq n \leq 10000)$  个整数，求它们的和与均值。

##### 【输入】

输入第一行是一个整数  $n$ ，表示有  $n$  个整数。第  $2 \sim n+1$  行每行包含 1 个整数。每个整数的绝对值均不超过 10000。

##### 【输出】

输出一行，先输出和，再输出平均值(保留到小数点后 5 位)，两个数间用单个空格分隔。

##### 【输入样例】

```

4
344
222
343
222

```

##### 【输出样例】

```

1131 282.75000

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int n,a[10001];
    double average;
    int i;

```

```

cin>>n;//输入数据数
a[0]=0;//用于存储数据和
for(i=1; i<=n; i++)
{
    cin>>a[i];//输入数据
    a[0]+=a[i];//求和
}

average=1.0*a[0]/n;//求平均值
printf("%d %.5lf\n",a[0],average);
return 0;
}

```

#### 1.1.1.4.1.4 最高的分数（信息学奥赛一本通-T1062）

##### 最高的分数（信息学奥赛一本通-T1062）

原创 Alex\_McAvoy 发布于 2018-02-12 18:50:01 阅读数 6065 收藏  
展开

##### 【题目描述】

孙老师讲授的《计算概论》这门课期中考试刚刚结束，他想知道考试中取得的最高分数。因为人数比较多，他觉得这件事情交给计算机来做比较方便。你能帮孙老师解决这个问题吗？

##### 【输入】

输入两行，第一行为整数  $n$  ( $1 \leq n < 100$ )，表示参加这次考试的人数。第二行是这  $n$  个学生的成绩，相邻两个数之间用单个空格隔开。所有成绩均为 0 到 100 之间的整数。

##### 【输出】

输出一个整数，即最高的成绩。

##### 【输入样例】

```

5
85 78 90 99 60

```

##### 【输出样例】

```

99

```

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{

```

```

int n,a[101];
int i;

cin>>n;//输入人数
a[0]=0;
for(i=1; i<=n; i++)
{
    cin>>a[i];//输入成绩
    if(a[i]>a[0])//进行比较
        a[0]=a[i];
}

cout<<a[0]<<endl;
return 0;
}

```

#### 1.1.1.4.1.5 最大跨度值（信息学奥赛一本通-T1063）

##### 最大跨度值（信息学奥赛一本通-T1063）

原创 Alex\_McAvoy 发布于 2018-02-12 18:56:22 阅读数 6541 收藏  
展开

##### 【题目描述】

给定一个长度为  $n$  的非负整数序列，请计算序列的最大跨度值(最大跨度值 = 最大值减去最小值)

##### 【输入】

一共 2 行，第一行为序列的个数  $n$  ( $1 \leq n \leq 1000$ )，第二行为序列的  $n$  个不超过 1000 的非负整数，整数之间以一个空格分隔。

##### 【输出】

输出一行，表示序列的最大跨度值。

##### 【输入样例】

```

6
3 0 8 7 5 9

```

##### 【输出样例】

```

9

```

##### 【源程序】

```

#include<iostream>
using namespace std;

```

```

int main()
{
    int n,a[1001];
    int min=9999,max=0;//初始化
    int i;

    cin>>n;
    for(i=1; i<=n; i++)
    {
        cin>>a[i];
        if(a[i]>max)//最大值
            max=a[i];
        if(a[i]<min)//最小值
            min=a[i];
    }

    cout<<max-min<<endl;//跨度值
    return 0;
}

```

#### 1.1.1.4.1.6 奥运奖牌计数（信息学奥赛一本通-T1064）

## 奥运奖牌计数（信息学奥赛一本通-T1064）

原创 Alex\_McAvoy 发布于 2018-02-12 19:05:36 阅读数 5972 收藏  
展开

### 【题目描述】

2008 年北京奥运会，A 国的运动员参与了  $n$  天的决赛项目( $1 \leq n \leq 17$ )。现在要统计一下 A 国所获得的金、银、铜牌数目及总奖牌数。输入第 1 行是 A 国参与决赛项目的天数  $n$ ，其后  $n$  行，每一行是该国某一天获得的金、银、铜牌数目。输出 4 个整数，为 A 国所获得的金、银、铜牌总数及总奖牌数。

### 【输入】

输入  $n+1$  行，第 1 行是 A 国参与决赛项目的天数  $n$ ，其后  $n$  行，每一行是该国某一天获得的金、银、铜牌数目，以一个空格分开。

### 【输出】

输出 1 行，包括 4 个整数，为 A 国所获得的金、银、铜牌总数及总奖牌数，以一个空格分开。

**【输入样例】**

```
3
1 0 3
3 1 0
0 3 0
```

**【输出样例】**

```
4 4 3 11
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int sum,a[20][3]={0};//初始化
    int i;

    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>a[i][0]>>a[i][1]>>a[i][2];//每天的奖牌数
        a[0][0]+=a[i][0];//金牌数
        a[0][1]+=a[i][1];//银牌数
        a[0][2]+=a[i][2];//铜牌数
    }

    sum=a[0][0]+a[0][1]+a[0][2];//总数
    cout<<a[0][0]<<" "<<a[0][1]<<" "<<a[0][2]<<" "<<sum<<endl;
    return 0;
}
```

**1.1.1.4.1.7 奇数求和（信息学奥赛一本通-T1065）****奇数求和（信息学奥赛一本通-T1065）**

原创 Alex\_McAvoy 发布于 2018-02-12 19:11:12 阅读数 5864 收藏  
展开

**【题目描述】**

计算非负整数  $m$  到  $n$  (包括  $m$  和  $n$ ) 之间的所有奇数的和, 其中,  $m$  不大于  $n$ , 且  $n$  不大于 300。例如  $m=3, n=12$ , 其和则为:  $3+5+7+9+11=35$ 。



**【输入】**

两个数  $m$  和  $n$ ，两个数以一个空格分开，其中  $0 \leq m \leq n \leq 300$ 。

**【输出】**

输出一行，包含一个整数，表示  $m$  到  $n$ （包括  $m$  和  $n$ ）之间的所有奇数的和

**【输入样例】**

7 15

**【输出样例】**

55

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    int sum=0;//初始化
    int i;

    cin>>a>>b;
    if(a%2==0) a++;//保证从奇数开始加
    for(i=a; i<=b; i+=2) //计算奇数和
        sum+=i;

    cout<<sum<<endl;
    return 0;
}
```

点赞

### 1.1.1.4.1.8 满足条件的数累加（信息学奥赛一本通-T1066）

#### 满足条件的数累加（信息学奥赛一本通-T1066）

原创 Alex\_McAvoy 发布于 2018-02-13 21:32:10 阅读数 5069 收藏  
展开

**【题目描述】**

将正整数  $m$  和  $n$  之间(包括  $m$  和  $n$ )能被 17 整除的数累加，其中  $0 < m < n < 1000$ 。

**【输入】**

一行，包含两个整数  $m$  和  $n$ ，其间，以一个空格间隔。

**【输出】**

输出一行，包行一个整数，表示累加的结果。

**【输入样例】**

50 85

**【输出样例】**

204

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int m,n,sum=0;
    int i;

    cin>>m>>n;//输入 m、n
    for(i=m; i<=n; i++) //在区间内循环
        if(i%17==0)//满足条件
```

---

版权声明：本文为 CSDN 博主「Alex\_McAvoy」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011815404/article/details/79323350> to insert a line break. Press Enter to add a paragraph return.)

### 1.1.1.4.1.9 整数的个数（信息学奥赛一本通-T1067）

#### 整数的个数（信息学奥赛一本通-T1067）

原创 Alex\_McAvoy 发布于 2018-02-13 21:46:28 阅读数 4979 收藏  
展开

**【题目描述】**

给定  $k(1 < k < 100)$  个正整数，其中每个数都是大于等于 1，小于等于 10 的数。写程序计算给定的  $k$  个正整数中，1，5 和 10 出现的次数。

**【输入】**

输入有两行：第一行包含一个正整数  $k$ ，第二行包含  $k$  个正整数，每两个正整数用一个空格分开。

**【输出】**

输出有三行，第一行为 1 出现的次数，，第二行为 5 出现的次数，第三行为 10 出现的次数。

**【输入样例】**

```
5
1 5 8 10 5
```

**【输出样例】**

```
1
2
1
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,number;
    int sum_1=0,sum_5=0,sum_10=0;
    int i;

    cin>>n;//输入数字个数
    for(i=1; i<=n; i++)
    {
        cin>>number;//输入各个数字
        if(number==1)
            sum_1++;//统计等于 1 的个数
        if(number==5)
            sum_5++;//统计等于 5 的个数
        if(number==10)
            sum_10++;//统计等于 10 的个数
    }
    cout<<sum_1<<endl;
    cout<<sum_5<<endl;
    cout<<sum_10<<endl;
    return 0;
}
```

点赞 1

### 1.1.1.4.1.10 与指定数字相同的数的个数(信息学奥赛一本通-T1068)

#### 与指定数字相同的数的个数（信息学奥赛一本通-T1068）

原创 Alex\_McAvoy 发布于 2018-02-13 22:01:53 阅读数 4623 收藏  
展开

##### 【题目描述】

输出一个整数序列中与指定数字相同的数的个数。输入包含 2 行：第 1 行为  $N$  和  $m$ ，表示整数序列的长度( $N \leq 100$ )和指定的数字；第 2 行为  $N$  个整数，整数之间以一个空格分开。输出为  $N$  个数中与  $m$  相同的数的个数。

##### 【输入】

第 1 行为  $N$  和  $m$ ，表示整数序列的长度( $N \leq 100$ )和指定的数字，中间用一个空格分开；

第 2 行为  $N$  个整数，整数之间以一个空格分开。

##### 【输出】

输出为  $N$  个数中与  $m$  相同的数的个数。

##### 【输入样例】

```
3 2
2 3 2
```

##### 【输出样例】

```
2
```

##### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n,number,N;
    int sum=0;
    int i;

    cin>>n>>number;//输入长度、指定数字
    for(i=1; i<=n; i++)
    {
        cin>>N;//输入各个数字
        if(N==number) sum++;//统计相同的个数
    }
}
```

```

    cout<<sum<<endl;
    return 0;
}

```

### 1.1.1.4.1.11 乘方计算（信息学奥赛一本通-T1069）

#### 乘方计算（信息学奥赛一本通-T1069）

原创 Alex\_McAvoy 发布于 2018-02-13 22:05:12 阅读数 5318 收藏

展开

##### 【题目描述】

给出一个整数  $a$  和一个正整数  $n$  ( $-1000000 \leq a \leq 1000000$ ,  $1 \leq n \leq 10000$ ), 求乘方  $a^n$ , 即乘方结果。最终结果的绝对值不超过 1000000。

##### 【输入】

一行, 包含两个整数  $a$  和  $n$ 。  $-1000000 \leq a \leq 1000000$ ,  $1 \leq n \leq 10000$ 。

##### 【输出】

一个整数, 即乘方结果。题目保证最终结果的绝对值不超过 1000000。

##### 【输入样例】

```

2
3

```

##### 【输出样例】

```

8

```

##### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int a,n;
    int result;

    cin>>a>>n;
    result=pow(a,n);//调用函数进行计算
    cout<<result<<endl;
    return 0;
}

```

### 1.1.1.4.1.12 人口增长（信息学奥赛一本通-T1070）

#### 人口增长（信息学奥赛一本通-T1070）

原创 Alex\_McAvoy 发布于 2018-02-13 22:15:21 阅读数 5966 收藏  
展开

##### 【题目描述】

我国现有  $x$  亿人口，按照每年 0.1% 的增长速度， $n$  年后将有多少人？保留小数点后四位。

##### 【输入】

一行，包含两个整数  $x$  和  $n$ ，分别是人口基数和年数，以单个空格分隔。

##### 【输出】

输出最后的人口数，以亿为单位，保留到小数点后四位。 $1 \leq x \leq 100, 1 \leq n \leq 100$ 。

##### 【输入样例】

13 10

##### 【输出样例】

13.1306

##### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    double x,n;
    int i;

    cin>>x>>n;
    for(i=1; i<=n; i++) //按年循环
        x=x*1.001;//计算

    printf("%.4lf\n",x);
    return 0;
}
```

### 1.1.1.4.1.13 菲波那契数（信息学奥赛一本通-T1071）

#### 菲波那契数（信息学奥赛一本通-T1071）

原创 Alex\_McAvoy 发布于 2018-02-13 22:23:03 阅读数 6382 收藏  
展开

#### 【题目描述】

菲波那契数列是指这样的数列: 数列的第一个和第二个数都为 1, 接下来每个数都等于前面 2 个数之和。给出一个正整数  $k$ , 要求菲波那契数列中第  $k$  个数是多少。

#### 【输入】

输入一行, 包含一个正整数  $k$ 。( $1 \leq k \leq 46$ )

#### 【输出】

输出一行, 包含一个正整数, 表示菲波那契数列中第  $k$  个数的大小。

#### 【输入样例】

19

#### 【输出样例】

4181

#### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int k;
    int a=1,b=1,temp;//a、b 为第 1、2 个数, temp 用于存储第 i 个数
    int i;

    cin>>k;

    for(i=3; i<=k; i++)
    {
        temp=a+b;//每个数都为前两数之和
        a=b;
        b=temp;
    }
    cout<<b<<endl;
    return 0;
}
```

### 1.1.1.4.1.14 鸡尾酒疗法（信息学奥赛一本通-T1072）

#### 鸡尾酒疗法（信息学奥赛一本通-T1072）

原创 Alex\_McAvoy 发布于 2018-02-13 22:36:45 阅读数 10543 收藏  
展开

### 【题目描述】

鸡尾酒疗法，指“高效抗逆转录病毒治疗”。人们在鸡尾酒疗法的基础上又提出了很多种改进的疗法。为了验证这些治疗方法是否在疗效上比鸡尾酒疗法更好，可用通过临床对照实验的方式进行。假设鸡尾酒疗法的有效率为  $x$ ，新疗法的有效率为  $y$ ，如果  $y-x$  大于 5%，则效果更好，如果  $x-y$  大于 5%，则效果更差，否则称为效果差不多。下面给出  $n$  组临床对照实验，其中第一组采用鸡尾酒疗法，其他  $n-1$  组为各种不同的改进疗法。请写程序判定各种改进疗法效果如何。

### 【输入】

第一行为整数  $n$  ( $1 < n \leq 20$ )；其余  $n$  行每行两个整数，第一个整数是临床实验的总病例数(小于等于 10000)，第二个疗效有效的病例数。这  $n$  行数据中，第一行为鸡尾酒疗法的数据，其余各行为各种改进疗法的数据。

### 【输出】

有  $n-1$  行输出，分别表示对应改进疗法的效果：如果效果更好，输出 **better**；如果效果更差，输出 **worse**；否则输出 **same**。

### 【输入样例】

```
5
125 99
112 89
145 99
99 97
123 98
```

### 【输出样例】

```
same
worse
better
same
```

### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n,a[30][2];
    double x,y;
    int i;

    cin>>n;//输入组数
```



```

for(i=0; i<n; i++) //输入数据
    cin>>a[i][0]>>a[i][1];
x=1.0*a[0][1]/a[0][0]; //计算第一种数据
for(i=1; i<n; i++) //进行比较
{
    y=1.0*a[i][1]/a[i][0]; //每组数据
    if(y-x>0.05) cout<<"better"<<endl;
    else if(x-y>0.05) cout<<"worse"<<endl;
    else cout<<"same"<<endl;
}
return 0;
}

```

#### 1.1.1.4.1.15 救援（信息学奥赛一本通-T1073）

## 救援（信息学奥赛一本通-T1073）

原创 Alex\_McAvoy 发布于 2018-02-13 22:48:42 阅读数 33645 收藏  
展开

### 【题目描述】

救生船从大本营出发，营救若干屋顶上的人回到大本营，屋顶数目以及每个屋顶的坐标和人数都将由输入决定，求出所有人都到达大本营并登陆所用的时间。

在直角坐标系的原点是大本营，救生船每次从大本营出发，救了人之后将人送回大本营。坐标系中的点代表屋顶，每个屋顶由其位置坐标和其上的人数表示。救生船每次从大本营出发，以速度 50 米/分钟驶向下一个屋顶，达到一个屋顶后，救下其上的所有人，每人上船 1 分钟，船原路返回，达到大本营，每人下船 0.5 分钟。假设原点与任意一个屋顶的连线不穿过其它屋顶。

### 【输入】

第一行，一个整数，表示屋顶数  $n$ 。接下来依次有  $n$  行输入，每一行上包含两个表示屋顶相对于大本营的平面坐标位置的实数(单位是米)、一个表示人数的整数。

### 【输出】

救援需要的总时间，精确到分钟(向上取整)。

### 【输入样例】

```

1
30 40 3

```

### 【输出样例】

```

7

```

## 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    double n,people,x,y;
    double l,time=0;
    int i;

    cin>>n;//输入屋顶数
    for(i=1; i<=n; i++) //每一个屋顶
    {
        cin>>x>>y;
        cin>>people;
        l=sqrt(1.0*x*x+1.0*y*y);
        time+=2*l/50+people*1.5;//计算时间
    }
    cout<<ceil(time)<<endl;//向上取整
    return 0;
}

```

## 1.1.1.4.1.16 津津的储蓄计划（信息学奥赛一本通-T1074）

## 津津的储蓄计划（信息学奥赛一本通-T1074）

原创 Alex\_McAvoy 发布于 2018-02-13 22:53:06 阅读数 5263 收藏  
展开

## 【题目描述】

津津的零花钱一直都是自己管理。每个月的月初妈妈给津津 300 元钱，津津会预算这个月的花销，并且总能做到实际花销和预算的相同。

为了让津津学习如何储蓄，妈妈提出，津津可以随时把整百的钱存在她那里，到了年末她会加上 20% 还给津津。因此津津制定了一个储蓄计划：每个月的月初，在得到妈妈给的零花钱后，如果她预计到这个月的月末手中还会有多于 100 元或恰好 100 元，她就会把整百的钱存在妈妈那里，剩余的钱留在自己手中。

例如 11 月初津津手中还有 83 元，妈妈给了津津 300 元。津津预计 11 月的花销是 180 元，那么她就会在妈妈那里存 200 元，自己留下 183 元。到了 11 月月末，津津手中会剩下 3 元钱。

现在请你根据 2004 年 1 月到 12 月每个月津津的预算，判断会不会出现这种情况。如果不会，计算到 2004 年年末，妈妈将津津平常存的钱加上 20% 还给津津之后，津津手中会有多少钱。

**【输入】**

包括 12 行数据，每行包含一个小于 350 的非负整数，分别表示 1 月到 12 月津津的预算。

**【输出】**

只包含一个整数。如果储蓄计划实施过程中出现某个月钱不够用的情况，输出 -X，X 表示出现这种情况的第一个月；否则输出到 2004 年年末津津手中会有多少钱。

**【输入样例】**

```
290
230
280
200
300
170
340
50
90
80
200
60
```

**【输出样例】**

```
-7
```

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int plan, month, overplus=0, flag=1, total=0;
    int i, m;

    for(i=0; i<12; i++) //总共 12 个月
    {
        cin>>plan;
        overplus=overplus+300-plan; //剩余的钱数

        if(overplus<0&&flag) //剩余钱数小于零且此前无超出预算的情况
        {
            month=i+1;
            flag=0;
        }
    }
}
```

```

    }
    m=overplus/100;//存入的钱的系数
    overplus=overplus-m*100;//剩余的钱数
    total=total+m*100;//存入的钱
}

if(flag==0) cout<<-1*month<<endl;//flag=0, 说明有超出预算的月份
else cout<<overplus+(total*6)/5<<endl;//输出总钱数

return 0;
}

```

### 1.1.1.4.1.17 药房管理（信息学奥赛一本通-T1075）

#### 药房管理（信息学奥赛一本通-T1075）

原创 Alex\_McAvoy 发布于 2018-02-13 23:26:01 阅读数 5992 收藏  
展开

##### 【题目描述】

随着信息技术的蓬勃发展，医疗信息化已经成为医院建设中必不可少的一部分。计算机可以很好地辅助医院管理医生信息、病人信息、药品信息等海量数据，使工作人员能够从这些机械的工作中解放出来，将更多精力投入真正的医疗过程中，从而极大地提高了医院整体的工作效率。

对药品的管理是其中的一项重要内容。现在药房的管理员希望使用计算机来帮助他管理。假设对于任意一种药品，每天开始工作时的库存总量已知，并且一天之内不会通过进货的方式增加。每天会有很多病人前来取药，每个病人希望取走不同数量的药品。如果病人需要的数量超过了当时的库存量，药房会拒绝该病人的请求。管理员希望知道每天会有多少病人没有取上药。

##### 【输入】

共 3 行,第一行是每天开始时的药品总量  $m$ 。

第二行是这一天取药的人数  $n(0 < n \leq 100)$ 。

第三行共有  $n$  个数，分别记录了每个病人希望取走的药品数量(按照时间先后的顺序)。

##### 【输出】

只有 1 行，为这一天没有取上药品的人数。

##### 【输入样例】

```

30
6
10 5 20 6 7 8

```

**【输出样例】**

2

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int m,n,a[101];
    int sum=0;
    int i;

    cin>>m>>n;//药品总量、取药人数
    for(i=1; i<=n; i++) //每个病人取药数量
    {
        cin>>a[i];
        if(m>=a[i]) m-=a[i];//判断第 i 个人的取药情况
        else sum++;
    }
    cout<<sum<<endl;
    return 0;
}
```

**1.1.1.4.1.18 正常血压（信息学奥赛一本通-T1076）****正常血压（信息学奥赛一本通-T1076）**

原创 Alex\_McAvoy 发布于 2018-02-13 23:26:44 阅读数 5078 收藏  
展开

**【题目描述】**

监护室每小时测量一次病人的血压，若收缩压在 90-140 之间并且舒张压在 60-90 之间(包含端点值)则称之为正常，现给出某病人若干次测量的血压值，计算病人保持正常血压的最长小时数。

**【输入】**

第一行为一个正整数  $n(n<100)$ ，其后有  $n$  行，每行 2 个正整数，分别为一次测量的收缩压和舒张压。

**【输出】**

输出仅一行，血压连续正常的最长小时数。

**【输入样例】**

```
4
100 80
```

90 50  
120 60  
140 90

【输出样例】

2

【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n,a[101][2];
    int max=0,sum=0;
    int i;

    cin>>n;//测量 n 次
    for(i=1; i<=n; i++) //每个小时的情况
    {
        cin>>a[i][0]>>a[i][1];//收缩压、舒张压
        if(a[i][0]>=90&&a[i][0]<=140&&a[i][1]>=60&&a[i][1]<=90)//判断是否正常
        {
            sum++;
            if(sum>max)//如果大于目前连续正常最长小时数
                max=sum;//替换 max 的值
        }
        else sum=0;//若不这个小时内不正常，清零 sum
    }
    cout<<max<<endl;
    return 0;
}
```

#### 1.1.1.4.1.19 统计满足条件的 4 位数（信息学奥赛一本通-T1077）

##### 统计满足条件的 4 位数（信息学奥赛一本通-T1077）

原创 Alex\_McAvoy 发布于 2018-02-14 11:56:43 阅读数 3757 收藏  
展开

**【题目描述】**

给定若干个四位数，求出其中满足以下条件的数的个数：个位数上的数字减去千位数上的数字，再减去百位数上的数字，再减去十位数上的数字的结果大于零。

**【输入】**

输入为两行，第一行为四位数的个数  $n$ ，第二行为  $n$  个的四位数。 $(n \leq 100)$

**【输出】**

输出为一行，包含一个整数，表示满足条件的四位数的个数。

**【输入样例】**

```
5
1234 1349 6119 2123 5017
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,m;
    int a,b,c,d,sum=0;
    int i;

    cin>>n;//数字个数
    for(i=1; i<=n; i++)
    {
        cin>>m;//每个四位数
        a=m/1000;//千位
        b=m/100%10;//百位
        c=m/10%10;//十位
        d=m%10;//个位
        if(d-a-b-c>0) sum++;//满足条件，累加
    }
    cout<<sum<<endl;
    return 0;
}
```

### 1.1.1.4.1.20 求分数序列和（信息学奥赛一本通-T1078）

#### 求分数序列和（信息学奥赛一本通-T1078）

原创 Alex\_McAvoy 发布于 2018-02-14 12:20:51 阅读数 4539 收藏  
展开

### 【题目描述】

有一个分数序列  $q_1/p_1, q_2/p_2, q_3/p_3, q_4/p_4, q_5/p_5, \dots$ , 其中  $q_{i+1}=q_i+p_i, p_{i+1}=q_i$ ,  $p_1=1, q_1=2$ 。比如这个序列前 6 项分别是 21,32,53,85,138,211。求这个分数序列的前  $n$  项之和。21,32

### 【输入】

输入有一行，包含一个正整数  $n(n \leq 30)$ 。

### 【输出】

输出有一行，包含一个浮点数，表示分数序列前  $n$  项的和，精确到小数点后 4 位。

### 【输入样例】

2

### 【输出样例】

3.5000

### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int n;
    double a=1,b=1,c,sum=0;
    int i;

    cin>>n;//前 n 项
    for(i=1; i<=n; i++)
    {
        c=a+b;
        a=b;
        b=c;//数列规律
        sum+=1.0*b/a;//求和
    }
    printf("%.4lf\n",sum);
    return 0;
}
```



### 1.1.1.4.1.21 计算分数加减表达式的值(信息学奥赛一本通-T1079)

#### 计算分数加减表达式的值（信息学奥赛一本通-T1079）

原创 Alex\_McAvoy 发布于 2018-02-14 12:27:41 阅读数 4529 收藏  
展开

##### 【题目描述】

编写程序，输入  $n$  的值，求  $1/1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 - 1/8 + \dots + (-1)^{(n+1)} \cdot 1/n$  的值。

##### 【输入】

输入一个正整数  $n$ 。  $1 \leq n \leq 1000$ 。

##### 【输出】

输出一个实数，为表达式的值，保留到小数点后四位。

##### 【输入样例】

2

##### 【输出样例】

0.5000

##### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int n;
    double sum=0;
    int i;

    cin>>n;//前 n 项
    for(i=1; i<=n; i++) //数列规律
    {
        if(i%2==0) sum-=1.0/i;
        else sum+=1.0/i;
    }
    printf("%.4lf\n",sum);
    return 0;
}
```

### 1.1.1.4.1.22 余数相同问题（信息学奥赛一本通-T1080）

#### 余数相同问题（信息学奥赛一本通-T1080）

原创 Alex\_McAvoy 发布于 2018-02-14 12:32:58 阅读数 4633 收藏  
展开

##### 【题目描述】

已知三个正整数  $a$ ,  $b$ ,  $c$ 。现有一个大于 1 的整数  $x$ ，将其作为除数分别除  $a$ ,  $b$ ,  $c$ ，得到的余数相同。

请问满足上述条件的  $x$  的最小值是多少？数据保证  $x$  有解。

##### 【输入】

一行，三个不大于 1000000 的正整数  $a$ ,  $b$ ,  $c$ ，两个整数之间用一个空格隔开。

##### 【输出】

一个整数，即满足条件的  $x$  的最小值。

##### 【输入样例】

300 262 205

##### 【输出样例】

19

##### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    int i;

    cin>>a>>b>>c;
    for(i=2; i<=a; i++) //从 2 开始寻找
    {
        if(a%i==b%i&&b%i==c%i)//余数相同
        {
            cout<<i<<endl;//输出
            break;//终止循环，退出
        }
    }
    return 0;
}
```

#### 1.1.1.4.1.23 分苹果（信息学奥赛一本通-T1081）

##### 分苹果（信息学奥赛一本通-T1081）

原创 Alex\_McAvoy 发布于 2018-02-14 12:36:53 阅读数 4163 收藏  
展开

###### 【题目描述】

把一堆苹果分给  $n$  个小朋友，要使每个人都能拿到苹果，而且每个人拿到的苹果数都不同的话，这堆苹果至少应该有多少个？

###### 【输入】

一个不大于 1000 的正整数  $n$ ，代表小朋友人数。

###### 【输出】

一个整数，表示满足条件的最少苹果个数。

###### 【输入样例】

8

###### 【输出样例】

36

###### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n,sum=0;
    int i;

    cin>>n;//人数
    for(i=1; i<=n; i++) //第一个从 1 开始，其后每人比前一人多一个
        sum+=i;//累计
    cout<<sum<<endl;
    return 0;
}
```

#### 1.1.1.4.1.24 求小数的某一位（信息学奥赛一本通-T1082）

##### 求小数的某一位（信息学奥赛一本通-T1082）

**【题目描述】**

分数  $a/b$  化为小数后，小数点后第  $n$  位的数字是多少？

**【输入】**

三个正整数  $a, b, n$ ，相邻两个数之间用单个空格隔开。 $0 < a < b < 100, 1 \leq n \leq 10000$ 。

**【输出】**

一个数字

**【输入样例】**

1 2 1

**【输出样例】**

5

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,n;
    int temp;
    int i;

    cin>>a>>b>>n;
    for(i=1; i<=n; i++)
    {
        a*=10;
        temp=a/b;//temp 模拟存储第 i 为小数
        a%=b;
    }
    cout<<temp<<endl;
    return 0;
}
```

### 1.1.1.4.1.25 计算星期几（信息学奥赛一本通-T1083）

#### 计算星期几（信息学奥赛一本通-T1083）

原创 Alex\_McAvoy 发布于 2018-02-14 12:48:41 阅读数 7270 收藏  
展开

**【题目描述】**

假设今天是星期日，那么  $a^b$  天之后是星期几？

**【输入】**

两个正整数  $a, b$ ，中间用单个空格隔开。 $0 < a \leq 100, 0 < b \leq 10000$ 。

**【输出】**

一个字符串，代表过  $a^b$  天之后是星期几。

其中，Monday 是星期一，Tuesday 是星期二，Wednesday 是星期三，Thursday 是星期四，Friday 是星期五，Saturday 是星期六，Sunday 是星期日。

**【输入样例】**

3 2000

**【输出样例】**

Tuesday

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 5000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

#include<iostream>
using namespace std;
int main()
{
    int a,b;
    int n=1;
    int i;
```

```

cin>>a>>b;
for(i=1; i<=b; i++) n=n*a%7; //循环计算 n 天后是周几
switch(n)
{
    case 0:
        cout<<"Sunday"<<endl;
        break;
    case 1:
        cout<<"Monday"<<endl;
        break;
    case 2:
        cout<<"Tuesday"<<endl;
        break;
    case 3:
        cout<<"Wednesday"<<endl;
        break;
    case 4:
        cout<<"Thursday"<<endl;
        break;
    case 5:
        cout<<"Friday"<<endl;
        break;
    case 6:
        cout<<"Saturday"<<endl;
        break;
}
return 0;
}

```

#### 1.1.1.4.1.26 幂的末尾（信息学奥赛一本通-T1084）

##### 幂的末尾（信息学奥赛一本通-T1084）

原创 Alex\_McAvoy 发布于 2018-02-14 12:53:22 阅读数 6075 收藏  
展开

##### 【题目描述】

$a^b$  的末 3 位数是多少？

##### 【输入】

两个正整数  $a$ ,  $b$ 。  $1 \leq a \leq 100$ ,  $1 \leq b \leq 10000$ 。

**【输出】**

从高位到低位输出幂的末三位数字，中间无分隔符。若幂本身不足三位，在前面补零。

**【输入样例】**

7 2011

**【输出样例】**

743

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    int k=1;
    int i;

    cin>>a>>b;
    for(i=1; i<=b; i++)
    {
        k*=a;
        k%=1000;//k 存储  $a^i$  的后三位
    }

    if(k>=100)
        cout<<k<<endl;
    else if(k>=10)
        cout<<"0"<<k<<endl;//处理结果不为 3 位的情况
    else
        cout<<"00"<<k<<endl;
    return 0;
}
```

### 1.1.1.4.2 第二节 while 与 do-while 语句

#### 1.1.1.4.2.1 球弹跳高度的计算（信息学奥赛一本通-T1085）

球弹跳高度的计算（信息学奥赛一本通-T1085）

原创 Alex\_McAvoy 发布于 2018-02-15 00:33:48 阅读数 5712 收藏  
展开

**【题目描述】**

一球从某一高度  $h$  落下(单位米), 每次落地后反跳回原来高度的一半, 再落下。编程计算气球在第 10 次落地时, 共经过多少米? 第 10 次反弹多高?

输出包含两行, 第 1 行: 到球第 10 次落地时, 一共经过的米数。第 2 行: 第 10 次弹跳的高度。

**【输入】**

输入一个整数  $h$ , 表示球的初始高度。

**【输出】**

第 1 行: 到球第 10 次落地时, 一共经过的米数。

第 2 行: 第 10 次弹跳的高度。

注意: 结果可能是实数, 结果用 `double` 类型保存。

提示: 输出时不需要对精度特殊控制, 用 `cout << ANSWER`, 或者 `printf("%g", ANSWER)`即可。

**【输入样例】**

20

**【输出样例】**

59.9219

0.0195312

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    double h,sum;
    int i;

    cin>>h;
    sum=h;
    for(i=1; i<=9; i++)
    {
        h/=2;//每次反弹回原高的一半
        sum+=h*2;//累加反弹与落地的米数
    }
    cout<<sum<<endl;//总米数
```



```

    cout<<h/2<<endl;//最后弹跳的高度
    return 0;
}

```

### 1.1.1.4.2.2 角谷猜想（信息学奥赛一本通-T1086）

#### 角谷猜想（信息学奥赛一本通-T1086）

原创 Alex\_McAvoy 发布于 2018-02-15 00:33:50 阅读数 6135 收藏

展开

##### 【题目描述】

角谷猜想，是指对于任意一个正整数，如果是奇数，则乘 3 加 1，如果是偶数，则除以 2，得到的结果再按照上述规则重复处理，最终总能够得到 1。如，假定初始整数为 5，计算过程分别为 16、8、4、2、1。程序要求输入一个整数，将经过处理得到 1 的过程输出来。

##### 【输入】

一个正整数  $N(N \leq 2,000,000)$ 。

##### 【输出】

从输入整数到 1 的步骤，每一步为一行，每一部中描述计算过程。最后一行输出"End"。如果输入为 1，直接输出"End"。

##### 【输入样例】

5

##### 【输出样例】

5\*3+1=16

16/2=8

8/2=4

4/2=2

2/2=1

End

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    long int n,a;

    cin>>n;
    while(n!=1)

```

```

{
    if(n%2)//n 为奇数
    {
        cout<<n<<"*3+1="<<n*3+1<<endl;
        n=n*3+1;
    }
    else//n 为偶数
    {
        cout<<n<<"/2="<<n/2<<endl;
        n/=2;
    }
}
cout<<"End"<<endl;
return 0;
}

```

### 1.1.1.4.2.3 级数求和（信息学奥赛一本通-T1087）

#### 级数求和（信息学奥赛一本通-T1087）

原创 Alex\_McAvoy 发布于 2018-02-15 00:52:14 阅读数 5467 收藏  
展开

##### 【题目描述】

已知： $S_n=1+1/2+1/3+\dots+1/n$ 。显然对于任意一个整数  $k$ ，当  $n$  足够大的时候， $S_n$  大于  $k$ 。现给出一个整数  $k$  ( $1\leq k\leq 15$ )，要求计算出一个最小的  $n$ ，使得  $S_n>k$ 。

##### 【输入】

一个整数  $k$

##### 【输出】

一个整数  $n$

##### 【输入样例】

1

##### 【输出样例】

2

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{

```

```

int k,n=1;
double sum=1;

cin>>k;
while(sum<=k)
{
    n++;//累加 n 的值
    sum+=1.0/n;//计算 n 项和
}
cout<<n<<endl;
return 0;
}

```

#### 1.1.1.4.2.4 分离整数的各个数（信息学奥赛一本通-T1088）

##### 分离整数的各个数（信息学奥赛一本通-T1088）

原创 Alex\_McAvoy 发布于 2018-02-15 17:50:12 阅读数 5602 收藏  
展开

##### 【题目描述】

给定一个整数  $n$  ( $1 \leq n \leq 1000000000$ )，要求从个位开始分离出它的每一位数字。从个位开始按照从低位到高位顺序依次输出每一位数字

##### 【输入】

输入一个整数，整数在 1 到 1000000000 之间

##### 【输出】

从个位开始按照从低位到高位顺序依次输出每一位数字。数字之间以一个空格分开

##### 【输入样例】

123

##### 【输出样例】

3 2 1

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    long int n;

    cin>>n;

```

```

while(n)//相当于 n!=0
{
    cout<<n%10<<" ";//分离当前个位
    n/=10;//整除 10，相当于删除个位
}
cout<<endl;
return 0;
}

```

点赞

### 1.1.1.4.2.5 数字反转（信息学奥赛一本通-T1089）

#### 数字反转（信息学奥赛一本通-T1089）

原创 Alex\_McAvoy 发布于 2018-02-15 17:51:16 阅读数 5770 收藏  
展开

##### 【题目描述】

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，例如输入-380，反转后得到的新数为-83。

##### 【输入】

输入共 1 行，一个整数 N。  $-1,000,000,000 \leq N \leq 1,000,000,000$ 。

##### 【输出】

输出共 1 行，一个整数，表示反转后的新数。

##### 【输入样例】

123

##### 【输出样例】

321

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    int n;
    int sum=0;

    cin>>n;

```

```

if(n<0)//若 n 为负数
{
    cout<<"-";//输出负号
    n=-n;//取正
}
while(n)//相当于 n!=0
{
    sum=sum*10+n%10;//原数个位相当于新数最高位
    n/=10;//整除 10，相当于删除个位
}
cout<<sum<<endl;
return 0;
}

```

### 1.1.1.4.2.6 含 k 个 3 的数（信息学奥赛一本通-T1090）

#### 含 k 个 3 的数（信息学奥赛一本通-T1090）

原创 Alex\_McAvoy 发布于 2018-02-15 17:52:19 阅读数 5856 收藏  
展开

##### 【题目描述】

输入两个正整数  $m$  和  $k$ ，其中  $1 < m < 100000$ ， $1 < k < 5$ ，判断  $m$  能否被 19 整除，且恰好含有  $k$  个 3，如果满足条件，则输出 YES，否则，输出 NO。例如，输入：43833 3，满足条件，输出 YES。如果输入：39331 3，尽管有 3 个 3，但不能被 19 整除，也不满足条件，应输出 NO。

##### 【输入】

$m$  和  $k$  的值，中间用单个空格间隔。

##### 【输出】

满足条件时输出 YES，不满足时输出 NO。

##### 【输入样例】

43833 3

##### 【输出样例】

YES

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{

```

```

int m,k,a[5],i,num=0;

cin>>m>>k;
/*依次读取各位数字*/
a[0]=m/10000;
a[1]=m%10000/1000;
a[2]=m%1000/100;
a[3]=m%100/10;
a[4]=m%10;

for(i=0; i<=4; i++) //统计 3 的个数
    if(a[i]==3)
        num++;

if((num==k)&&(m%19==0))//依照题意判断
    cout<<"YES"<<endl;
else
    cout<<"NO"<<endl;

return 0;
}

```

### 1.1.1.4.3 第三节 循环嵌套

#### 1.1.1.4.3.1 求阶乘的和（信息学奥赛一本通-T1091）

##### 求阶乘的和（信息学奥赛一本通-T1091）

原创 Alex\_McAvoy 发布于 2018-02-15 18:14:54 阅读数 4277 收藏  
展开

##### 【题目描述】

给定正整数  $n$ ，求不大于  $n$  的正整数的阶乘的和（即求  $1!+2!+3!+\dots+n!$ ），输出阶乘的和。

##### 【输入】

输入有一行，包含一个正整数  $n$  ( $1 < n < 12$ )。

##### 【输出】

输出有一行：阶乘的和。

**【输入样例】**

5

**【输出样例】**

153

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int num,i;
    long int sum=0,temp=1;

    cin>>num;
    for(i=1; i<=num; i++)
    {
        temp=temp*i;//计算阶乘
        sum=sum+temp;//累加
    }
    cout<<sum<<endl;

    return 0;
}
```

**1.1.1.4.3.2 求出 e 的值（信息学奥赛一本通-T1092）****求出 e 的值（信息学奥赛一本通-T1092）**

原创 Alex\_McAvoy 发布于 2018-02-15 18:24:53 阅读数 5883 收藏  
展开

**【题目描述】**

利用公式  $e=1+1/1!+1/2!+1/3!+...+1/n!$ ，求 e 的值，要求保留小数点后 10 位。

**【输入】**

输入只有一行，该行包含一个整数 n ( $2 \leq n \leq 15$ )，表示计算 e 时累加到  $1/n!$

**【输出】**

输出只有一行，该行包含计算出来的 e 的值，要求打印小数点后 10 位。

**【输入样例】**

10

**【输出样例】**

2.7182818011

**【源程序】**

```

#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int num,i;
    double e=1;
    long long temp=1;

    cin>>num;
    for(i=1;i<=num;i++)
    {
        temp=temp*i;//计算阶乘
        e=e+1.0/temp;//累加
    }
    printf("%.10lf\n",e);

    return 0;
}

```

**1.1.1.4.3.3 计算多项式的值（信息学奥赛一本通-T1093）****计算多项式的值（信息学奥赛一本通-T1093）**

原创 Alex\_McAvoy 发布于 2018-02-15 18:31:50 阅读数 4918 收藏  
展开

**【题目描述】**

假定多项式的形式为  $X^n + X^{n-1} + \dots + X^2 + X + 1$ ，请计算给定单精度浮点数  $x$  和正整数  $n$  值的情况下这个多项式的值。 $x$  在 `float` 范围内， $n \leq 1000000$ 。多项式的值精确到小数点后两位，保证最终结果在 `float` 范围内。

**【输入】**

输入仅一行，包括  $x$  和  $n$ ，用单个空格隔开。 $x$  在 `float` 范围内， $n \leq 1000000$ 。

**【输出】**

输出一个实数，即多项式的值，精确到小数点后两位。保证最终结果在 `float` 范围内。

**【输入样例】**

2.0 4



**【输出样例】**

31.00

**【源程序】**

```

#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int num,i;
    double sum=1,temp=1,x;

    cin>>x>>num;
    for(i=1; i<=num; i++)
    {
        temp=temp*x;//计算  $x^i$ 
        sum=sum+temp;//累加  $x^i$ 
    }
    printf("%.2f\n",sum);

    return 0;
}

```

**1.1.1.4.3.4 与 7 无关的数（信息学奥赛一本通-T1094）****与 7 无关的数（信息学奥赛一本通-T1094）**

原创 Alex\_McAvoy 发布于 2018-02-15 18:39:05 阅读数 5422 收藏  
展开

**【题目描述】**

一个正整数，如果它能被 7 整除，或者它的十进制表示法中某一位上的数字为 7，则称其为与 7 相关的数。现求所有小于等于  $n(n < 100)$  与 7 无关的正整数的平方和。

**【输入】**

输入为一行,正整数  $n(n < 100)$ 。

**【输出】**

输出一行，包含一个整数，即小于等于  $n$  的所有与 7 无关的正整数的平方和。

**【输入样例】**

21

**【输出样例】**

2336

**【源程序】**

```

#include<iostream>
using namespace std;
int main()
{
    int n,i;
    long int sum=0;

    cin>>n;

    for(i=1; i<=n; i++)
        if( i%7!=0 && i/10!=7 && i%10!=7)//不能被 7 整除、个位、十位没有 7
            sum+=i*i;//累加平方和

    cout<<sum<<endl;
    return 0;
}

```

**1.1.1.4.3.5 数 1 的个数（信息学奥赛一本通-T1095）****数 1 的个数（信息学奥赛一本通-T1095）**

原创 Alex\_McAvoy 发布于 2018-02-16 05:02:22 阅读数 6583 收藏  
展开

**【题目描述】**

给定一个十进制正整数  $n(1 \leq n \leq 10000)$ ，写下从 1 到  $n$  的所有整数，然后数一下其中出现的数字“1”的个数。

例如当  $n=2$  时，写下 1,2。这样只出现了 1 个“1”；当  $n=12$  时，写下 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12。这样出现了 5 个“1”。

**【输入】**

正整数  $n$ 。  $1 \leq n \leq 10000$ 。

**【输出】**

一个正整数，即“1”的个数。

**【输入样例】**

12

**【输出样例】**

5

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,sum=1;
    int i,j;

    cin>>n;
    for(i=2; i<=n; i++) //从 2 开始枚举
        for(j=i; j>0; j/=10) //分离个位
            if(j%10==1) sum++; //若个位为 1，计数器+1

    cout<<sum<<endl;
    return 0;
}
```

**1.1.1.4.3.6 数字统计（信息学奥赛一本通-T1096）****数字统计（信息学奥赛一本通-T1096）**

原创 Alex\_McAvoy 发布于 2018-02-16 05:14:15 阅读数 4765 收藏  
展开

**【题目描述】**

请统计某个给定范围[L, R]的所有整数中，数字 2 出现的次数。

比如给定范围[2, 22]，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现 1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

**【输入】**

输入共 1 行，为两个正整数 L 和 R，之间用一个空格隔开

**【输出】**

输出共 1 行，表示数字 2 出现的次数。

**【输入样例】**

2 22

**【输出样例】**

6

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int left,right,sum=0;
    int i,j;

    cin>>left>>right;
    for(i=left; i<=right; i++)
        for(j=i; j>0; j/=10) //分离个位
            if(j%10==2) sum++;//若个位为 1，计数器+1

    cout<<sum<<endl;
    return 0;
}
```

**1.1.1.4.3.7 画矩形（信息学奥赛一本通-T1097）****画矩形（信息学奥赛一本通-T1097）**

原创 Alex\_McAvoy 发布于 2018-02-16 05:21:17 阅读数 6391 收藏  
展开

**【题目描述】**

根据参数，画出矩形。输入四个参数：前两个参数为整数，依次代表矩形的高和宽（高不少于 3 行不多于 10 行，宽不少于 5 列不多于 10 列）；第三个参数是一个字符，表示用来画图的矩形符号；第四个参数为 1 或 0，0 代表空心，1 代表实心。

**【输入】**

输入一行，包括四个参数：前两个参数为整数，依次代表矩形的高和宽（高不少于 3 行不多于 10 行，宽不少于 5 列不多于 10 列）；第三个参数是一个字符，表示用来画图的矩形符号；第四个参数为 1 或 0，0 代表空心，1 代表实心。

**【输出】**

输出画出的图形。

**【输入样例】**

7 7 @ 0

## 【输出样例】

```

@@@@@@@
@           @
@           @
@           @
@           @
@           @
@@@@@@@

```

## 【源程序】

```

#include <iostream>
using namespace std;
int main()
{
    int a,b;
    char ch;
    bool x;
    int i,j;

    cin>>a>>b;//行、列
    cin>>ch;//填充符号
    cin>>x;//空心或实心

    if(x)//实心
    {
        for(i=1; i<=a; i++)
        {
            for(j=1; j<=b; j++)    cout<<ch;
            cout<<endl;
        }
    }
    else//空心
    {
        for(i=1; i<=a; i++)
        {
            for(j=1; j<=b; j++)
            {
                if((i==1)||i==a||j==1||j==b)//四条边均按条件输出
                    cout<<ch;
                else
                    cout<<" ";
            }
            cout<<endl;
        }
    }
}

```

```

    cout<<endl;
    return 0;
}

```

### 1.1.1.4.3.8 质因数分解（信息学奥赛一本通-T1098）

#### 质因数分解（信息学奥赛一本通-T1098）

原创 Alex\_McAvoy 发布于 2018-02-16 05:30:23 阅读数 5762 收藏  
展开

##### 【题目描述】

已知正整数  $n$  是两个不同的质数的乘积，试求出较大的那个质数。

##### 【输入】

输入只有一行，包含一个正整数  $n$ 。

对于 60% 的数据， $6 \leq n \leq 1000$ 。

对于 100% 的数据， $6 \leq n \leq 2 \times 10^9$ 。

##### 【输出】

输出只有一行，包含一个正整数  $p$ ，即较大的那个质数。

##### 【输入样例】

21

##### 【输出样例】

7

##### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int n;
    int i,j;

    cin>>n;
    for(i=2; i<=sqrt(n); i++) //两个不同质数，其中必有一个≤sqrt(n)
    {
        if(n%i==0)//找到质数
        {
            cout<<n/i<<endl;//较大质数=n/较小质数，输出
        }
    }
}

```

```

        break;//输出后，终止
    }
}

return 0;
}

```

### 1.1.1.4.3.9 第 n 小的质数（信息学奥赛一本通-T1099）

#### 第 n 小的质数（信息学奥赛一本通-T1099）

原创 Alex\_McAvoy 发布于 2018-02-16 06:33:38 阅读数 7326 收藏  
展开

##### 【题目描述】

输入一个正整数 n，求第 n 小的质数。

##### 【输入】

一个不超过 10000 的正整数 n。

##### 【输出】

第 n 小的质数。

##### 【输入样例】

10

##### 【输出样例】

29

##### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int n;
    int i,j;
    int sum=0;

    cin>>n;
    for(i=2;; i++)
    {
        for(j=2; j<=sqrt(i)&&i%j!=0; j++); //判断 i 是否为素数
    }
}

```

```

        if(j>sqrt(i))//若是素数
        {
            sum++;//累加 sum, 该数是第 sum 小的
            if(sum==n)
            {
                cout<<i<<endl;
                break;
            }
        }
    }

    return 0;
}

```

点赞 3

### 1.1.1.4.3.10 金币（信息学奥赛一本通-T1100）

#### 金币（信息学奥赛一本通-T1100）

原创 Alex\_McAvoy 发布于 2018-02-16 06:18:49 阅读数 4620 收藏  
展开

##### 【题目描述】

国王将金币作为工资，发放给忠诚的骑士。第 1 天，骑士收到一枚金币；之后两天(第 2 天和第 3 天)里，每天收到两枚金币；之后三天(第 4、5、6 天)里，每天收到三枚金币；之后四天(第 7、8、9、10 天)里，每天收到四枚金币……这种工资发放模式会一直这样延续下去：当连续  $n$  天每天收到  $n$  枚金币后，骑士会在之后的连续  $n+1$  天里，每天收到  $n+1$  枚金币( $n$  为任意正整数)。

你需要编写一个程序，确定从第一天开始的给定天数内，骑士一共获得了多少金币。

##### 【输入】

一个整数（范围 1 到 10000），表示天数

##### 【输出】

骑士获得的金币数。

##### 【输入样例】

6

##### 【输出样例】

14



## 【源程序】

```

#include <iostream>
using namespace std;
int main()
{
    int day;
    int date=1,sum=0;
    int i,j=1;

    cin>>day;
    for(i=1; i<=day; i++)
    {
        sum+=date;//累加金币数
        if(i==j)//目前天数
        {
            date++;
            j+=date;
        }
    }
    cout<<sum<<endl;

    return 0;
}

```

## 1.1.1.4.3.11 不定方程求解（信息学奥赛一本通-T1101）

## 不定方程求解（信息学奥赛一本通-T1101）

原创 Alex\_McAvoy 发布于 2018-02-16 06:24:48 阅读数 4461 收藏  
展开

## 【题目描述】

给定正整数  $a$ ,  $b$ ,  $c$ 。求不定方程  $ax+by=c$  关于未知数  $x$  和  $y$  的所有非负整数解组数。

## 【输入】

一行, 包含三个正整数  $a$ ,  $b$ ,  $c$ , 两个整数之间用单个空格隔开。每个数均不大于 1000。

## 【输出】

一个整数, 即不定方程的非负整数解组数。

## 【输入样例】

2 3 18

## 【输出样例】

4

## 【源程序】

```

#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    int x,y;
    int sum=0;

    cin>>a>>b>>c;
    for(x=0; x<=c/a; x++) //由题意得: (c-by)/a=x,可知: c/a>=x
    {
        y=(c-a*x)/b;//由题意得: (c-ax)/b=y
        if(a*x+b*y==c) sum++;//符合条件, 记录
    }
    cout<<sum<<endl;

    return 0;
}

```

## 1.1.1.5 第五章 数组

### 1.1.1.5.1 第一节 一维数组

#### 1.1.1.5.1.1 与指定数字相同的数的个数（信息学奥赛一本通-T1102）

与指定数字相同的数的个数（信息学奥赛一本通-T1102）

原创 Alex\_McAvoy 发布于 2018-02-26 02:34:58 阅读数 5968 收藏  
展开

**【题目描述】**

输出一个整数序列中与指定数字相同的数的个数。

**【输入】**

输入包含三行：

第一行为  $N$ ，表示整数序列的长度( $N \leq 100$ )；

第二行为  $N$  个整数，整数之间以一个空格分开；

第三行包含一个整数，为指定的数字  $m$ 。

**【输出】**

输出为  $N$  个数中与  $m$  相同的数的个数。

**【输入样例】**

```
3
2 3 2
2
```

**【输出样例】**

```
2
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,m,a[101];
    int ans=0;
    int i;

    cin>>n;//共有 n 个数
    for(i=1;i<=n;i++)
        cin>>a[i];//读入 n 个数
    cin>>m;//指定数 m

    for(i=1;i<=n;i++)//对数据逐个进行比较
        if(a[i]==m)//若与指定数字相同
            ans++;//计数器+1

    cout<<ans<<endl;

    return 0;
}
```

### 1.1.1.5.1.2 陶陶摘苹果（信息学奥赛一本通-T1103）

#### 陶陶摘苹果（信息学奥赛一本通-T1103）

原创 Alex\_McAvoy 发布于 2018-02-26 02:36:22 阅读数 5583 收藏  
展开

##### 【题目描述】

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出 10 个苹果。苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有个 30 厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知 10 个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度，请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

##### 【输入】

包括两行数据。第一行包含 10 个 100 到 200 之间(包括 100 和 200)的整数(以厘米为单位)分别表示 10 个苹果到地面的高度，两个相邻的整数之间用一个空格隔开。第二行只包括一个 100 到 120 之间(包含 100 和 120)的整数(以厘米为单位)，表示陶陶把手伸直的时候能够达到的最大高度。

##### 【输出】

包括一行，这一行只包含一个整数，表示陶陶能够摘到的苹果的数目。

##### 【输入样例】

```
100 200 150 140 129 134 167 198 200 111
110
```

##### 【输出样例】

```
5
```

##### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int apple[10],high,total=0;
    int i;

    for(i=0;i<10;i++) cin>>apple[i];//输入苹果高度
    cin>>high;//输入伸直手的高度

    high+=30;//踩上板凳后的高度
    for(i=0;i<10;i++)//对每个苹果进行比较
        if(apple[i]<=high)//如果苹果高度<能够到的高度
```

```

    total++; //说明可以够到苹果，计数器+1

    cout<<total<<endl;

    return 0;
}

```

[点赞](#)  
[收藏](#)  
[分享](#)

### 1.1.1.5.1.3 计算书费（信息学奥赛一本通-T1104）

#### 计算书费（信息学奥赛一本通-T1104）

原创 Alex\_McAvoy 发布于 2018-02-26 02:49:41 阅读数 5952 收藏  
展开

##### 【题目描述】

下面是一个图书的单价表：

计算概论 28.9 元/本  
 数据结构与算法 32.7 元/本  
 数字逻辑 45.6 元/本  
 C++程序设计教程 78 元/本  
 人工智能 35 元/本  
 计算机体系结构 86.2 元/本  
 编译原理 27.8 元/本  
 操作系统 43 元/本  
 计算机网络 56 元/本  
 JAVA 程序设计 65 元/本

给定每种图书购买的数量，编程计算应付的总费用。

##### 【输入】

输入一行，包含 10 个整数(大于等于 0，小于等于 100)，分别表示购买的《计算概论》、《数据结构与算法》、《数字逻辑》、《C++程序设计教程》、《人工智能》、《计算机体系结构》、《编译原理》、《操作系统》、《计算机网络》、《JAVA 程序设计》的数量（以本为单位）。每两个整数用一个空格分开。

##### 【输出】

输出一行，包含一个浮点数 f，表示应付的总费用。精确到小数点后一位。

**【输入样例】**

1 5 8 10 5 1 1 2 3 4

**【输出样例】**

2410.2

**【源程序】**

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    double price[10]={28.9,32.7,45.6,78,35,86.2,27.8,43,56,65};//给定每种书的价格
    int num[10];
    double sum=0;
    int i;

    for(i=0;i<10;i++)
    {
        cin>>num[i];//输入要买的每本书的数量
        sum+=num[i]*price[i];//累加每种书的价格
    }
    printf("%.1f\n",sum);//保留 1 位小数输出

    return 0;
}
```

点赞

## 1.1.1.5.1.4 数组逆序重存放（信息学奥赛一本通-T1105）

### 数组逆序重存放（信息学奥赛一本通-T1105）

原创 Alex\_McAvoy 发布于 2018-02-26 02:50:28 阅读数 4972 收藏  
展开

**【题目描述】**

将一个数组中的值按逆序重新存放。例如，原来的顺序为 8,6,5,4,1。要求改为 1,4,5,6,8。

**【输入】**

输入为两行：第一行数组中元素的个数  $n$  ( $1 \leq n \leq 100$ )，第二行是  $n$  个整数，每两个整数之间用空格分隔。

**【输出】**

输出为一行：输出逆序后数组的整数，每两个整数之间用空格分隔。

**【输入样例】**

5  
8 6 5 4 1

**【输出样例】**

1 4 5 6 8

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int num[101];
    int i;

    cin>>n;//元素个数 n
    for(i=1;i<=n;i++)
        cin>>num[i];//n 个整数

    for(i=n;i>=1;i--)//倒序输出
        cout<<num[i]<<" ";

    cout<<endl;
    return 0;
}
```

### 1.1.1.5.1.5 年龄与疾病（信息学奥赛一本通-T1106）

#### 年龄与疾病（信息学奥赛一本通-T1106）

原创 Alex\_McAvoy 发布于 2018-02-26 02:55:20 阅读数 6672 收藏  
展开

**【题目描述】**

某医院想统计一下某项疾病的获得与否与年龄是否有关，需要对以前的诊断记录进行整理，按照 0-18、19-35、36-60、61 以上（含 61）四个年龄段统计的患病人数占总患病人数的比例。

**【输入】**

共 2 行，第一行为过往病人的数目  $n$  ( $0 < n \leq 100$ )，第二行为每个病人患病时的年龄。

**【输出】**

按照 0-18、19-35、36-60、61 以上(含 61)四个年龄段输出该段患病人数占总患病人数的比例，以百分比的形式输出，精确到小数点后两位。每个年龄段占一行，共四行。

**【输入样例】**

```
10
1 11 21 31 41 51 61 71 81 91
```

**【输出样例】**

```
20.00%
20.00%
20.00%
40.00%
```

**【源程序】**

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int n,i;
    int age[100];
    double percent[4]= {0};

    cin>>n;//病人数 n
    for(i=0; i<n; i++)
        cin>>age[i]; //患病年龄

    for(i=0; i<n; i++)
    {
        if(age[i]<=18)
            percent[0]++;//统计年龄在 19 岁以下的人数
        if((age[i]>=19)&&(age[i]<=35))
            percent[1]++;//统计年龄在 19 到 35 岁的人数
        if((age[i]>=36)&&(age[i]<=60))
            percent[2]++;//统计年龄在 36 到 60 岁的人数
        if(age[i]>=61)
            percent[3]++;//统计年龄在 60 岁以上的人数
    }

    cout<<setiosflags(ios::fixed)<<setprecision(2);//格式输出控制，保留两位小数
    for(i=0; i<4; i++) //转换成百分数的形式输出
    {
        percent[i]=percent[i]*100.0/n;
        cout<<percent[i]<<"%"<<endl;
```



```

    }

    return 0;
}

```

点赞 1

### 1.1.1.5.1.6 校门外的树（信息学奥赛一本通-T1107）

#### 校门外的树（信息学奥赛一本通-T1107）

原创 Alex\_McAvoy 发布于 2018-02-26 03:04:54 阅读数 6418 收藏  
展开

##### 【题目描述】

某校大门外长度为  $L$  的马路上有一排树，每两棵相邻的树之间的间隔都是 1 米。我们可以把马路看成一个数轴，马路的一端在数轴 0 的位置，另一端在  $L$  的位置；数轴上的每个整数点，即 0, 1, 2, …,  $L$ ，都种有一棵树。

由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

##### 【输入】

第一行有两个整数  $L$  ( $1 \leq L \leq 10000$ ) 和  $M$  ( $1 \leq M \leq 100$ )， $L$  代表马路的长度， $M$  代表区域的数目， $L$  和  $M$  之间用一个空格隔开。接下来的  $M$  行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点和终止点的坐标。

对于 20% 的数据，区域之间没有重合的部分；对于其它的数据，区域之间有重合的情况。

##### 【输出】

包括一行，这一行只包含一个整数，表示马路上剩余的树的数目。

##### 【输入样例】

```

500 3
150 300
100 200
470 471

```

##### 【输出样例】

```

298

```

## 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    int L,M,i,j;
    int section[10001]= {0};
    int a,b,tree_sum=0,tree_ex,tree_new;

    cin>>L>>M;//输入马路长度与区域数目

    for (i=0; i<M; i++)
    {
        cin>>a>>b;//输入起始点与终止点
        for(j=a; j<=b; j++) //令范围内的树移走
            section[j]=1;
    }

    for(i=0; i<=L; i++) //统计马路长度内被移走的树的数目
        if(section[i]==1)
            tree_sum++;

    tree_ex=L+1;//原来的树的数目
    tree_new=tree_ex-tree_sum;//移走后的树的数目

    cout<<tree_new<<endl;

    return 0;
}

```

## 1.1.1.5.1.7 向量点积计算（信息学奥赛一本通-T1108）

## 向量点积计算（信息学奥赛一本通-T1108）

原创 Alex\_McAvoy 发布于 2018-02-26 03:11:31 阅读数 4002 收藏  
展开

## 【题目描述】

在线性代数、计算几何中，向量点积是一种十分重要的运算。给定两个  $n$  维向量  $a=(a_1,a_2,...,a_n)$  和  $b=(b_1,b_2,...,b_n)$ ，求点积  $a \cdot b = a_1b_1 + a_2b_2 + ... + a_nb_n$ 。

## 【输入】

第一行是一个整数  $n(1 \leq n \leq 1000)$ 。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ 。

第三行包含  $n$  个整数  $b_1, b_2, \dots, b_n$ 。

相邻整数之间用单个空格隔开。每个整数的绝对值都不超过 1000。

**【输出】**

一个整数，即两个向量的点积结果。

**【输入样例】**

```
3
1 4 6
2 1 5
```

**【输出样例】**

```
36
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int a[1001],b[1001];
    int result=0;
    int i;

    cin>>n;//输入整数 n

    for(i=1; i<=n; i++) //输入向量 ai
        cin>>a[i];
    for(i=1; i<=n; i++) //输入向量 bi
        cin>>b[i];
    for(i=1; i<=n; i++) //计算向量点积
        result+=a[i]*b[i];

    cout<<result<<endl;
    return 0;
}
```

### 1.1.1.5.1.8 开关灯（信息学奥赛一本通-T1109）

#### 开关灯（信息学奥赛一本通-T1109）

原创 Alex\_McAvoy 发布于 2018-02-26 16:02:11 阅读数 7245 收藏  
展开

### 【题目描述】

假设有  $N$  盏灯( $N$  为不大于 5000 的正整数), 从 1 到  $N$  按顺序依次编号, 初始时全部处于开启状态; 有  $M$  个人( $M$  为不大于  $N$  的正整数)也从 1 到  $M$  依次编号。

第一个人(1 号)将灯全部关闭, 第二个人(2 号)将编号为 2 的倍数的灯打开, 第三个人(3 号)将编号为 3 的倍数的灯做相反处理 (即将打开的灯关闭, 将关闭的灯打开)。依照编号递增顺序, 以后的人都和 3 号一样, 将凡是自己编号倍数的灯做相反处理。

请问: 当第  $M$  个人操作之后, 哪几盏灯是关闭的, 按从小到大输出其编号, 其间用逗号间隔。

### 【输入】

输入正整数  $N$  和  $M$ , 以单个空格隔开。

### 【输出】

顺次输出关闭的灯的编号, 其间用逗号间隔。

### 【输入样例】

10 10

### 【输出样例】

1,4,9

### 【源程序】

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    int n,m;
    int a[5001];
    int i,j;
    int flag=1;

    cin>>n>>m;

    memset(a,0,sizeof(a));//初始化将灯关上
    for(i=1; i<=m; i++) //m 个人依次进行操作
        for(j=1; j<=n; j++) //对 n 盏灯进行操作
            if(j%i==0)//只对 i 编号的倍数进行操作
                a[j]=!a[j];//对灯进行相反处理
```

```

for(i=1; i<=n; i++) //输出灯是关闭的序号
{
    if(a[i])
    {
        if(flag)
            flag=0;//仅第一次不用先输出逗号
        else
            cout<<" ";
        cout<<i;
    }
}
cout<<endl;

return 0;
}

```

### 1.1.1.5.1.9 查找特定的值（信息学奥赛一本通-T1110）

#### 查找特定的值（信息学奥赛一本通-T1110）

原创 Alex\_McAvoy 发布于 2018-02-26 16:15:45 阅读数 4962 收藏  
展开

##### 【题目描述】

在一个序列(下标从 1 开始)中查找一个给定的值，输出第一次出现的位置。

##### 【输入】

第一行包含一个正整数  $n$ ，表示序列中元素个数。 $1 \leq n \leq 10000$ 。

第二行包含  $n$  个整数，依次给出序列的每个元素，相邻两个整数之间用单个空格隔开。  
元素的绝对值不超过 10000。

第三行包含一个整数  $x$ ，为需要查找的特定值。 $x$  的绝对值不超过 10000。

##### 【输出】

若序列中存在  $x$ ，输出  $x$  第一次出现的下标；否则输出 -1。

##### 【输入样例】

```

5
2 3 6 7 3
3

```

##### 【输出样例】

```

2

```

**【源程序】**

```

#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    int n,x;
    int a[10001];
    int i;
    int flag=0;

    cin>>n;//输入序列元素个数 n
    for(i=1; i<=n; i++) //输入每个元素
        cin>>a[i];
    cin>>x;//输入查找特定值

    for(i=1; i<=n; i++) //对每个数依次比较
    {
        if(a[i]==x)//若查找到了
        {
            flag=1;//记录标志，说明数列中存在 x
            cout<<i<<endl;//输出下标位置
            break;//终止循环
        }
    }
    if(flag==0)//如果记录标志为 0，说明数列中不存在 x
        cout<<-1<<endl;//输出-1

    return 0;
}

```

**1.1.1.5.1.10 不高兴的津津（信息学奥赛一本通-T1111）****不高兴的津津（信息学奥赛一本通-T1111）**

原创 Alex\_McAvoy 发布于 2018-02-26 16:17:42 阅读数 4934 收藏  
展开

**【题目描述】**

津津上初中了。妈妈认为津津应该更加用功学习，所以津津除了上学之外，还要参加妈妈为她报名的各科复习班。另外每周妈妈还会送她去学习朗诵、舞蹈和钢琴。但是津津如果一天上课超过八个小时就会不高兴，而且上得越久就会越不高兴。假设津津不会因为其它事不高兴，并且她的不高兴不会持续到第二天。请你帮忙检查一下津津下周的日程安排，看看下周她会不会不高兴；如果会的话，哪天最不高兴。

**【输入】**

包括七行数据，分别表示周一到周日的日程安排。每行包括两个小于 10 的非负整数，用空格隔开，分别表示津津在学校上课的时间和妈妈安排她上课的时间。

**【输出】**

包括一行，这一行只包含一个数字。如果不会不高兴则输出 0，如果会则输出最不高兴的是周几（用 1, 2, 3, 4, 5, 6, 7 分别表示周一，周二，周三，周四，周五，周六，周日）。如果有两天或两天以上不高兴的程度相当，则输出时间最靠前的一天。

**【输入样例】**

```
5 3
6 2
7 2
5 3
5 4
0 4
0 6
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int school,extra;
    int time[8];
    int day,max=0;
    int i;

    for(i=1; i<=7; i++)
    {
        cin>>school>>extra;//输入在校时间与额外上课时间
        time[i]=school+extra;//记录每天上课时间
    }

    for(i=1; i<=7; i++)
    {
        if(time[i]>max)
        {
            max=time[i];//找出最长上课时间
            day=i;//记录当天是周几
        }
    }
}
```

```

if(max>8)
    cout<<day<<endl;//如果这周最长上课时间大于 8 小时，输出最不高兴的时间
else
    cout<<0<<endl;//如果没有，输出 0
return 0;
}

```

### 1.1.1.5.1.11 最大值和最小值的差（信息学奥赛一本通-T1112）

#### 最大值和最小值的差（信息学奥赛一本通-T1112）

原创 Alex\_McAvoy 发布于 2018-02-26 16:39:41 阅读数 4520 收藏  
展开

##### 【题目描述】

输出一个整数序列中最大的数和最小的数的差。

##### 【输入】

第一行为  $M$ ，表示整数个数，整数个数不会大于 10000；

第二行为  $M$  个整数，以空格隔开，每个整数的绝对值不会大于 10000。

##### 【输出】

输出  $M$  个数中最大值和最小值的差。

##### 【输入样例】

```

5
2 5 7 4 2

```

##### 【输出样例】

```

5

```

##### 【源程序】

```

#include <iostream>
using namespace std;
int main()
{
    int m,a[10001];
    intmax=0,min=99999;
    int difference;
    int i;

```



```

cin>>m;//输入整数个数 m
for(i=1;i<=m;i++)
{
    cin>>a[i];//输入 m 个整数
    if(a[i]>max) max=a[i];//记录最大值
    if(a[i]<min) min=a[i];//记录最小值
}
difference=max-min;//计算最大值与最小值的差
cout<<difference<<endl;
return 0;
}

```

### 1.1.1.5.1.12 不与最大数相同的数字之和(信息学奥赛一本通-T1113)

#### 不与最大数相同的数字之和（信息学奥赛一本通-T1113）

原创 Alex\_McAvoy 发布于 2018-02-26 16:48:56 阅读数 4557 收藏  
展开

##### 【题目描述】

输出一个整数数列中不与最大数相同的数字之和。

##### 【输入】

输入分为两行：

第一行为  $N$  ( $N$  为接下来数的个数， $N \leq 100$ )；

第二行  $N$  个整数，数与数之间以一个空格分开，每个整数的范围是 -1000,000 到 1000,000。

##### 【输出】

输出为  $N$  个数中除去最大数其余数字之和。

##### 【输入样例】

```

3
1 2 3

```

##### 【输出样例】

```

3

```

##### 【源程序】

```

#include <iostream>
using namespace std;

```

```

int main()
{
    int n,a[101];
    int max=-1000000;
    int sum=0;
    int i;

    cin>>n;//输入整数个数 n
    for(i=1;i<=n;i++)
    {
        cin>>a[i];//输入 n 个整数
        if(a[i]>max) max=a[i];//记录最大值
    }
    for(i=1;i<=n;i++)//计算除去与最大值相同的数的和
        if(a[i]!=max)
            sum+=a[i];

    cout<<sum<<endl;
    return 0;
}

```

### 1.1.1.5.1.13 白细胞计数（信息学奥赛一本通-T1114）

#### 白细胞计数（信息学奥赛一本通-T1114）

原创 Alex\_McAvoy 发布于 2018-02-26 17:22:42 阅读数 6110 收藏  
展开

##### 【题目描述】

医院采样了某临床病例治疗期间的白细胞数量样本  $n$  份，用于分析某种新抗生素对该病例的治疗效果。为了降低分析误差，要先从这  $n$  份样本中去除一个数值最大的样本和一个数值最小的样本，然后将剩余  $n-2$  个有效样本的平均值作为分析指标。同时，为了观察该抗生素的疗效是否稳定，还要给出该平均值的误差，即所有有效样本（即不包括已扣除的两个样本）与该平均值之差的绝对值的最大值。

现在请你编写程序，根据提供的  $n$  个样本值，计算出该病例的平均白细胞数量和对应的误差。

##### 【输入】

输入的第一行是一个正整数  $n$  ( $2 < n \leq 300$ )，表明共有  $n$  个样本。

以下共有  $n$  行，每行为一个浮点数，为对应的白细胞数量，其单位为  $10^9/L$ 。数与数之间以一个空格分开。

**【输出】**

输出为两个浮点数，中间以一个空格分开。分别为平均白细胞数量和对应的误差，单位也是  $10^9/L$ 。计算结果需保留到小数点后 2 位。

**【输入样例】**

```
5
12.0
13.0
11.0
9.0
10.0
```

**【输出样例】**

```
11.00 1.00
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    double n;
    int flag_max,flag_min;
    int i;
    double max=-99999,min=99999;
    double a[3001],b[3001],sum=0;
    double average,ave_max=-99999;

    cin>>n;//输入样本数量 n

    for(i=1;i<=n;i++)
    {
        cin>>a[i];//输入样本值
        if(a[i]>max)
        {
            max=a[i];//记录最大值
            flag_max=i;//记录最大值位置
        }
        if(a[i]<min)
        {
            min=a[i];//记录最小值
            flag_min=i;//记录最小值位置
        }
        sum+=a[i];//计算 n 个样本白细胞数量之和
```

```

}
average=1.0*(sum-max-min)/(n-2);//计算平均白细胞数量

for(i=1;i<=n;i++)
{
    if(i!=flag_max&&i!=flag_min)//不计算已舍去的两个样本
    {
        b[i]=a[i]-average;//记录每个样本与平均数之差的绝对值
        if(b[i]<0)
            b[i]=-b[i];
    }
    else
        b[i]=-99999;//否则赋值最小，使其不可能为最大值
}
for(i=1;i<=n;i++)//找最大的差值
    if(b[i]>ave_max)
        ave_max=b[i];

printf("%.2lf %.2lf\n",average,ave_max);
return 0;
}

```

### 1.1.1.5.1.14 直方图（信息学奥赛一本通-T1115）

#### 直方图（信息学奥赛一本通-T1115）

原创 Alex\_McAvoy 发布于 2018-02-26 17:36:33 阅读数 4411 收藏  
展开

##### 【题目描述】

现在请你编写程序，根据提供的  $n$  个样本值，计算出该病例的平均白细胞数量和对应的误差。给定一个非负整数数组，统计里面每一个数的出现次数。我们只统计到数组里最大的数。

假设  $F_{\max}$  ( $F_{\max} < 10000$ ) 是数组里最大的数，那么我们只统计  $\{0, 1, 2, \dots, F_{\max}\}$  里每个数出现的次数。

##### 【输入】

以下共有  $n$  行，每行为一个浮点数，为对应的白细胞数量，其单位为  $10^9/L$ 。数与数之间以一个空格分开。第一行  $n$  是数组的大小。  $1 \leq n \leq 10000$ 。

紧接着一行是数组的  $n$  个元素。

**【输出】**

按顺序输出每个数的出现次数，一行一个数。如果没有出现过，则输出 0。

对于例子中的数组，最大的数是 3，因此我们只统计{0,1,2,3}的出现频数。

**【输入样例】**

```
5
1 1 2 3 1
```

**【输出样例】**

```
0
3
1
1
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,x;
    int a[10001]={0};
    int max=-9999,flag;
    int i;

    /*桶排的思想*/

    cin>>n;//输入数组大小 n
    for(i=1;i<=n;i++)
    {
        cin>>x;//输入元素
        a[x]++;
        if(x>max)
            max=x;//记录最大值
    }
    for(i=0;i<=max;i++)//输出到 max 为止的频数
        cout<<a[i]<<endl;

    return 0;
}
```

点赞

### 1.1.1.5.1.15 最长平台（信息学奥赛一本通-T1116）

#### 最长平台（信息学奥赛一本通-T1116）

原创 Alex\_McAvoy 发布于 2018-02-26 17:47:42 阅读数 5016 收藏  
展开

##### 【题目描述】

已知一个已经从小到大排序的数组，这个数组的一个平台（Plateau）就是连续的一串值相同的元素，并且这一串元素不能再延伸。例如，在 1, 2, 2, 3, 3, 3, 4, 5, 5, 6 中 1, 2-2, 3-3-3, 4, 5-5, 6 都是平台。

试编写一个程序，接收一个数组，把这个数组最长的平台找出来。在上面的例子中 3-3-3 就是最长的平台。

##### 【输入】

输出最长平台的长度。

##### 【输出】

按顺序输出每个数的出现次数，一行一个数。如果没有出现过，则输出 0。

对于例子中的数组，最大的数是 3，因此我们只统计{0,1,2,3}的出现频数。

##### 【输入样例】

```
10
1 2 2 3 3 3 4 5 5 6
```

##### 【输出样例】

```
3
```

##### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n,a[100001];
    int max=-9999,len=1;
    int i;

    cin>>n;//输入数组大小 n
    a[0]=-3456497;//保证 a[0]与 a[1]不相同
    for(i=1;i<=n;i++)
    {
        cin>>a[i];//输入元素
```

```

if(a[i]==a[i-1])//若当前元素与之前一个元素相同
    len++;//平台长度+1
else//否则
    len=1;//平台长度重归为 1
if(len>max)//记录最大平台值
    max=len;
}
cout<<max<<endl;
return 0;
}

```

### 1.1.1.5.1.16 整数去重（信息学奥赛一本通-T1117）

#### 整数去重（信息学奥赛一本通-T1117）

原创 Alex\_McAvoy 发布于 2018-02-26 18:06:52 阅读数 5705 收藏  
展开

##### 【题目描述】

给定含有  $n$  个整数的序列，要求对这个序列进行去重操作。所谓去重，是指对这个序列中每个重复出现的数，只保留该数第一次出现的位置，删除其余位置。

##### 【输入】

输入包含两行：

第一行包含一个正整数  $n$  ( $1 \leq n \leq 20000$ )，表示第二行序列中数字的个数；

第二行包含  $n$  个整数，整数之间以一个空格分开。每个整数大于等于 10、小于等于 5000。

##### 【输出】

输出只有一行，按照输入的顺序输出其中不重复的数字，整数之间用一个空格分开。

##### 【输入样例】

```

5
10 12 93 12 75

```

##### 【输出样例】

```

10 12 93 75

```

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{

```

```

int n;
int a[20001];
bool b[20001];
int i,j;

cin>>n;//输入数组大小 n
for(i=1; i<=n; i++)
{
    cin>>a[i];//输入元素
    b[i]=true;//初始化为没有重复
}

for(i=1; i<=n; i++)
    if(b[i])//判断 a[i]是否被标记为重复
        for(j=i+1; j<=n; j++) //寻找 a[i]后是否有重复的数
            if(a[i]==a[j])//如果有重复的数
                b[j]=false;//标记为不能输出

for(i=1; i<=n; i++) //输出没有重复的数
    if(b[i]==true)
        cout<<a[i]<<" ";
cout<<endl;

return 0;
}

```

### 1.1.1.5.1.17 铺地毯（信息学奥赛一本通-T1118）

#### 铺地毯（信息学奥赛一本通-T1118）

原创 Alex\_McAvoy 发布于 2018-02-26 18:19:55 阅读数 4421 收藏  
展开

##### 【题目描述】

为了准备一个独特的颁奖典礼，组织者在会场的一片矩形区域（可看做是平面直角坐标系的第一象限）铺上一些矩形地毯。一共有  $n$  张地毯，编号从 1 到  $n$ 。现在将这些地毯按照编号从小到大的顺序平行于坐标轴先后铺设，后铺的地毯覆盖在前面已经铺好的地毯之上。地毯铺设完成后，组织者想知道覆盖地面某个点的最上面的那张地毯的编号。注意：在矩形地毯边界和四个顶点上的点也算被地毯覆盖。

样例解释:如下图，1 号地毯用实线表示，2 号地毯用虚线表示，3 号用双实线表示，覆盖点 (2,2) 的最上面一张地毯是 3 号地毯。



**【输入】**

第一行，一个整数  $n$ ，表示总共有  $n$  张地毯。

接下来的  $n$  行中，第  $i+1$  行表示编号  $i$  的地毯的信息，包含四个正整数  $a, b, g, k$ ，每两个整数之间用一个空格隔开，分别表示铺设地毯的左下角的坐标  $(a, b)$  以及地毯在  $x$  轴和  $y$  轴方向的长度。

第  $n+2$  行包含两个正整数  $x$  和  $y$ ，表示所求的地面的点的坐标  $(x, y)$ 。

**【输出】**

输出共 1 行，一个整数，表示所求的地毯的编号；若此处没有被地毯覆盖则输出 -1。

**【输入样例】**

```
3
1 0 2 3
0 2 3 3
2 1 3 3
2 2
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<iostream>
using namespace std;
int main()
{
    int n,x,y;
    int a[10001],b[10001],g[10001],k[10001];
    int i;

    cin>>n;//输入地毯数 n
    for(i=1;i<=n;i++)
    {
        cin>>a[i]>>b[i];//输入地毯左下角坐标(a,b)
        cin>>g[i]>>k[i];//输入地毯在 x 轴、y 轴的长度
    }
    cin>>x>>y;//输入所求的地面的点的坐标(x,y)

    for(i=n;i>=1;i--)//找覆盖在最上的地毯，所以上到下找
    {
        if((x>=a[i]&&x<=a[i]+g[i]) && (y>=b[i]&&y<=b[i]+k[i]))//判断是否在范围内
        {
            cout<<i<<endl;//找到后输出
```

```

        return 0;//结束
    }
}
cout<<-1<<endl;//没有覆盖，输出-1
return 0;
}

```

### 1.1.1.5.2 第二节 二维数组

#### 1.1.1.5.2.1 矩阵交换行（信息学奥赛一本通-T1119）

##### 【题目描述】

给定一个  $5 \times 5$  的矩阵(数学上，一个  $r \times c$  的矩阵是一个由  $r$  行  $c$  列元素排列成的矩形阵列)，将第  $n$  行和第  $m$  行交换，输出交换后的结果。

##### 【输入】

输入共 6 行，前 5 行为矩阵的每一行元素,元素与元素之间以一个空格分开。

第 6 行包含两个整数  $m$ 、 $n$ ，以一个空格分开 ( $1 \leq m, n \leq 5$ )。

##### 【输出】

输出交换之后的矩阵，矩阵的每一行元素占一行，元素之间以一个空格分开。

##### 【输入样例】

```

1 2 2 1 2
5 6 7 8 3
9 3 0 5 3
7 2 1 4 6

```

```
3 0 8 2 4
```

```
1 5
```

### 【输出样例】

```
3 0 8 2 4
```

```
5 6 7 8 3
```

```
9 3 0 5 3
```

```
7 2 1 4 6
```

```
1 2 2 1 2
```

### 【源程序】

```
#include<iostream>

#include<algorithm>

using namespace std;

int main()
{
    int n,m;

    int a[6][6];

    int i,j;

    for(i=1;i<=5;i++)//输入矩阵元素
        for(j=1;j<=5;j++)
            cin>>a[i][j];

    cin>>n>>m;//输入要交换的两行
```

```

for(i=1;i<=5;i++)//交换两行元素

    swap(a[m][i],a[n][i]);

for(i=1;i<=5;i++)//输出矩阵

{

    for(j=1;j<=5;j++)

        cout<<a[i][j]<<" ";

    cout<<endl;

}

return 0;

}

```

### 1.1.1.5.2.2 同行列对角线的格（信息学奥赛一本通-T1120）

#### 同行列对角线的格（信息学奥赛一本通-T1120）

原创 Alex\_McAvoy 发布于 2018-02-27 17:25:30 阅读数 5977 收藏  
展开

##### 【题目描述】

输入三个自然数  $N$ ,  $i$ ,  $j$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ), 输出在一个  $N \times N$  格的棋盘中(行列均从 1 开始编号), 与格子  $(i, j)$  同行、同列、同一对角线的所有格子的位置。

如:  $n=4$ ,  $i=2$ ,  $j=3$  表示了棋盘中的第二行第三列的格子,

当  $n=4$ ,  $i=2$ ,  $j=3$  时, 输出的结果是:

(2,1) (2,2) (2,3) (2,4)	同一行上格子的位置
(1,3) (2,3) (3,3) (4,3)	同一列上格子的位置
(1,2) (2,3) (3,4)	左上到右下对角线上的格子的位置
(4,1) (3,2) (2,3) (1,4)	左下到右上对角线上的格子的位置

**【输入】**

一行，三个自然数  $N, i, j$ ，相邻两个数之间用单个空格隔开( $1 \leq N \leq 10$ )。

**【输出】**

第一行：从左到右输出同一行格子位置；

第二行：从上到下输出同一列格子位置；

第三行：从左上到右下输出同一对角线格子位置；

第四行：从左下到右上输出同一对角线格子位置。

其中每个格子位置用如下格式输出： $(x,y)$ ， $x$  为行号， $y$  为列号，采用英文标点，中间无空格。相邻两个格子位置之间用单个空格隔开。

**【输入样例】**

4 2 3

**【输出样例】**

(2,1) (2,2) (2,3) (2,4)

(1,3) (2,3) (3,3) (4,3)

(1,2) (2,3) (3,4)

(4,1) (3,2) (2,3) (1,4)

**【源程序】**

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int n,x,y;
    int i,j;

    cin>>n;//输入棋盘大小 n
    cin>>x>>y;//输入棋子所在行列 (x, y)

    for(i=1;i<=n;i++)//输出同一行的位置
        printf("(%d,%d) ",x,i);
    cout<<endl;
    for(i=1;i<=n;i++)//输出同一列的位置
        printf("(%d,%d) ",i,y);
    cout<<endl;

    for(i=1;i<=n;i++)//输出左上到右下对角线的位置
        for(j=1;j<=n;j++)
            if(x-y==i-j)
```

```

        printf("(%d,%d) ",i,j);
    cout<<endl;
    for(i=n;i>=1;i--)//输出左下到右上对角线的位置
        for(j=n;j>=1;j--)
            if(x+y==i+j)
                printf("(%d,%d) ",i,j);
    cout<<endl;
    return 0;
}

```

### 1.1.1.5.2.3 计算矩阵边缘元素之和（信息学奥赛一本通-T1121）

#### 计算矩阵边缘元素之和（信息学奥赛一本通-T1121）

原创 Alex\_McAvoy 发布于 2018-02-27 17:33:21 阅读数 4996 收藏  
展开

##### 【题目描述】

输入一个整数矩阵，计算位于矩阵边缘的元素之和。所谓矩阵边缘的元素，就是第一行和最后一行的元素以及第一列和最后一列的元素。

##### 【输入】

第一行分别为矩阵的行数  $m$  和列数  $n$  ( $m < 100$ ,  $n < 100$ )，两者之间以一个空格分开。

接下来输入的  $m$  行数据中，每行包含  $n$  个整数，整数之间以一个空格分开。

##### 【输出】

输出对应矩阵的边缘元素和。

##### 【输入样例】

```

3 3
3 4 1
3 7 1
2 0 1

```

##### 【输出样例】

```

15

```

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{

```

```

int n,m;
int a[101][101];
int sum=0;
int i,j;

cin>>n>>m;//输入矩阵行列

for(i=1; i<=n; i++)
{
    for(j=1; j<=m; j++)
    {
        cin>>a[i][j];//读入元素
        if(i==1||i==n||j==1||j==m)//判断是不是边缘
            sum+=a[i][j];//如果时，累加
    }
}

cout<<sum<<endl;
return 0;
}

```

#### 1.1.1.5.2.4 计算鞍点（信息学奥赛一本通-T1122）

##### 计算鞍点（信息学奥赛一本通-T1122）

原创 Alex\_McAvoy 发布于 2018-02-27 17:52:30 阅读数 4792 收藏

展开

##### 【题目描述】

给定一个  $5 \times 5$  的矩阵，每行只有一个最大值，每列只有一个最小值，寻找这个矩阵的鞍点。鞍点指的是矩阵中的一个元素，它是所在行的最大值，并且是所在列的最小值。

例如：在下面的例子中（第 4 行第 1 列的元素就是鞍点，值为 8）。

```

11 3 5 6 9
12 4 7 8 10
10 5 6 9 11
8 6 4 7 2
15 10 11 20 25

```

##### 【输入】

输入包含一个 5 行 5 列的矩阵。

**【输出】**

如果存在鞍点，输出鞍点所在的行、列及其值，如果不存在，输出"not found"。

**【输入样例】**

```
11 3 5 6 9
12 4 7 8 10
10 5 6 9 11
8 6 4 7 2
15 10 11 20 25
```

**【输出样例】**

```
4 1 8
```

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int a[5][5];
    int i,j;
    intmax,point,flag=1,work=1;

    for(i=0;i<5;i++)//数据输入
        for(j=0; j<5; j++)
            cin>>a[i][j];

    for(i=0;i<5;i++)
    {
        max=a[i][0];//每循环一行初始化最大值
        point=0;

        for(j=1;j<5;j++)
        {
            if(max<a[i][j])//寻找最大值
            {
                max=a[i][j];
                point=j;//记录所在列
            }
        }
        flag=1;
        for(j=0;j<5;j++)
        {
            if(a[j][point]<max)//进行比较确定是否是所在列最小值
                flag=0;
        }
    }
}
```



```

    if(flag)//如果满足行最大，列最小，按要求输出，结束程序
    {
        cout<<i+1<<" "<<point+1<<" "<<max<<endl;
        return 0;
    }
}
cout<<"not found"<<endl;//如果未满足，输出 not found
return 0;
}

```

点赞

### 1.1.1.5.2.5 图像相似度（信息学奥赛一本通-T1123）

#### 图像相似度（信息学奥赛一本通-T1123）

原创 Alex\_McAvoy 发布于 2018-02-27 18:25:14 阅读数 3392 收藏  
展开

##### 【题目描述】

给出两幅相同大小的黑白图像（用 0-1 矩阵）表示，求它们的相似度。说明：若两幅图像在相同位置上的像素点颜色相同，则称它们在该位置具有相同的像素点。两幅图像的相似度定义为相同像素点数占总像素点数的百分比。

##### 【输入】

第一行包含两个整数  $m$  和  $n$ ，表示图像的行数和列数，中间用单个空格隔开。 $1 \leq m \leq 100$ ,  $1 \leq n \leq 100$ 。

之后  $m$  行，每行  $n$  个整数 0 或 1，表示第一幅黑白图像上各像素点的颜色。相邻两个数之间用单个空格隔开。

之后  $m$  行，每行  $n$  个整数 0 或 1，表示第二幅黑白图像上各像素点的颜色。相邻两个数之间用单个空格隔开。

##### 【输出】

一个实数，表示相似度（以百分比的形式给出），精确到小数点后两位。

##### 【输入样例】

```

3 3
1 0 1
0 0 1
1 1 0

```

```
1 1 0
0 0 1
0 0 1
```

【输出样例】

44.44

【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int main()
{
    int m,n;
    int a[101][101],b[101][101];
    int sum=0;
    int i,j;
    double semblance;

    cin>>m>>n;//输入图像列数、行数

    for(i=1;i<=m;i++)//输入第一个图像
        for(j=1;j<=n;j++)
            cin>>a[i][j];

    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
        {
            cin>>b[i][j];//输入第二个图像
            if(a[i][j]==b[i][j])//与第一个图像比较，若像素点颜色相同，计数器+1
                sum++;
        }

    semblance=sum*1.0/(n*m)*100;//相似度计算
    printf("%.2lf\n",semblance);

    return 0;
}
```

点赞

### 1.1.1.5.2.6 矩阵加法（信息学奥赛一本通-T1124）

矩阵加法（信息学奥赛一本通-T1124）

原创 Alex\_McAvoy 发布于 2018-02-27 18:31:57 阅读数 4063 收藏  
展开

### 【题目描述】

输入两个  $n$  行  $m$  列的矩阵  $A$  和  $B$ ，输出它们的和  $A+B$ 。

### 【输入】

第一行包含两个整数  $n$  和  $m$ ，表示矩阵的行数和列数( $1 \leq n \leq 100$ ,  $1 \leq m \leq 100$ )。

接下来  $n$  行，每行  $m$  个整数，表示矩阵  $A$  的元素。

接下来  $n$  行，每行  $m$  个整数，表示矩阵  $B$  的元素。

相邻两个整数之间用单个空格隔开，每个元素均在  $1 \sim 1000$  之间。

### 【输出】

$n$  行，每行  $m$  个整数，表示矩阵加法的结果。相邻两个整数之间用单个空格隔开。

### 【输入样例】

```
3 3
1 2 3
1 2 3
1 2 3
1 2 3
1 2 3
4 5 6
7 8 9
```

### 【输出样例】

```
2 4 6
5 7 9
8 10 12
```

### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int m,n;
    int a[101][101],b[101][101];
    int i,j;

    cin>>m>>n;//输入矩阵列数、行数

    for(i=1;i<=m;i++)//输入第一个矩阵
        for(j=1;j<=n;j++)
            cin>>a[i][j];
```

```

for(i=1;i<=m;i++)//输入第二个矩阵
    for(j=1;j<=n;j++)
        cin>>b[i][j];

for(i=1;i<=m;i++)//输出矩阵和
{
    for(j=1;j<=n;j++)
        cout<<a[i][j]+b[i][j]<<" ";
    cout<<endl;
}
return 0;
}

```

### 1.1.1.5.2.7 矩阵乘法（信息学奥赛一本通-T1125）

#### 矩阵乘法（信息学奥赛一本通-T1125）

原创 Alex\_McAvoy 发布于 2018-02-27 18:42:38 阅读数 2709 收藏  
展开

##### 【题目描述】

计算两个矩阵的乘法。 $n \times m$  阶的矩阵  $A$  乘以  $m \times k$  阶的矩阵  $B$  得到的矩阵  $C$  是  $n \times k$  阶的，且  $C[i][j] = A[i][0] \times B[0][j] + A[i][1] \times B[1][j] + \dots + A[i][m-1] \times B[m-1][j]$  ( $C[i][j]$  表示  $C$  矩阵中第  $i$  行第  $j$  列元素)。

##### 【输入】

相邻两个整数之间用单个空格隔开，每个元素均在  $1 \sim 1000$  之间。第一行为  $n, m, k$ ，表示  $A$  矩阵是  $n$  行  $m$  列， $B$  矩阵是  $m$  行  $k$  列， $n, m, k$  均小于  $100$ 。

然后先后输入  $A$  和  $B$  两个矩阵， $A$  矩阵  $n$  行  $m$  列， $B$  矩阵  $m$  行  $k$  列，矩阵中每个元素的绝对值不会大于  $1000$ 。

##### 【输出】

输出矩阵  $C$ ，一共  $n$  行，每行  $k$  个整数，整数之间以一个空格分开。

##### 【输入样例】

```

3 2 3
1 1
1 1
1 1
1 1 1
1 1 1

```

##### 【输出样例】

```

2 2 2

```

```
2 2 2
2 2 2
```

## 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int n,m,k;
    int a[101][101],b[101][101],c[101][101]={0};
    int i,j,l;

    cin>>n>>m>>k;//输入矩阵列数、行数

    for(i=1;i<=n;i++)//输入第一个矩阵
        for(j=1;j<=m;j++)
            cin>>a[i][j];

    for(i=1;i<=m;i++)//输入第二个矩阵
        for(j=1;j<=k;j++)
            cin>>b[i][j];

    for(i=1;i<=n;i++)//计算矩阵 C
        for(j=1;j<=k;j++)
            for(l=1;l<=m;l++)
                c[i][j]+=a[i][l]*b[l][j];

    for(i=1;i<=n;i++)//输出 n*k 阶的矩阵 C
    {
        for(j=1;j<=k;j++)
            cout<<c[i][j]<<" ";
        cout<<endl;
    }
    return 0;
}
```

### 1.1.1.5.2.8 矩阵转置（信息学奥赛一本通-T1126）

#### 矩阵转置（信息学奥赛一本通-T1126）

原创 Alex\_McAvoy 发布于 2018-02-27 18:46:56 阅读数 3918 收藏  
展开

**【题目描述】**

输入一个  $n$  行  $m$  列的矩阵  $A$ ，输出它的转置  $A^T$ 。

**【输入】**

第一行包含两个整数  $n$  和  $m$ ，表示矩阵  $A$  的行数和列数( $1 \leq n \leq 100$ ,  $1 \leq m \leq 100$ )。

接下来  $n$  行，每行  $m$  个整数，表示矩阵  $A$  的元素。相邻两个整数之间用单个空格隔开，每个元素均在  $1 \sim 1000$  之间。

**【输出】**

$m$  行，每行  $n$  个整数，为矩阵  $A$  的转置。相邻两个整数之间用单个空格隔开。

**【输入样例】**

```
3 3
1 2 3
4 5 6
7 8 9
```

**【输出样例】**

```
1 4 7
2 5 8
3 6 9
```

**【源程序】**

```
#include <iostream>
using namespace std;
int main()
{
    int m,n;
    int a[101][101];
    int i,j;

    cin>>m>>n;//输入矩阵列数、行数

    for(i=1;i<=m;i++)//输入矩阵
        for(j=1;j<=n;j++)
            cin>>a[i][j];

    for(i=1;i<=n;i++)//输出转至后的矩阵
    {
        for(j=1;j<=m;j++)
            cout<<a[j][i]<<" ";
        cout<<endl;
    }
    return 0;
}
```

### 1.1.1.5.2.9 图像旋转（信息学奥赛一本通-T1127）

#### 图像旋转（信息学奥赛一本通-T1127）

原创 Alex\_McAvoy 发布于 2018-02-27 18:52:15 阅读数 3573 收藏  
展开

##### 【题目描述】

输入一个  $n$  行  $m$  列的黑白图像，将它顺时针旋转  $90$  度后输出。

##### 【输入】

第一行包含两个整数  $n$  和  $m$ ，表示图像包含像素点的行数和列数。 $1 \leq n \leq 100, 1 \leq m \leq 100$ 。

接下来  $n$  行，每行  $m$  个整数，表示图像的每个像素点灰度。相邻两个整数之间用单个空格隔开，每个元素均在  $0 \sim 255$  之间。

##### 【输出】

$m$  行，每行  $n$  个整数，为顺时针旋转  $90$  度后的图像。相邻两个整数之间用单个空格隔开。

##### 【输入样例】

```
3 3
1 2 3
4 5 6
7 8 9
```

##### 【输出样例】

```
7 4 1
8 5 2
9 6 3
```

##### 【源程序】

```
#include <iostream>
using namespace std;
int main()
{
    int m,n;
    int a[101][101];
    int i,j;

    cin>>n>>m;//输入矩阵列数、行数

    for(i=1;i<=n;i++)//输入矩阵
        for(j=1;j<=m;j++)
```

```

cin>>a[i][j];

for(i=1;i<=m;i++)//输出旋转后的矩阵
{
    for(j=n;j>=1;j--)
        cout<<a[j][i]<<" ";
    cout<<endl;
}
return 0;
}

```

### 1.1.1.5.2.10 图像模糊处理（信息学奥赛一本通-T1128）

图像模糊处理（信息学奥赛一本通-T1128）

原创 Alex\_McAvoy 发布于 2018-02-27 19:04:04 阅读数 4641 收藏  
展开

#### 【题目描述】

给定  $m$  行  $n$  列的图像各像素点的灰度值，要求用如下方法对其进行模糊化处理：

1.四周最外侧的像素点灰度值不变；

2.中间各像素点新灰度值为该像素点及其上下左右相邻四个像素点原灰度值的平均(舍入到最接近的整数)。

#### 【输入】

第一行包含两个整数  $n$  和  $m$ ，表示图像包含像素点的行数和列数。 $1 \leq n \leq 100, 1 \leq m \leq 100$ 。

接下来  $n$  行，每行  $m$  个整数，表示图像的每个像素点灰度。相邻两个整数之间用单个空格隔开，每个元素均在  $0 \sim 255$  之间。

#### 【输出】

$m$  行，每行  $n$  个整数，为模糊处理后的图像。相邻两个整数之间用单个空格隔开。

#### 【输入样例】

```

4 5
100 0 100 0 50
50 100 200 0 0
50 50 100 100 200
100 100 50 50 100

```

#### 【输出样例】

```

100 0 100 0 50
50 80 100 60 0

```



```
50 80 100 90 200
100 100 50 50 100
```

### 【源程序】

```
#include<iostream>
#include<cmath>
using namespace std;
int a[1001][1001];
double b[1001][1001];
int main()
{
    int m,n;

    int i,j;

    cin>>n>>m;//输入图像行数、列数

    for(i=1;i<=n;i++)//输入每个像素点灰度
        for(j=1;j<=m;j++)
            cin>>a[i][j];

    for(i=1;i<=n;i++)
    {
        for(j=1;j<=m;j++)
        {
            if(i==1||i==n||j==1||j==m)//边界灰度不变
                b[i][j]=a[i][j];
            else//内部按要求求平均
                b[i][j]=round((a[i][j]+a[i-1][j]+a[i+1][j]+a[i][j-1]+a[i][j+1])/5.0);//按要求用
round()四舍五入
        }
    }

    for(i=1;i<=n;i++)//输出模糊后的图像
    {
        for(j=1;j<=m;j++)
            cout<<b[i][j]<<" ";
        cout<<endl;
    }
    return 0;
}
```

### 1.1.1.5.3 第三节 字符类型和字符数组

#### 1.1.1.5.3.1 统计数字字符个数（信息学奥赛一本通-T1129）

##### 统计数字字符个数（信息学奥赛一本通-T1129）

原创 Alex\_McAvoy 发布于 2018-02-28 22:33:25 阅读数 6230 收藏  
展开

###### 【题目描述】

输入一行字符，统计出其中数字字符的个数。

###### 【输入】

一行字符串，总长度不超过 255。

###### 【输出】

输出为 1 行，输出字符串里面数字字符的个数。

###### 【输入样例】

Peking University is set up at 1898.

###### 【输出样例】

4

###### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char ch[256];
    int len;
    int sum=0;
    int i;

    gets(ch);//获取一行字符串
    len=strlen(ch);//求字符串长度

    for(i=0;i<=len;i++)//对字符串进行遍历
        if(ch[i]>='0'&&ch[i]<='9')//如果是数字
```

```

sum++;//计数器+1

cout<<sum<<endl;//输入统计的个数

return 0;
}

```

### 1.1.1.5.3.2 找第一个只出现一次的字符（信息学奥赛一本通-T1130）

#### 找第一个只出现一次的字符（信息学奥赛一本通-T1130）

原创 Alex\_McAvoy 发布于 2018-02-28 22:47:10 阅读数 10398 收藏  
展开

##### 【题目描述】

给定一个只包含小写字母的字符串，请你找到第一个仅出现一次的字符。如果没有，输出 no。

##### 【输入】

一个字符串，长度小于 100000。

##### 【输出】

输出第一个仅出现一次的字符，若没有则输出 no。

##### 【输入样例】

abcabd

##### 【输出样例】

c

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
char ch[100001];
int main()
{
    int letters[26]= {0};
    int len;
    int sum=0;
    int i;

```

```

gets(ch);//获取一行字符串
len=strlen(ch);//求字符串长度

for(i=0; i<len; i++) //桶排的思想统计 26 个小写字母出现频次
    letters[ch[i]-'a']++;

for(i=0; i<len; i++) //对字符串进行遍历
{
    if(letters[ch[i]-'a']==1)//如果出现过一次
    {
        cout<<ch[i]<<endl;//输出字母
        return 0;//结束
    }
}
cout<<"no"<<endl;
return 0;
}

```

### 1.1.1.5.3.3 基因相关性（信息学奥赛一本通-T1131）

#### 基因相关性（信息学奥赛一本通-T1131）

原创 Alex\_McAvoy 发布于 2018-03-03 16:48:37 阅读数 4813 收藏  
展开

##### 【题目描述】

为了获知基因序列在功能和结构上的相似性，经常需要将几条不同序列的 DNA 进行比对，以判断该比对的 DNA 是否具有相关性。

现比对两条长度相同的 DNA 序列。定义两条 DNA 序列相同位置的碱基为一个碱基对，如果一个碱基对中的两个碱基相同的话，则称为相同碱基对。接着计算相同碱基对占总碱基对数量的比例，如果该比例大于等于给定阈值时则判定该两条 DNA 序列是相关的，否则不相关。

##### 【输入】

有三行，第一行是用来判定出两条 DNA 序列是否相关的阈值，随后 2 行是两条 DNA 序列(长度不大于 500)。

##### 【输出】

若两条 DNA 序列相关，则输出“yes”，否则输出“no”。

##### 【输入样例】

0.85

ATCGCCGTAAGTAACGGTTTTAAATAGGCC  
 ATCGCCGGAAGTAACGGTCTTAAATAGGCC

【输出样例】

yes

【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    char s1[501],s2[501];
    int len;
    double x,sum=0;
    int i;

    cin>>x>>s1>>s2;
    len=strlen(s1);//计算字符串长度
    for(i=1; i<=len; i++) //从头到尾枚举串 1 与串 2 是否相同
        if(s1[i]==s2[i])
            sum++;//统计相同个数

    if(sum/len>=x)//所占比例大于所给阈值
        cout<<"yes"<<endl;//相关
    else//否则
        cout<<"no"<<endl;//不相关

    return 0;
}
```

点赞

### 1.1.1.5.3.4 石头剪子布（信息学奥赛一本通-T1132）

#### 石头剪子布（信息学奥赛一本通-T1132）

原创 Alex\_McAvoy 发布于 2018-03-03 17:04:07 阅读数 5759 收藏  
 展开

【题目描述】

石头剪子布，是一种猜拳游戏。起源于中国，然后传到日本、朝鲜等地，随着亚欧贸易的不断发展它传到了欧洲，到了近现代逐渐风靡世界。简单明了的规则，使得石头

剪子布没有任何规则漏洞可钻，单次玩法比拼运气，多回合玩法比拼心理博弈，使得石头剪子布这个古老的游戏同时用于“意外”与“技术”两种特性，深受世界人民喜爱。

游戏规则：石头打剪刀，布包石头，剪刀剪布。

现在，需要你写一个程序来判断石头剪子布游戏的结果。

#### 【输入】

第一行是一个整数  $N$ ，表示一共进行了  $N$  次游戏。  $1 \leq N \leq 100$ 。

接下来  $N$  行的每一行包括两个字符串，表示游戏参与者 **Player1**，**Player2** 的选择（石头、剪子或者是布）：**S1 S2**

字符串之间以空格隔开 **S1,S2** 只可能取值在{"Rock", "Scissors", "Paper"}(大小写敏感)中。

#### 【输出】

输出包括  $N$  行，每一行对应一个胜利者（**Player1** 或者 **Player2**），或者游戏出现平局，则输出 **Tie**。

#### 【输入样例】

```
3
Rock Scissors
Paper Paper
Rock Paper
```

#### 【输出样例】

```
Player1
Tie
Player2
```

#### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    char a[101],b[101];
    int n;
    int i;

    cin>>n;

    for(i=0; i<n; i++)
    {
        cin>>a>>b;
        if(a[0]=='R'&&b[0]=='S'||a[0]=='S'&&b[0]=='P'||a[0]=='P'&&b[0]=='R')//判断第一
```

个字母即可

```

        cout<<"Player1"<<endl;
    else if(a[0]==b[0])
        cout<<"Tie"<<endl;
    else
        cout<<"Player2"<<endl;
    }

    return 0;
}

```

点赞

### 1.1.1.5.3.5 输出亲朋字符串（信息学奥赛一本通-T1133）

#### 输出亲朋字符串（信息学奥赛一本通-T1133）

原创 Alex\_McAvoy 发布于 2018-03-03 17:16:45 阅读数 3564 收藏  
展开

##### 【题目描述】

编写程序，求给定字符串 **s** 的亲朋字符串 **s1**。

亲朋字符串 **s1** 定义如下：给定字符串 **s** 的第一个字符的 ASCII 值加第二个字符的 ASCII 值，得到第一个亲朋字符；给定字符串 **s** 的第二个字符的 ASCII 值加第三个字符的 ASCII 值，得到第二个亲朋字符；依此类推，直到给定字符串 **s** 的倒数第二个字符。亲朋字符串的最后一个字符由给定字符串 **s** 的最后一个字符 ASCII 值加 **s** 的第一个字符的 ASCII 值。

##### 【输入】

输入一行，一个长度大于等于 2，小于等于 100 的字符串。字符串中每个字符的 ASCII 值不大于 63。

##### 【输出】

输出一行，为变换后的亲朋字符串。输入保证变换后的字符串只有一行。

##### 【输入样例】

1234

##### 【输出样例】

cege

##### 【源程序】

```

#include<iostream>
#include<cstdio>

```

```

#include<cstring>
using namespace std;
int main()
{
    char a[101];
    int len;
    int i;

    gets(a);
    len=strlen(a);//计算字符串长度

    for(i=0; i<=len-2; i++) //除最后一位，其他依次变换输出
        cout<<char(a[i]+a[i+1]);
    cout<<char(a[len-1]+a[0])<<endl;//输出变化后的最后一位
    return 0;
}

```

点赞 2

### 1.1.1.5.3.6 合法 C 标识符（信息学奥赛一本通-T1134）

#### 合法 C 标识符（信息学奥赛一本通-T1134）

原创 Alex\_McAvoy 发布于 2018-03-03 17:29:51 阅读数 4102 收藏  
展开

##### 【题目描述】

给定一个不包含空白符的字符串，请判断是否是 C 语言合法的标识符号(注：题目保证这些字符串一定不是 C 语言的保留字)。

C 语言标识符要求：

- 1.非保留字；
- 2.只包含字母、数字及下划线（“\_”）。
- 3.不以数字开头。

##### 【输入】

一行，包含一个字符串，字符串中不包含任何空白字符，且长度不大于 20。

##### 【输出】

一行，如果它是 C 语言的合法标识符，则输出 **yes**，否则输出 **no**。



**【输入样例】**

RKPEGX9R;TWyYcp

**【输出样例】**

no

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char a[256];
    int len;
    int i;

    gets(a);
    len=strlen(a);//计算字符串长度

    if(a[0]>='0'&&a[0]<='9')
    {
        cout<<"no"<<endl;//开头是数字，不满足条件
        return 0;
    }
    for(i=0; i<len; i++) //若不含包含字母、数字、下划线，不满足条件
    {
        if(!( (a[i]>='0'&&a[i]<='9') || (a[i]>='A'&&a[i]<='Z') || (a[i]>='a'&&a[i]<='z') ||
a[i]=='_' ) )
        {
            cout<<"no"<<endl;
            return 0;
        }
    }
    cout<<"yes"<<endl;
    return 0;
}

```

点赞 2

**1.1.1.5.3.7 配对碱基链（信息学奥赛一本通-T1135）****配对碱基链（信息学奥赛一本通-T1135）**

原创 Alex\_McAvoy 发布于 2018-03-03 17:40:40 阅读数 3102 收藏  
展开

### 【题目描述】

脱氧核糖核酸(DNA)由两条互补的碱基链以双螺旋的方式结合而成。而构成 DNA 的碱基共有 4 种，分别为腺嘌呤(A)、鸟嘌呤(G)、胸腺嘧啶(T)和胞嘧啶(C)。我们知道，在两条互补碱基链的对应位置上，腺嘌呤总是和胸腺嘧啶配对，鸟嘌呤总是和胞嘧啶配对。你的任务就是根据一条单链上的碱基序列，给出对应的互补链上的碱基序列。

### 【输入】

一个字符串，表示一条碱基链。这个字符串只含有大写字母 A、T、G、C，分别表示腺嘌呤、胸腺嘧啶、鸟嘌呤和胞嘧啶。字符串长度不超过 255。

### 【输出】

一个只含有大写字母 A、T、G、C 的字符串，为与输入的碱基链互补的碱基链。

### 【输入样例】

ATATGGATGGTGTGTTGGCTCTG

### 【输出样例】

TATACCTACCACAAACCGAGAC

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char a[256];
    int len;
    int i;

    gets(a);
    len=strlen(a);//计算字符串长度
    for(i=0; i<len; i++) //输出配对碱基
    {
        if(a[i]=='A') cout<<"T";
        if(a[i]=='T') cout<<"A";
        if(a[i]=='G') cout<<"C";
        if(a[i]=='C') cout<<"G";
    }
    cout<<endl;
```

```
    return 0;
}
```

### 1.1.1.5.3.8 密码翻译（信息学奥赛一本通-T1136）

#### 密码翻译（信息学奥赛一本通-T1136）

原创 Alex\_McAvoy 发布于 2018-03-03 19:51:24 阅读数 4252 收藏  
展开

##### 【题目描述】

在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。我们给出一种最简的加密方法，对给定的一个字符串，把其中从 **a-y**，**A-Y** 的字母用其后继字母替代，把 **z** 和 **Z** 用 **a** 和 **A** 替代，其他非字母字符不变，则可得到一个简单的加密字符串。

##### 【输入】

输入一行，包含一个字符串，长度小于 80 个字符。

##### 【输出】

输出每行字符串的加密字符串。

##### 【输入样例】

Hello! How are you!

##### 【输出样例】

Ifmmp! lpx bsf zpv!

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char a[256];
    int len;
    int i;
    gets(a);
    len=strlen(a);//计算字符串长度
    for(i=0;i<len;i++) //按规则进行加密
```

```

{
    if(a[i]=='Z' || a[i]=='z')
        a[i]-=25;
    else if(a[i]>='A' && a[i]<'Z')
        a[i]+=1;
    else if(a[i]>='a' && a[i]<'z')
        a[i]+=1;
}
for(int i=0; i<len; i++)
    cout<<a[i];
cout<<endl;
return 0;
}

```

### 1.1.1.5.3.9 加密的病历单（信息学奥赛一本通-T1137）

#### 加密的病历单（信息学奥赛一本通-T1137）

原创 Alex\_McAvoy 发布于 2018-03-03 18:19:32 阅读数 4781 收藏  
展开

##### 【题目描述】

小英是药学专业大三的学生，暑假期间获得了去医院药房实习的机会。

在药房实习期间，小英扎实的专业基础获得了医生的一致好评，得知小英在计算概论中取得过好成绩后，主任又额外交给她一项任务，解密抗战时期被加密过的一些伤员的名单。

经过研究，小英发现了如下加密规律(括号中是一个“原文 -> 密文”的例子)

- 1.原文中所有的字符都在字母表中被循环左移了三个位置 (dec -> abz)
- 2.逆序存储 (abcd -> dcba )
- 3.大小写反转 (abXY -> ABxy)

##### 【输入】

一个加密的字符串。(长度小于 50 且只包含大小写字母)

##### 【输出】

输出解密后的字符串。

##### 【输入样例】

GSOOWFASOq

## 【输出样例】

Trvdizrrvj

## 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char a[256],temp;
    int len;
    int i;

    gets(a);
    len=strlen(a);//计算字符串长度

    for(i=0; i<len; i++) //按规则进行加密
    {
        if(a[i]>='a'&&a[i]<='z')//小写的情况
        {
            temp=a[i];
            temp+=3;//向右移动 3 个字符
            if(temp>122) temp-=26;//若为 x, y, z, 转换为 a, b, c
            temp-=32;
            a[i]=temp;//小写转换为大写
        }
        elseif(a[i]>='A'&&a[i]<='Z')//大写的情况
        {
            temp=a[i];
            temp+=3;//向右移动 3 个字符
            if(temp>90) temp-=26;//若为 X, Y, Z, 转换为 A, B, C
            temp+=32;
            a[i]=temp;//大写转换为小写
        }
    }

    for(i=len-1; i>=0; i--) //逆序输出
        cout<<a[i];
    cout<<endl;

    return 0;
}
```

}

点赞

### 1.1.1.5.3.10 将字符串中的小写字母转换成大写字母 （信息学奥赛一本通-T1138）

#### 将字符串中的小写字母转换成大写字母 （信息学奥赛一本通-T1138）

原创 Alex\_McAvoy 发布于 2018-03-03 18:31:34 阅读数 4416 收藏  
展开

##### 【题目描述】

给定一个字符串，将其中所有的小写字母转换成大写字母。

##### 【输入】

输入一行，包含一个字符串（长度不超过 100，可能包含空格）。

##### 【输出】

输出转换后的字符串。

##### 【输入样例】

helloworld123Ha

##### 【输出样例】

HELLOWORLD123HA

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char a[256];
    int len;
    int i;

    gets(a);
    len=strlen(a);//计算字符串长度

    for(i=0; i<len; i++) //小写转换为大写
        if(a[i]>='a'&&a[i]<='z')
            a[i]-=32;
```

```
puts(a);

return 0;
}
```

点赞 1

### 1.1.1.5.3.11 整理药名（信息学奥赛一本通-T1139）

#### 整理药名（信息学奥赛一本通-T1139）

原创 Alex\_McAvoy 发布于 2018-03-03 19:44:56 阅读数 4183 收藏  
展开

##### 【题目描述】

医生在书写药品名的时候经常不注意大小写，格式比较混乱。现要求你写一个程序将医生书写混乱的药品名整理成统一规范的格式，即药品名的第一个字符如果是字母要大写，其他字母小写。如将 **ASPIRIN**、**aspirin** 整理成 **Aspirin**。

##### 【输入】

第一行一个数字  $n$ ，表示有  $n$  个药品名要整理， $n$  不超过 100。

接下来  $n$  行，每行一个单词，长度不超过 20，表示医生手书的药品名。药品名由字母、数字和-组成。

##### 【输出】

$n$  行，每行一个单词，对应输入的药品名的规范写法。

##### 【输入样例】

```
4
AspiRin
cisapride
2-PENICILLIN
Cefradine-6
```

##### 【输出样例】

```
Aspirin
Cisapride
2-penicillin
Cefradine-6
```

##### 【源程序】

```
#include<iostream>
```

```

#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    int n;
    char str[25];
    int len;
    int i,j;

    cin>>n;
    for(i=0; i<n; i++)
    {
        scanf("%s",str);
        if(str[0]>='a'&&str[0]<='z')//以小写开头，转为大写
            str[0]-=32;
        len=strlen(str);
        for(j=1; j<len; j++) //除开头外均转为小写字母
            if(str[j]>='A'&&str[j]<='Z')
                str[j]+=32;
        puts(str);
    }
    return 0;
}

```

### 1.1.1.5.3.12 验证子串（信息学奥赛一本通-T1140）

#### 验证子串（信息学奥赛一本通-T1140）

原创 Alex\_McAvoy 发布于 2018-03-04 18:58:13 阅读数 3822 收藏  
展开

##### 【题目描述】

输入两个字符串，验证其中一个串是否为另一个串的子串。

##### 【输入】

输入两个字符串， 每个字符串占一行，长度不超过 200 且不含空格。

##### 【输出】

若第一个串 **s1** 是第二个串 **s2** 的子串，则输出(s1) is substring of (s2)

否则，若第二个串 **s2** 是第一个串 **s1** 的子串，输出(s2) is substring of (s1)

否则，输出 No substring。



**【输入样例】**

abc  
dddncabca

**【输出样例】**

abc is substring of dddncabca

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s1[200],s2[200];

    gets(s1);//获取字符串 s1
    gets(s2);//获取字符串 s2

    if(strstr(s2,s1))//判断 s1 是否是 s2 的子串
    {
        cout<<s1<<" is substring of "<<s2<<endl;
        return 0;
    }
    if(strstr(s1,s2))//判断 s2 是否是 s1 的子串
    {
        cout<<s2<<" is substring of "<<s1<<endl;
        return 0;
    }
    cout<<"No substring"<<endl;
    return 0;
}
```

点赞 1

### 1.1.1.5.3.13 删除单词后缀（信息学奥赛一本通-T1141）

#### 删除单词后缀（信息学奥赛一本通-T1141）

原创 Alex\_McAvoy 发布于 2018-03-04 19:04:46 阅读数 3299 收藏  
展开

**【题目描述】**

给定一个单词，如果该单词以 **er**、**ly** 或者 **ing** 后缀结尾，则删除该后缀（题目保证删除后缀后的单词长度不为 0），否则不进行任何操作。

**【输入】**

输入一行，包含一个单词（单词中间没有空格，每个单词最大长度为 32）。

**【输出】**

输出按照题目要求处理后的单词。

**【输入样例】**

referer

**【输出样例】**

refer

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[200];
    int len;
    int i;

    gets(s);//获取字符串 s
    len=strlen(s);

    if(s[len-2]=='e'&&s[len-1]=='r')//判断以 er 结尾的情况
    {
        s[len-2]='\0';
        s[len-1]='\0';
    }

    if(s[len-2]=='l'&&s[len-1]=='y')//判断以 ly 结尾的情况
    {
        s[len-2]='\0';
        s[len-1]='\0';
    }

    if(s[len-3]=='i'&&s[len-2]=='n'&&s[len-1]=='g')//判断以 ing 结尾的情况
    {
        s[len-3]='\0';
        s[len-2]='\0';
        s[len-1]='\0';
    }
}
```

```

    }
    puts(s);
    return 0;
}

```

### 1.1.1.5.3.14 单词的长度（信息学奥赛一本通-T1142）

#### 单词的长度（信息学奥赛一本通-T1142）

原创 Alex\_McAvoy 发布于 2018-03-04 19:11:07 阅读数 4222 收藏  
展开

##### 【题目描述】

输入一行单词序列，相邻单词之间由 1 个或多个空格间隔，请对应地计算各个单词的长度。

注意:如果有标点符号(如连字符，逗号)，标点符号算作与之相连的词的一部分。没有被空格间开的符号串，都算作单词。

##### 【输入】

一行单词序列，最少 1 个单词，最多 300 个单词，单词之间用至少 1 个空格间隔。单词序列总长度不超过 1000。

##### 【输出】

依次输出对应单词的长度，之间以逗号间隔。

##### 【输入样例】

She was born in 1990-01-02 and from Beijing city.

##### 【输出样例】

3,3,4,2,10,3,4,7,5

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[1000];
    int len;
    int sum=0;
    int i;

```

```

gets(s);//获取字符串 s
len=strlen(s);//求字符串长度

for(i=0; i<len; i++)
{
    if(s[i]!=' ') sum++;//若不是空格，累加单词长度
    else//若是空格，输出目前已累加的单词长度
    {
        if(sum>0) cout<<sum<<" ";
        sum=0;//输出完后计数器归零
    }
}
cout<<sum<<endl;//最后一个单词没有空格，在此直接输出最后一个单词长度

return 0;
}

```

### 1.1.1.5.3.15 最长最短单词（信息学奥赛一本通-T1143）

#### 最长最短单词（信息学奥赛一本通-T1143）

原创 Alex\_McAvoy 发布于 2018-03-04 19:23:25 阅读数 5095 收藏  
展开

##### 【题目描述】

输入 1 行句子(不多于 200 个单词，每个单词长度不超过 100)，只包含字母、空格和逗号。单词由至少一个连续的字母构成，空格和逗号都是单词间的间隔。

试输出第 1 个最长的单词和第 1 个最短单词。

##### 【输入】

一行句子。

##### 【输出】

第 1 行，第一个最长的单词。

第 2 行，第一个最短的单词。

##### 【输入样例】

I am studying Programming language C in Peking University

##### 【输出样例】

Programming

I

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[10000];
    int len;
    int sum=0,max_len=0,min_len=9999;
    int max_position,min_position;
    int i;

    gets(s);//获取字符串 s
    len=strlen(s);//求字符串长度
    s[len]=' ';//在字符串后加以空格，方便判定

    for(i=0; i<len; i++)
    {
        if(s[i]!=' ' && s[i]!='.') sum++; //若不是空格或逗号，累加单词长度
        else if(sum>0) //若是空格或逗号
        {
            if(sum>max_len)
            {
                max_len=sum; //存储目前最长的单词长度
                max_position=i-sum; //记录最长单词的位置
            }
            if(sum<min_len)
            {
                min_len=sum; //存储目前最短的单词长度
                min_position=i-sum; //记录最短单词的位置
            }
            sum=0; //计数器归零
        }
    }

    for(i=max_position; i<=max_position+max_len-1; i++) //输出最长单词
        cout<<s[i];
    cout<<endl;
    for(i=min_position; i<=min_position+min_len-1; i++) //输出最短单词
        cout<<s[i];

    return 0;
}

```

### 1.1.1.5.3.16 单词翻转（信息学奥赛一本通-T1144）

#### 单词翻转（信息学奥赛一本通-T1144）

原创 Alex\_McAvoy 发布于 2018-03-04 19:54:12 阅读数 4207 收藏  
展开

##### 【题目描述】

输入一个句子(一行)，将句子中的每一个单词翻转后输出。

##### 【输入】

只有一行，为一个字符串，不超过 500 个字符。单词之间以空格隔开。

##### 【输出】

翻转每一个单词后的字符串，单词之间的空格需与原文一致。

##### 【输入样例】

hello world

##### 【输出样例】

olleh dlrow

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[1000];
    int len,position;
    int sum=0;
    int i,j;

    gets(s);//获取字符串 s
    len=strlen(s);//求字符串长度
    s[len]=' ';//末尾加一空格方便计算

    for(i=0; i<=len; i++)
    {
        if(s[i]!=' ')
            sum++;//计算单词长度
        else
        {
```

```

    position=i;//记录单词末位置
    for(j=1; j<=sum; j++) //倒序输出
        cout<<s[--position];
    sum=0;//计数器归零
    if(i!=len)//若不是自己加上的空格就输出空格
        cout<<" ";
    }
}
cout<<endl;
return 0;
}

```

### 1.1.1.5.3.17 字符串 p 型编码（信息学奥赛一本通-T1145）

#### 字符串 p 型编码（信息学奥赛一本通-T1145）

原创 Alex\_McAvoy 发布于 2018-03-04 20:00:26 阅读数 2671 收藏  
展开

##### 【题目描述】

给定一个完全由数字字符（'0','1','2',..., '9'）构成的字符串 **str**，请写出 **str** 的 p 型编码串。例如：字符串 122344111 可被描述为"1 个 1、2 个 2、1 个 3、2 个 4、3 个 1"，因此我们说 122344111 的 p 型编码串为 1122132431；类似的道理，编码串 101 可以用来描述 1111111111；0000000000 可描述为"11 个 0"，因此它的 p 型编码串即为 110；100200300 可描述为"1 个 1、2 个 0、1 个 2、2 个 0、1 个 3、2 个 0"，因此它的 p 型编码串为 112012201320。

##### 【输入】

输入仅一行，包含字符串 **str**。每一行字符串最多包含 1000 个数字字符。

##### 【输出】

输出该字符串对应的 p 型编码串。

##### 【输入样例】

122344111

##### 【输出样例】

1122132431

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;

```

```

int main()
{
    char s[1000];
    int len,position;
    int sum=1;
    int i;

    gets(s);//获取字符串 s
    len=strlen(s);//求字符串长度

    for(i=0; i<len; i++)
    {
        if(s[i]==s[i+1])
            sum++;//若第 i 个与第 i+1 个数字相同，累加连续出现次数
        else
        {
            cout<<sum<<s[i];//无法连续时，输出第 i 个数字连续出现次数与该数字
            sum=1;//开始累加第 i+1 个数字的连续出现次数
        }
    }
    cout<<endl;
    return 0;
}

```

点赞

### 1.1.1.5.3.18 判断字符串是否为回文（信息学奥赛一本通-T1146）

#### 判断字符串是否为回文（信息学奥赛一本通-T1146）

原创 Alex\_McAvoy 发布于 2018-03-04 20:05:22 阅读数 3327 收藏  
展开

##### 【题目描述】

输入一个字符串，输出该字符串是否回文。回文是指顺读和倒读都一样的字符串。

##### 【输入】

输入为一行字符串（字符串中没有空白字符，字符串长度不超过 100）。

##### 【输出】

如果字符串是回文，输出 **yes**；否则，输出 **no**。



**【输入样例】**

abcdedcba

**【输出样例】**

yes

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[100];
    int len,position;
    int i,j;

    gets(s);//获取字符串 s
    len=strlen(s);//求字符串长度
    i=0;
    j=len-1;//记录字符串首、尾位置

    while( (i<j) &&(s[i]==s[j]) )//从首尾同时向中间判定，若不是回文串，则退出循环
    {
        i++;
        j--;
    }

    if(i>=j)
        cout<<"yes"<<endl;
    else
        cout<<"no"<<endl;

    return 0;
}
```

点赞 1

### 1.1.1.5.3.19 最高分数的学生姓名（信息学奥赛一本通-T1147）

最高分数的学生姓名（信息学奥赛一本通-T1147）

原创 Alex\_McAvoy 发布于 2018-03-04 20:35:00 阅读数 4501 收藏  
展开

### 【题目描述】

输入学生的人数，然后再输入每位学生的分数和姓名，求获得最高分数的学生的姓名。

### 【输入】

第一行输入一个正整数  $N$  ( $N \leq 100$ )，表示学生人数。接着输入  $N$  行，每行格式：分数 姓名

分数是一个非负整数，且小于等于 100；

姓名为一个连续的字符串，中间没有空格，长度不超过 20。

数据保证最高分只有一位同学。

### 【输出】

获得最高分数同学的姓名。

### 【输入样例】

```
5
87 lilei
99 hanmeimei
97 lily
96 lucy
77 jim
```

### 【输出样例】

```
hanmeimei
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<string>
using namespace std;

struct student
{
    string name;
    int score;
} a[100];

int main()
{
    int n;
```

```

int i,j;

cin>>n;//输入学生数
for(i=0; i<n; i++)
{
    cin>>a[i].score;//输入分数
    cin>>a[i].name;//输入姓名
}

for(i=0; i<n-1; i++) //冒泡排序
    for(j=0; j<n-1-i; j++)
        if(a[j].score<a[j+1].score)
            swap(a[j],a[j+1]);

cout<<a[0].name<<endl;
return 0;
}

```

### 1.1.1.5.3.20 连续出现的字符（信息学奥赛一本通-T1148）

#### 连续出现的字符（信息学奥赛一本通-T1148）

原创 Alex\_McAvoy 发布于 2018-03-04 20:45:14 阅读数 3428 收藏  
展开

##### 【题目描述】

给定一个字符串，在字符串中找到第一个连续出现至少  $k$  次的字符。

##### 【输入】

第一行包含一个正整数  $k$ ，表示至少需要连续出现的次数。 $1 \leq k \leq 1000$ 。

第二行包含需要查找的字符串。字符串长度在 1 到 2500 之间，且不包含任何空白符。

##### 【输出】

若存在连续出现至少  $k$  次的字符，输出该字符；否则输出 No。

##### 【输入样例】

```

3
abcccaaab

```

##### 【输出样例】

```

c

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[2500];
    int k;
    int len;
    int sum=0;
    int i;

    cin>>k;//输入要找的次数
    cin>>s;//读入字符串
    len=strlen(s);//求字符串长

    for(i=0; i<len; i++)
    {
        if(sum==k)//若次数已达题目要求
        {
            cout<<s[i]<<endl;//输出
            return 0;//结束
        }
        if(s[i]==s[i+1])//若下一项与该项相同
            sum++;//次数加 1
        else//否则，次数归 1
            sum=1;
    }
    cout<<"No"<<endl;
    return 0;
}

```

**1.1.1.5.3.21 最长单词（信息学奥赛一本通-T1149）****最长单词（信息学奥赛一本通-T1149）**

原创 Alex\_McAvoy 发布于 2018-03-04 20:59:11 阅读数 3484 收藏  
展开

**【题目描述】**

一个以‘.’结尾的简单英文句子，单词之间用空格分隔，没有缩写形式和其它特殊形式。

**【输入】**

一个以‘.’结尾的简单英文句子（长度不超过 500），单词之间用空格分隔，没有缩写形式和其它特殊形式。

**【输出】**

该句子中最长的单词。如果多于一个，则输出第一个。

**【输入样例】**

I am a student of Peking University.

**【输出样例】**

University

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    char s[2500];
    int k;
    int len;
    int max_len=0,max_position;
    int sum=0;
    int i;

    gets(s);//读入字符串
    len=strlen(s)-1;//求字符串长

    for(i=0; i<=len; i++)
    {
        if(s[i]!=' '&&s[i]!='.') sum++;//若不是空格或句号，累加长度
        else if(sum>0)
        {
            if(sum>max_len)//若当前单词长度大于最长单词
            {
                max_len=sum;//替换最长单词长度
                max_position=i-sum;//记录最长单词起始位置
            }
            sum=0;//长度清零
        }
    }
    for(i=max_position; i<=max_len+max_position-1; i++)
        cout<<s[i];
```

```
cout<<endl;
return 0;
```

## 1.1.1.6 第六章 函数

### 1.1.1.6.1 第一节 函数

#### 1.1.1.6.1.1 求正整数 2 和 n 之间的完全数（信息学奥赛一本通-T1150）

#### 求正整数 2 和 n 之间的完全数（信息学奥赛一本通-T1150）

原创 Alex\_McAvoy 发布于 2018-03-05 21:50:25 阅读数 6288 收藏  
展开

##### 【题目描述】

求正整数 2 和 n 之间的完全数(一行一个数)。

完全数：因子之和等于它本身的自然数，如  $6=1+2+3$

##### 【输入】

输入 n。

##### 【输出】

一行一个数,按由小到大的顺序。

##### 【输入样例】

7

##### 【输出样例】

6

##### 【源程序】

```
#include<iostream>
using namespace std;
int judge(int x);
int main()
```

```

{
    int n;
    int i;

    cin>>n;
    for(i=2; i<=n; i++)
        if(judge(i)==i)//judge(i)为 i 的因子之和, judge(i)若与 i 相等, 满足条件, 输出
            cout<<i<<endl;
    return 0;
}
int judge(int x)
{
    int i;
    int sum=0;
    for(i=1; i<=x-1; i++) //枚举 1 到 x 中, 判断是否为 x 的因子
        if(x%i==0)//若是 x 的因子, 累加
            sum+=i;
    return sum;
}

```

点赞 2

### 1.1.1.6.1.2 素数个数（信息学奥赛一本通-T1151）

#### 素数个数（信息学奥赛一本通-T1151）

原创 Alex\_McAvoy 发布于 2018-03-05 22:00:35 阅读数 5569 收藏  
展开

##### 【题目描述】

编程求  $2 \sim n$  ( $n$  为大于 2 的正整数)中有多少个素数。

##### 【输入】

输入  $n(2 \leq n \leq 50000)$ 。

##### 【输出】

素数个数。

##### 【输入样例】

10

##### 【输出样例】

4

## 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);
int main()
{
    int n;
    int i;
    int sum=0;

    cin>>n;
    for(i=2; i<=n; i++)
        if(judge(i))//若是素数
            sum++;//累加素数个数
    cout<<sum<<endl;

    return 0;
}
bool judge(int x)//判断素数
{
    int i=2;
    while(i<=floor(sqrt(x))&&(x%i)!=0)
        i++;
    if(i>floor(sqrt(x)))
        return true;
    return false;
}

```

点赞 1

### 1.1.1.6.1.3 最大数 $\max(x,y,z)$ （信息学奥赛一本通-T1152）

#### 最大数 $\max(x,y,z)$ （信息学奥赛一本通-T1152）

原创 Alex\_McAvoy 发布于 2018-03-05 22:13:34 阅读数 4462 收藏  
展开

## 【题目描述】

已知:



$m = \max(a, b, c) \max(a+b, b, c) \times \max(a, b, b+c)$   
 $m = \max(a, b, c) \max(a+b, b, c) \times \max(a, b, b+c)$   
 )

输入  $a, b, c$ ，求  $m$ 。把求三个数的最大数  $\max(x, y, z)$  分别定义成函数和过程来做。

**【输入】**

输入  $a, b, c$ 。

**【输出】**

求  $m$ ，保留到小数点后三位。

**【输入样例】**

1 2 3

**【输出样例】**

0.200

**【源程序】**

```
#include<iostream>
#include<cstdio>
using namespace std;
int max(int a,int b,int c);

int main()
{
    int a,b,c;
    double m;

    cin>>a>>b>>c;//输入 a、b、c
    m=1.0*max(a,b,c)/(max(a+b,b,c)*max(a,b,b+c));//计算 m
    printf("%.3lf\n",m);
    return 0;
}

int max(int a,int b,int c)//求三个数中的最大值
{
    int temp=a;
    if(a<b)
        temp=b;
    if(temp<c)
        temp=c;
    return temp;
}
```

### 1.1.1.6.1.4 绝对素数（信息学奥赛一本通-T1153）

#### 绝对素数（信息学奥赛一本通-T1153）

原创 Alex\_McAvoy 发布于 2018-03-05 22:20:29 阅读数 4437 收藏  
展开

##### 【题目描述】

如果一个自然数是素数,且它的数字位置经过对换后仍为素数,则称为绝对素数,例如13。试求出所有二位绝对素数。

##### 【输入】

(无)

##### 【输出】

所有二位绝对素数（由小到大，一个数一行）。

##### 【输入样例】

(无)

##### 【输出样例】

(无)

##### 【源程序】

```
#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int i;

    for(i=10; i<=99; i++) //从 10 枚举到 99
        if( judge(i) && judge((i%10)*10+(i/10)) )//分别判断 i 以及 i 变换后是否是素数
            cout<<i<<endl;
    return 0;
}

bool judge(int x)
{
    int i=2;
    while( i<=floor(sqrt(x)) && (x%i!=0) )
        i++;
}
```

```

    if(i>floor(sqrt(x)))
        return true;
    return false;
}

```

### 1.1.1.6.1.5 亲和数（信息学奥赛一本通-T1154）

#### 亲和数（信息学奥赛一本通-T1154）

原创 Alex\_McAvoy 发布于 2018-03-06 16:30:39 阅读数 4403 收藏  
展开

##### 【题目描述】

自然数  $a$  的因子是指能整除  $a$  的所有自然数，但不含  $a$  本身。例如 12 的因子为：1,2,3,4,6。若自然数  $a$  的因子之和为  $b$ ，而且  $b$  的因子之和又等于  $a$ ，则称  $a,b$  为一对“亲和数”。求最小的一对亲和数( $a < b$ )。

##### 【输入】

(无)

##### 【输出】

1 行，分别为  $a$  和  $b(a < b)$ 。

##### 【输入样例】

(无)

##### 【输出样例】

(无)

##### 【源程序】

```

#include<iostream>
using namespace std;
int judge(int n);

int main()
{
    long int i;

    for(i=1;; i++) //从 1 开始枚举
    {
        if( i==judge(judge(i)) && i!=judge(i) )//num(num(i))即为 i 因子之和的因子之和
        {
            cout<<i<<" "<<judge(i)<<endl;//i 与 judge(i)即满足条件的一组亲和数
            break;//退出
        }
    }
}

```

```

    }
}
return 0;
}

int judge(int n)
{
    long int sum=0;
    long int i;

    for(i=1; i<n/2+1; i++)
        if(n%i==0)
            sum+=i;
    return sum;
}

```

点赞

### 1.1.1.6.1.6 回文三位数（信息学奥赛一本通-T1155）

#### 回文三位数（信息学奥赛一本通-T1155）

原创 Alex\_McAvoy 发布于 2018-03-06 16:36:09 阅读数 3621 收藏  
展开

##### 【题目描述】

如果一个数从左边读和从右边读都是同一个数，就称为回文数。例如 6886 就是一个回文数，求出所有的既是回文数又是素数的三位数。

##### 【输入】

（无）

##### 【输出】

所有的既是回文数又是素数的三位数。一个数一行。

##### 【输入样例】

（无）

##### 【输出样例】

（无）

##### 【源程序】

```
#include<iostream>
```

```

#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int i;
    for(i=100; i<=999; i++) //从枚举所有三位数
        if( judge(i) && (i/100==i%10))//该数是素数且百位与个位相同
            cout<<i<<endl;//输出该数
    return 0;
}

bool judge(int x)//判断素数
{
    int i=2;
    while( i<=floor(sqrt(x)) && (x%i!=0) )
        i++;
    if(i>floor(sqrt(x)))
        return true;
    return false;
}

```

### 1.1.1.6.1.7 求 $\pi$ 的值（信息学奥赛一本通-T1156）

#### 求 $\pi$ 的值（信息学奥赛一本通-T1156）

原创 Alex\_McAvoy 发布于 2018-03-08 15:03:33 阅读数 5039 收藏  
展开

##### ??【题目描述】

根据公式： $\arctan x(x)=x-x^3/3+x^5/5-x^7/7+\dots$  和  $\pi=6\arctan x(1/\sqrt{3})$ . 定义函数  $\arctan x(x)$ ，求当最后一项小于  $10^{-(?6)}$  时  $\pi$  的值。

##### 【输入】

(无)

##### 【输出】

$\pi$  的值。保留到小数点后 10 位。

##### 【输入样例】

(无)

**【输出样例】**

(无)

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
double calculate(double x);

int main()
{
    double a,pi;
    a=1/sqrt(3);//计算公式的参数
    pi=6*(calculate(a));//调用函数计算
    printf("%.10lf\n",pi);//保留十位小数输出
    return 0;
}
double calculate(double x)
{
    double sum=0,temp=x;
    int i=1;

    while(fabs(temp/i)>=1e-6)//最后一项绝对值大于 10-6时进行循环
    {
        sum+=temp/i;//累加各项
        temp=-1*x*x*temp;//求下一项
        i+=2;//系数+2
    }
    return sum;
}

```

点赞 5

**1.1.1.6.1.8 哥德巴赫猜想（信息学奥赛一本通-T1157）****哥德巴赫猜想（信息学奥赛一本通-T1157）**

原创 Alex\_McAvoy 发布于 2018-03-06 17:33:10 阅读数 4358 收藏  
展开

**【题目描述】**

哥德巴赫猜想的命题之一是:大于 6 的偶数等于两个素数之和。编程将 6~100 所有偶数表示成两个素数之和。

**【输入】**

(无)

**【输出】**

分行输出:

例如:6=3+3

8=3+5

...

(每个数只拆开一次, 请保证第一个加数最小)

**【输入样例】**

(无)

**【输出样例】**

(无)

**【源程序】**

```
#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int x;
    int i;

    for(x=6; x<=100; x+=2) //枚举 6-100 的偶数
        for(i=2; i<=x/2; i++) //将 x 分解为 i 与 x-i 两个数
            if(judge(i)&&judge(x-i))//若被分解的两个数皆为素数
            {
                cout<<x<<"="<<i<<"+"<<x-i<<endl;//输出
                break;//终止循环
            }
    return 0;
}

bool judge(int x)//判断素数
{
    int i=2;
```

```

while( i<=floor(sqrt(x)) && (x%i!=0) )
    i++;
if(i>floor(sqrt(x)))
    return true;
return false;
}

```

点赞 2

### 1.1.1.6.1.9 简单算术表达式求值（信息学奥赛一本通-T1397）

#### 简单算术表达式求值（信息学奥赛一本通-T1397）

原创 Alex\_McAvoy 发布于 2018-03-07 21:53:43 阅读数 3368 收藏  
展开

##### 【题目描述】

两位正整数的简单算术运算（只考虑整数运算），算术运算为：

+，加法运算；  
 -，减法运算；  
 \*，乘法运算；  
 /，整除运算；  
 %，取余运算。

算术表达式的格式为（运算符前后可能有空格）：运算数 运算符 运算数

请输出相应的结果。

##### 【输入】

一行算术表达式。

##### 【输出】

整型算数运算的结果（结果值不一定为 2 位数，可能多于 2 位或少于 2 位）。

##### 【输入样例】

32+64

##### 【输出样例】

96

##### 【源程序】

```

#include<iostream>
using namespace std;

```



```

int calculate(int x,char z,int y);

int main()
{
    int x,y;
    char z;

    cin>>x>>z>>y;//x、y 为运算数，z 为运算符
    cout<<calculate(x,z,y)<<endl;//根据 z 进行计算，输出结果
    return 0;
}

int calculate(int x,char z,int y)
{
    if(z=='+') return x+y;
    if(z=='-') return x-y;
    if(z=='*') return x*y;
    if(z=='/') return x/y;
    if(z=='%') return x%y;
}

```

点赞 2

### 1.1.1.6.1.10 短信计费（信息学奥赛一本通-T1398）

#### 短信计费（信息学奥赛一本通-T1398）

原创 Alex\_McAvoy 发布于 2018-03-07 22:03:03 阅读数 4533 收藏  
展开

##### 【题目描述】

用手机发短信，一条短信资费为 0.1 元，但限定一条短信的内容在 70 个字以内(包括 70 个字)。如果你一次所发送的短信超过了 70 个字，则会按照每 70 个字一条短信的限制把它分割成多条短信发送。假设已经知道你当月所发送的短信的字数，试统计一下你当月短信的总资费。

##### 【输入】

第一行是整数  $n$ ，表示当月发送短信的总次数，接着  $n$  行每行一个整数，表示每次短信的字数。

##### 【输出】

输出一行，当月短信总资费，单位为元，精确到小数点后 1 位。

##### 【输入样例】

10

39  
49  
42  
61  
44  
147  
42  
72  
35  
46

### 【输出样例】

1.3

### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
int calculate(int x);

int main()
{
    int n,a;
    int i;
    double sum=0;

    cin>>n;//输入当月发送短信总次数 n
    for(i=1; i<=n; i++)
    {
        cin>>a;//输入每次发送短信的字数
        sum+=calculate(a);//累加每次短信的资费
    }
    printf("%.1lf\n",sum/10.0);//将以角为单位的数转为以元为单位的数，输出结果
    return 0;
}

int calculate(int x)
{
    if(x%70==0)
        return x/70;
    else
        return x/70+1;
}
```

### 1.1.1.6.1.11 甲流病人初筛（信息学奥赛一本通-T1399）

#### 甲流病人初筛（信息学奥赛一本通-T1399）

原创 Alex\_McAvoy 发布于 2018-03-08 19:40:25 阅读数 4515 收藏  
展开

##### 【题目描述】

目前正是甲流盛行时期，为了更好地进行分流治疗，医院在挂号时要求对病人的体温和咳嗽情况进行检查，对于体温超过 37.5 度（含等于 37.5 度）并且咳嗽的病人初步判定为甲流病人（初筛）。现需要统计某天前来挂号就诊的病人中有多少人被初筛为甲流病人。

##### 【输入】

第一行是某天前来挂号就诊的病人数  $n$ 。( $n < 200$ )

其后有  $n$  行，每行是病人的信息，包括三个信息：姓名（字符串，不含空格，最多 8 个字符）、体温（float）、是否咳嗽（整数，1 表示咳嗽，0 表示不咳嗽）。每行三个信息之间以一个空格分开。

##### 【输出】

按输入顺序依次输出所有被筛选为甲流的病人的姓名，每个名字占一行。之后在输出一行，表示被筛选为甲流的病人数量。

##### 【输入样例】

```
5
Zhang 38.3 0
Li 37.5 1
Wang 37.1 1
Zhao 39.0 1
Liu 38.2 1
```

##### 【输出样例】

```
Li
Zhao
Liu
3
```

##### 【源程序】

```
#include<iostream>
#include<string>
using namespace std;
bool judge(float x,int y);
```

```

int main()
{
    int n;
    float temperature[201];
    int cough[201];
    string name[201];
    int sum=0;
    int i;

    cin>>n;//输入病人数
    for(i=1;i<=n;i++)//依次输入姓名、体温、是否咳嗽
        cin>>name[i]>>temperature[i]>>cough[i];
    for(i=1;i<=n;i++)
        if(judge(temperature[i],cough[i]))//进行判断
        {
            cout<<name[i]<<endl;//输出姓名
            sum++;//总数+1
        }
    cout<<sum<<endl;//输出总数
    return 0;
}

bool judge(float x,int y)
{
    if((x>=37.5)&&(y==1))
        return true;
    else
        return false;
}

```

### 1.1.1.6.1.12 统计单词数（信息学奥赛一本通-T1400）

#### 统计单词数（信息学奥赛一本通-T1400）

原创 Alex\_McAvoy 发布于 2018-03-08 19:45:01 阅读数 5269 收藏  
展开

##### 【题目描述】

一般的文本编辑器都有查找单词的功能，该功能可以快速定位特定单词在文章中的位置，有的还能统计出特定单词在文章中出现的次数。

现在，请你编程实现这一功能，具体要求是：给定一个单词，请你输出它在给定的文章中出现的次数和第一次出现的位置。注意：匹配单词时，不区分大小写，但要求完

全匹配，即给定单词必须与文章中的某一独立单词在不区分大小写的情况下完全相同（参见样例 1），如果给定单词仅是文章中某一单词的一部分则不算匹配（参见样例 2）。

### 【输入】

第 1 行为一个字符串，其中只含字母，表示给定单词；

第 2 行为一个字符串，其中只可能包含字母和空格，表示给定的文章。

### 【输出】

只有一行，如果在文章中找到给定单词则输出两个整数，两个整数之间用一个空格隔开，分别是单词在文章中出现的次数和第一次出现的位置（即在文章中第一次出现时，单词首字母在文章中的位置，位置从 0 开始）；如果单词在文章中没有出现，则直接输出一个整数-1。

### 【输入样例】

#样例一

To

to be or not to be is a question

#样例二

to

Did the Ottoman Empire lose its power at that time

### 【输出样例】

#样例一

2 0

#样例二

-1

### 【源程序】

```
#include<iostream>
#include<cstring>
#include<cstdio>
#include<string>
using namespace std;

int main()
{
    string word,sentence;
    int len1,len2;
    int i,j;
    int ans=0,direction;

    getline(cin,word);//输入单词
    getline(cin,sentence);//输入句子
```

```

len1=word.size();//记录单词的长度
len2=sentence.size();//记录句子的长度

for(i=0; i<len2; i++)
{
    for(j=0; j<len1; j++)
    {
        if(toupper(sentence[i+j])!=toupper(word[j]))
            break;//若字符不匹配退出循环，再从第 i+1 个开始判定
        /*toupper()为将字符从小写转为大写的库函数*/
        if(i>0&&sentence[i-1]!=' ')
            break;//若第 i 个字符不是文章首且前一个字符不是空格，则不是独立单词，
退出循环
    }
    if(j==len1&&(sentence[i+j]==' '||j+i==len2))//若循环判定通过，且该单词词尾为
文章尾或后有空格
    {
        ans++;//累加满足条件单词数
        if(ans==1)//若第一次出现
            direction=i;//记录位置
    }
}
if(ans)
    cout<<ans<<" "<<direction<<endl;
else
    cout<<"-1"<<endl;

return 0;

```

### 1.1.1.6.1.13 机器翻译（信息学奥赛一本通-T1401）

#### 机器翻译（信息学奥赛一本通-T1401）

原创 Alex\_McAvoy 发布于 2018-03-08 20:33:54 阅读数 3112 收藏  
展开

##### 【题目描述】

小晨的电脑上安装了一个机器翻译软件，他经常用这个软件来翻译英语文章。

这个翻译软件的原理很简单，它只是从头到尾，依次将每个英文单词用对应的中文含义来替换。对于每个英文单词，软件会先在内存中查找这个单词的中文含义，如果内存中有，软件就会用它进行翻译；如果内存中没有，软件就会在外存中的词典内查找，

查出单词的中文含义然后翻译，并将这个单词和译义放入内存，以备后续的查找和翻译。

假设内存中有  $M$  个单元，每单元能存放一个单词和译义。每当软件将一个新单词存入内存前，如果当前内存中已存入的单词数不超过  $M-1$ ，软件会将新单词存入一个未使用的内存单元；若内存中已存入  $M$  个单词，软件会清空最早进入内存的那个单词，腾出单元来，存放新单词。

假设一篇英语文章的长度为  $N$  个单词。给定这篇待译文章，翻译软件需要去外存查找多少次词典？假设在翻译开始前，内存中没有任何单词。

### 【输入】

2 行。每行中两个数之间用一个空格隔开。

第一行为两个正整数  $M$  和  $N$ ，代表内存容量和文章的长度。

第二行为  $N$  个非负整数，按照文章的顺序，每个数（大小不超过 1000）代表一个英文单词。文章中两个单词是同一个单词，当且仅当它们对应的非负整数相同。

### 【输出】

共 1 行，包含一个整数，为软件需要查词典的次数。

### 【输入样例】

```
3 7
1 2 1 5 4 4 1
```

### 【输出样例】

```
5
```

### 【源程序】

```
#include<iostream>
using namespace std;

void replace();
int a[1001]= {0};
int ans=0;
int m,n;
int x;

int main()
{
    int i;

    cin>>m>>n;
    for(i=1; i<=n; i++)
    {
```

```

cin>>x;//输入单词
if(a[x]==0&&ans<m)//需要查找且内存未满
{
    ans++;//查找次数+1
    a[x]=ans;//存储 ans，桶排的思想
}
else//若内存已满
    replace();//调用函数
}
cout<<ans<<endl;
return 0;
}
void replace()
{
    int sum;
    int i;

    if(a[x]==0&&ans>=m)//需要查找内存已满
    {
        ans++;
        sum=0;
        for(i=0; i<=1000; i++)
        {
            if(a[i]>=1)//每个单词存入顺序向前移动一位
            {
                a[i]=a[i]-1;
                sum++;
            }
            if(sum==m)//移动过后结束循环
                break;
        }
        if(m>0)//若内存非零则存入新单词
            a[x]=m;
    }
}
}

```

#### 1.1.1.6.1.14 Vigenère 密码（信息学奥赛一本通-T1402）

##### Vigenère 密码（信息学奥赛一本通-T1402）

原创 Alex\_McAvoy 发布于 2018-03-08 21:00:44 阅读数 3130 收藏  
展开



**【题目描述】**

16 世纪法国外交家 **Blaise de Vigenère** 设计了一种多表密码加密算法——**Vigenère** 密码。**Vigenère** 密码的加密解密算法简单易用，且破译难度比较高，曾在美国南北战争中为南军所广泛使用。

在密码学中，我们称需要加密的信息为明文，用 **M** 表示；称加密后的信息为密文，用 **C** 表示；而密钥是一种参数，是将明文转换为密文或将密文转换为明文的算法中输入的数据，记为 **k**。在 **Vigenère** 密码中，密钥 **k** 是一个字母串， $k=k_1k_2\dots k_n$ 。当明文  $M=m_1m_2\dots m_n$  时，得到的密文  $C=c_1c_2\dots c_n$ ，其中  $c_i=m_i?k_i$ ，运算?的规则如下表所示：

**Vigenère** 加密在操作时需要注意：

- 1.?运算忽略参与运算的字母的大小写，并保持字母在明文 **M** 中的大小写形式；
- 2.当明文 **M** 的长度大于密钥 **k** 的长度时，将密钥 **k** 重复使用。

例如，明文  $M=Helloworld$ ，密钥  $k=abc$  时，密文  $C=Hfnlpyosnd$ 。

??

**【输入】**

第一行为一个字符串，表示密钥 **k**，长度不超过 100，其中仅包含大小写字母。

第二行为一个字符串，表示经加密后的密文，长度不超过 1000，其中仅包含大小写字母。

对于 100% 的数据，输入的密钥的长度不超过 100，输入的密文的长度不超过 1000，且都仅包含英文字母。

**【输出】**

输出共 1 行，一个字符串，表示输入密钥和密文所对应的明文。

**【输入样例】**

```
CompleteVictory
Yvqgpxaimmklongnzfwpxmniytm
```

**【输出样例】**

```
Wherethereisawillthereisaway
```

**【源程序】**

```
#include<iostream>
```

```

#include<cstdio>
#include<cstring>
using namespace std;
void turn(char &x);
int main()
{
    char key[1000],secret[1000];
    int len1,len2;
    int i,j;

    gets(key);//输入密钥
    gets(secret);//输入密文
    len1=strlen(key);//求密钥长度
    len2=strlen(secret);//求密文长度

    for(i=0; i<len1; i++) //将密钥转换为大写字母
        turn(key[i]);

    for(i=0,j=0; i<len2; i++) //从头开始解密
    {
        if(secret[i]<='Z')//大写的情况
        {
            secret[i]=secret[i]-(key[j]-'A');//按规则解密
            j++;
            if(secret[i]<'A')//若明文小于 A
                secret[i]='Z'-'A'-secret[i]+1;//从字母表倒序转换
        }
        else//小写的情况
        {
            secret[i]=secret[i]-(key[j]-'A');//按规则解密
            j++;
            if(secret[i]<'a')//若明文小于 a
                secret[i]='z'-'a'-secret[i]+1;//从字母表倒序转换
        }
        if(j>len1-1)//若明文长度大于密钥长度，重复使用密钥
            j=0;
    }
    cout<<secret<<endl;
    return 0;
}
void turn(char &x)
{
    if((x>='a')&&(x<='z'))

```

```

    x-=32;
}

```

### 1.1.1.6.1.15 素数对（信息学奥赛一本通-T1403）

#### 素数对（信息学奥赛一本通-T1403）

原创 Alex\_McAvoy 发布于 2018-03-06 22:21:51 阅读数 2819 收藏  
展开

##### 【题目描述】

两个相差为 2 的素数称为素数对，如 5 和 7，17 和 19 等，本题目要求找出所有两个数均不大于  $n$  的素数对。

##### 【输入】

一个正整数  $n(1 \leq n \leq 10000)$ 。

##### 【输出】

所有小于等于  $n$  的素数对。每对素数对输出一行，中间用单个空格隔开。若没有找到任何素数对，输出 `empty`。

##### 【输入样例】

100

##### 【输出样例】

```

3 5
5 7
11 13
17 19
29 31
41 43
59 61
71 73

```

##### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int n;
    int i;

```

```

bool flag=true;

cin>>n;
for(i=2; i<=n-2; i++) //从 2 遍历到 n-2
    if( judge(i+2) && judge(i) )//若存在素数对则输出
    {
        cout<<i<<" "<<i+2<<endl;;
        flag=false;
    }
if(flag)//不存在素数对则输出 empty
    cout<<"empty"<<endl;
return 0;
}

bool judge(int x)//判断素数
{
    int i=2;
    for(i=2; i<=sqrt(x); i++)
        if(x%i==0)
            return false;
    return true;
}

```

### 1.1.1.6.1.16 我家的门牌号（信息学奥赛一本通-T1404）

#### 我家的门牌号（信息学奥赛一本通-T1404）

原创 Alex\_McAvoy 发布于 2018-03-08 22:15:19 阅读数 4439 收藏  
展开

##### 【题目描述】

我家住在一条短胡同里，这条胡同的门牌号从 1 开始顺序编号。

若其余各家的门牌号之和减去我家门牌号的两倍，恰好等于  $n$ ，求我家的门牌号及总共有多少家。数据保证有唯一解。

##### 【输入】

一个正整数  $n$ 。  $n < 100000$ 。

##### 【输出】

一行，包含两个正整数，分别是我家的门牌号及总共有多少家，中间用单个空格隔开。

##### 【输入样例】

100

## 【输出样例】

12 16

## 【源程序】

```
#include<iostream>
#include<cmath>
using namespace std;

int main()
{
    int n,home,x;
    int i;

    cin>>n;//输入 n

    /*
    设最后门牌号为 k，则总和为： $k*(k+1)/2$ 
    设家的门牌号为 x，则由题意： $k*(k+1)/2=3*x+n$ 
    即： $k*(k+1)=6*x+2*n$ 
    又： $k*(k+1)+k+1>6*x+2*n$ 
    得： $(k+1)*(k+1)>6*x+2*n$ 
    开方： $k+1>\sqrt{6*x+2*n}$ 
    两边减一： $k>\sqrt{6*x+2*n}-1$ 
    x 取最小值 1： $home=\sqrt{6+2*n}-1$ 
    */

    home=sqrt(6+2*n)-1;
    for(i=home;; i++) //从家的门牌号开始逐个尝试
    {
        if((i*i+i-2*n)%6==0)
        {
            x=(i*i+i-2*n)/6;
            if(x<=0) continue;//x 不可能<1
            if(x>0)//找到后
            {
                cout<<x<<" "<<i;//输出
                break;//终止循环
            }
        }
    }
    return 0;
}
```

### 1.1.1.6.1.17 质数的和与积（信息学奥赛一本通-T1405）

#### 质数的和与积（信息学奥赛一本通-T1405）

原创 Alex\_McAvoy 发布于 2018-03-08 21:40:47 阅读数 2188 收藏  
展开

##### 【题目描述】

两个质数的和是  $S$ ，它们的积最大是多少？

##### 【输入】

一个不大于 10000 的正整数  $S$ ，为两个质数的和。

##### 【输出】

一个整数，为两个质数的最大乘积。数据保证有解。

##### 【输入样例】

50

##### 【输出样例】

589

##### 【源程序】

```
#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int s;
    int i;

    cin>>s;//输入正整数 s
    for(i=s/2; i>=2; i--) //和为定值的两个数，越接近，数越大，因此从中间拆分
        if(judge(i)&&judge(s-i))//判断拆分的两个数是否为素数
        {
            cout<<i*(s-i)<<endl;//若是，输出两数积
            break;//终止循环
        }

    return 0;
}
bool judge(int x)//判断素数
```

```

{
    int i;
    if(x==2) return false;
    for(i=2; i<=sqrt(x); i++)
        if(x%i==0)
            return false;
    return true;
}

```

### 1.1.1.6.1.18 单词替换（信息学奥赛一本通-T1406）

#### 单词替换（信息学奥赛一本通-T1406）

原创 Alex\_McAvoy 发布于 2018-03-08 21:33:57 阅读数 3340 收藏  
展开

##### 【题目描述】

输入一个字符串，以回车结束（字符串长度 $\leq 200$ ）。该字符串由若干个单词组成，单词之间用一个空格隔开，所有单词区分大小写。现需要将其中的某个单词替换成另一个单词，并输出替换之后的字符串。

##### 【输入】

第 1 行是包含多个单词的字符串  $s$ ;

第 2 行是待替换的单词  $a$ (长度  $\leq 100$ );

第 3 行是  $a$  将被替换的单词  $b$ (长度  $\leq 100$ )。

$s, a, b$  最前面和最后面都没有空格。

##### 【输出】

输出只有 1 行，将  $s$  中所有单词  $a$  替换成  $b$  之后的字符串。

##### 【输入样例】

```

You want someone to help you
You
I

```

##### 【输出样例】

```

I want someone to help you

```

##### 【源程序】

```

#include<iostream>
#include<string>
#include<cstring>

```

```

#include<cstdio>
using namespace std;

void work(int i);
string str[101],a,b;

int main()
{
    int n=0;
    int i;
    char space;

    /*由于至少需要输入一个单词，因此使用 do-while 结构输入*/
    do
    {
        n++;
        cin>>str[n];//输入一个单词（字符组）
        scanf("%c",&space);//读入一个空格
    }
    while(space==' ');

    cin>>a>>b;//输入代替换的单词与要替换的单词

    for(i=1; i<=n; i++) //从第一个单词开始枚举
        work(i);//调用函数进行判断，是否需要替换

    cout<<endl;
    return 0;
}

void work(int i)
{
    if(a==str[i])
        cout<<b<<" ";
    else
        cout<<str[i]<<" ";
}

```

### 1.1.1.6.1.19 笨小猴（信息学奥赛一本通-T1407）

#### 笨小猴（信息学奥赛一本通-T1407）

原创 Alex\_McAvoy 发布于 2018-03-10 21:27:22 阅读数 3247 收藏  
展开



**【题目描述】**

笨小猴的词汇量很小，所以每次做英语选择题的时候都很头疼。但是他找到了一种方法，经试验证明，用这种方法去选择选项的时候选对的几率非常大！

这种方法的具体描述如下：假设  $\text{maxn}$  是单词中出现次数最多的字母的出现次数， $\text{minn}$  是单词中出现次数最少的字母的出现次数，如果  $\text{maxn}-\text{minn}$  是一个质数，那么笨小猴就认为这是个 **Lucky Word**，这样的单词很可能就是正确的答案。

**【输入】**

只有一行，是一个单词，其中只可能出现小写字母，并且长度小于 100。

**【输出】**

共两行，第一行是一个字符串，假设输入的的单词是 **Lucky Word**，那么输出“**Lucky Word**”，否则输出“**No Answer**”；

第二行是一个整数，如果输入单词是 **Lucky Word**，输出  $\text{maxn}-\text{minn}$  的值，否则输出 0。

**【输入样例】**

error

**【输出样例】**

Lucky Word

2

**【源程序】**

```
#include<iostream>
#include<cmath>
#include<string>
using namespace std;
bool prime(int x);

int main()
{
    string str;
    int a[26]= {0};
    int max=0,min=9999;
    int i;

    cin>>str;//输入字符串
    for(i=0; i<str.length(); i++) //统计各字母出现次数
        a[str[i]-'a']+=1;
    for(i=0; i<26; i++) //统计最多、最少字母的出现次数
    {
        if(max<a[i])
            max=a[i];
```

```

        if(min>a[i]&& a[i]!=0)
            min=a[i];
    }
    if(prime(max-min))
        cout<<"Lucky Word"<<endl<<(max-min)<<endl;
    else
        cout<<"No Answer"<<endl<<0<<endl;
    return 0;
}

bool prime(int x)//判断素数
{
    int i;
    if(x<2)
        return false;
    else
        for(i=2; i<=sqrt(x); i++)
            if(x%i==0)
                return false;
    return true;
}

```

### 1.1.1.6.1.20 素数回文数的个数（信息学奥赛一本通-T1408）

#### 素数回文数的个数（信息学奥赛一本通-T1408）

原创 Alex\_McAvoy 发布于 2018-03-06 21:32:49 阅读数 2998 收藏  
展开

##### 【题目描述】

求 11 到  $n$  之间（包括  $n$ ），既是素数又是回文数的整数有多少个。

##### 【输入】

一个大于 11 小于 1000 的整数  $n$ 。

##### 【输出】

11 到  $n$  之间的素数回文数个数。

##### 【输入样例】

23

##### 【输出样例】

1

## 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
bool prime(int x);
int palindrome(int n);

int main()
{
    int n;
    int ans=0;
    int i;

    cin>>n;
    for(i=11; i<=n; i++) //从 11 枚举到 n
        if( prime(i) && palindrome(i)==i )//分别判断 i 是否是素数以及回文数
            ans++;
    cout<<ans<<endl;
    return 0;
}

bool prime(int x)//判断素数
{
    int i=2;
    while( i<=floor(sqrt(x)) && (x%i!=0) )
        i++;
    if(i>floor(sqrt(x)))
        return true;
    return false;
}

int palindrome(int n)//判断回文数
{
    int sum=0;
    while(n>0)
    {
        sum=sum*10+n%10;
        n/=10;
    }
    return sum;
}

```

### 1.1.1.6.1.21 判决素数个数（信息学奥赛一本通-T1409）

#### 判决素数个数（信息学奥赛一本通-T1409）

原创 Alex\_McAvoy 发布于 2018-03-06 21:36:53 阅读数 1922 收藏  
展开

##### 【题目描述】

输入两个整数  $X$  和  $Y$ ，输出两者之间的素数个数（包括  $X$  和  $Y$ ）。

##### 【输入】

两个整数  $X$  和  $Y$  ( $1 \leq X, Y \leq 105$ )。

##### 【输出】

输出一个整数，表示  $X$ ， $Y$  之间的素数个数（包括  $X$  和  $Y$ ）。

##### 【输入样例】

1 100

##### 【输出样例】

25

##### 【源程序】

```
#include<iostream>
#include<cmath>
using namespace std;
bool prime(int x);

int main()
{
    int x,y;
    int ans=0;
    int i;

    cin>>x>>y;
    for(i=x; i<=y; i++) //从 x 枚举到 y
        if( prime(i) )//分别判断 i 是否是素数
            ans++;
    cout<<ans<<endl;
    return 0;
}

bool prime(int x)//判断素数
{
```

```

int i=2;
if(x==1||x==0) return false;
while( i<=floor(sqrt(x)) && (x%i!=0) )
    i++;
if(i>floor(sqrt(x)))
    return true;
return false;
}

```

### 1.1.1.6.1.22 最大质因子序列（信息学奥赛一本通-T1410）

#### 最大质因子序列（信息学奥赛一本通-T1410）

原创 Alex\_McAvoy 发布于 2018-03-06 22:13:48 阅读数 2534 收藏  
展开

##### 【题目描述】

任意输入两个正整数  $m, n (1 < m < n \leq 5000)$ ，依次输出  $m$  到  $n$  之间每个数的最大质因子 (包括  $m$  和  $n$ ；如果某个数本身是质数，则输出这个数自身)。

##### 【输入】

一行，包含两个正整数  $m$  和  $n$ ，其间以单个空格间隔。

##### 【输出】

一行，每个整数的最大质因子，以逗号间隔。

##### 【输入样例】

5 10

##### 【输出样例】

5,3,7,2,3,5

##### 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);

int main()
{
    int m,n;
    int i,j;

    cin>>m>>n;

```

```

for(i=m; i<=n; i++) //从 m 枚举到 n
    for(j=i; j>=2; j--) //从小到大找每个数的最大质因子
        if( (i%j==0) && judge(j) )//找到最大质因子
        {
            cout<<j;
            if(i!=n)
                cout<<" ";
            break;
        }
    cout<<endl;
return 0;
}

bool judge(int x)//判断素数
{
    int i=2;
    for(i=2; i<=sqrt(x); i++)
        if(x%i==0)
            return false;
    return true;
}

```

### 1.1.1.6.1.23 区间内的真素数（信息学奥赛一本通-T1411）

#### 区间内的真素数（信息学奥赛一本通-T1411）

原创 Alex\_McAvoy 发布于 2018-03-06 22:01:19 阅读数 3660 收藏  
展开

##### 【题目描述】

找出正整数  $M$  和  $N$  之间（ $N$  不小于  $M$ ）的所有真素数。

例如，11，13 均为真素数，因为 11 的反序还是为 11，13 的反序为 31 也为素数。

真素数的定义：如果一个正整数  $P$  为素数，且其反序也为素数，那么  $P$  就为真素数。

##### 【输入】

输入两个数  $M$  和  $N$ ，空格间隔， $1 \leq M \leq N \leq 100000$ 。

##### 【输出】

按从小到大输出  $M$  和  $N$  之间（包括  $M$  和  $N$ ）的真素数，逗号间隔。如果之间没有真素数，则输出 No。

## 【输入样例】

10 35

## 【输出样例】

11,13,17,31

## 【源程序】

```

#include<iostream>
#include<cmath>
using namespace std;
bool judge(int x);
int inverted(int n);
int a[100000];

int main()
{
    int m,n;
    int i;
    int k=0;
    bool flag=false;

    cin>>m>>n;
    for(i=m; i<=n; i++) //从 m 枚举到 n
        if( judge(i) && judge(inverted(i)) )//分别判断 i 以及 i 变换后是否是素数
        {
            k++;
            a[k]=i;
            flag=true;
        }
    if(flag)
    {
        for(i=1; i<=k; i++)
            cout<<a[i]<<" ";
        cout<<a[k]<<endl;
    }
    else
        cout<<"No"<<endl;
    return 0;
}

bool judge(int x)//判断素数
{
    int i=2;
    if(x==0||x==1) return false;
    while( i<=floor(sqrt(x)) && (x%i!=0) )

```

```

        i++;
    if(i>floor(sqrt(x)))
        return true;
    return false;
}

int inverted(int n)//求倒序数
{
    int sum=0;
    while(n>0)
    {
        sum=sum*10+n%10;
        n/=10;
    }
    return sum;
}

```

### 1.1.1.6.1.24 二进制分类（信息学奥赛一本通-T1412）

#### 二进制分类（信息学奥赛一本通-T1412）

原创 Alex\_McAvoy 发布于 2018-03-10 22:12:24 阅读数 3006 收藏  
展开

##### 【题目描述】

若将一个正整数化为二进制数，在此二进制数中，我们将数字 1 的个数多于数字 0 的个数的这类二进制数称为 A 类数，否则就称其为 B 类数。

例如：

(13)<sub>10</sub>=(1101)<sub>2</sub>，其中 1 的个数为 3，0 的个数为 1，则称此数为 A 类数；  
(10)<sub>10</sub>=(1010)<sub>2</sub>，其中 1 的个数为 2，0 的个数也为 2，称此数为 B 类数；  
(24)<sub>10</sub>=(11000)<sub>2</sub>，其中 1 的个数为 2，0 的个数为 3，则称此数为 B 类数；  
程序要求：求出 1~1000 之中（包括 1 与 1000），全部 A、B 两类数的个数。

##### 【输入】

(无)

##### 【输出】

一行，包含两个整数，分别是 A 类数和 B 类数的个数，中间用单个空格隔开。

##### 【输入样例】

(无)



**【输出样例】**

(无)

**【源程序】**

```

#include<iostream>
using namespace std;
bool judge(int x);

int main()
{
    int a=0,b=0;
    int i;

    for(i=1; i<=1000; i++) //枚举 1-1000 的数
        if(judge(i))//若 i 是 A 类数
            a++;
        else//若 i 是 B 类数
            b++;
    cout<<a<<" "<<b<<endl;
    return 0;
}

bool judge(int x)
{
    int a=0,b=0;
    while(x>0)//将一个数化为二进制数,并判断是 A 类数还是 B 类数
    {
        if(x%2)
            a++;//a 统计 1 的个数
        else
            b++;//b 统计 0 的个数
        x/=2;
    }

    return a>b;//
}

```

点赞 1

收藏

分享

### 1.1.1.6.1.25 确定进制（信息学奥赛一本通-T1413）

#### 确定进制（信息学奥赛一本通-T1413）

原创 Alex\_McAvoy 发布于 2018-03-10 22:27:36 阅读数 2852 收藏  
展开

##### 【题目描述】

$6*9=42$  对于十进制来说是错误的，但是对于 13 进制来说是正确的。即  $6(13)*9(13)=42(13)$ ，而  $42(13)=4*13+2*130=54(10)$ 。

你的任务是写一段程序，读入三个整数  $p$ 、 $q$  和  $r$ ，然后确定一个进制  $B(2 \leq B \leq 40)$  使得  $p*q=r$ 。如果  $B$  有很多选择，输出最小的一个。

例如： $p=11, q=11, r=121$ 。则有  $11(3)*11(3)=121(3)$  因为  $11(3)=1*3+1*30=4(10)$  和  $121(3)=1*3^2+2*3+1*30=16(10)$ 。对于进制 10，同样有  $11(10)*11(10)=121(10)$ 。这种情况下，应该输出 3。如果没有合适的进制，则输出 0。

##### 【输入】

一行，包含三个整数  $p$ 、 $q$ 、 $r$ 。 $p$ 、 $q$ 、 $r$  的所有位都是数字，并且  $1 \leq p, q, r \leq 1,000,000$ 。

##### 【输出】

一个整数：即使得  $p*q=r$  成立的最小的  $B$ 。如果没有合适的  $B$ ，则输出 0。

##### 【输入样例】

6 9 42

##### 【输出样例】

13

##### 【源程序】

```
#include<iostream>
using namespace std;
int judge(int x,int B);

int main()
{
    int a,b,c;
    int i;

    cin>>a>>b>>c;
    for(i=2; i<=40; i++) //枚举 2-40 进制
        if(judge(a,i)*judge(b,i)==judge(c,i))//找到使 a*b=c 成立的最小进制 B
        {
```

```

        cout<<i<<endl;
        return 0;
    }
    cout<<0<<endl;
    return 0;
}

int judge(int x,int B)
{
    int value=1,num=0;
    while(x!=0)//将一个数化为 B 进制数
    {
        if((x%10)>=B)//B 进制下不可能出现大于等于 B 的数位
            return 99999;
        num+=((x%10)*value);
        value*=B;
        x/=10;
    }
    return num;
}

```

### 1.1.1.6.2 第二节 递归算法

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第一部分：C++语言](#) > [第六章 函数](#) > 第二节 递归算法

#### 1.1.1.6.2.1 求 $1+2+3+\dots$ （信息学奥赛一本通-T1158）

#### 求 $1+2+3+\dots$ （信息学奥赛一本通-T1158）

原创 Alex\_McAvoy 发布于 2018-03-11 15:35:42 阅读数 2784 收藏  
展开

##### 【题目描述】

用递归的方法求  $1+2+3+\dots+N$  的值。

##### 【输入】

输入  $n$ 。

##### 【输出】

输出和。

**【输入样例】**

5

**【输出样例】**

15

**【源程序】**

```
#include<iostream>
using namespace std;
int calculate(int n);

int main()
{
    int n;
    cin>>n;//输入 n 的值
    cout<<calculate(n)<<endl;//调用函数计算从 1—n 的累加值
    return 0;
}

int calculate(int n)
{
    if(n==0) return 0;//判断是否到达递归边界
    else return n+calculate(n-1);//未到达继续递归
```

### 1.1.1.6.2.2 斐波那契数列（信息学奥赛一本通-T1159）

#### 斐波那契数列（信息学奥赛一本通-T1159）

原创 Alex\_McAvoy 发布于 2018-03-11 15:39:28 阅读数 3560 收藏  
展开

**【题目描述】**

用递归函数输出斐波那契数列第  $n$  项。0,1, 1, 2, 3, 5, 8, 13.....

**【输入】**

一个正整数  $n$ ,表示第  $n$  项。

**【输出】**

第  $n$  项是多少。

**【输入样例】**

3

**【输出样例】**

1

**【源程序】**

```
#include<iostream>
using namespace std;
int calculate(int n);

int main()
{
    int n;
    cin>>n;//输入 n 的值
    cout<<calculate(n)<<endl;//调用函数计算并输出斐波那契数列第 n 项
    return 0;
}

int calculate(int n)
{
    if(n==1) return 0;//判断是否到达递归边界 n=1
    else if(n==2) return 1;//判断是否到达递归边界 n=2
    else return calculate(n-1)+calculate(n-2);//未到达继续递归
}
```

点赞 1

**1.1.1.6.2.3 倒序数（信息学奥赛一本通-T1160）****倒序数（信息学奥赛一本通-T1160）**

原创 Alex\_McAvoy 发布于 2018-03-11 15:44:47 阅读数 3989 收藏  
展开

**【题目描述】**

输入一个非负整数，输出这个数的倒序数。例如输入 123，输出 321。

**【输入】**

输入一个非负整数(保证个位不为零)。

**【输出】**

输出倒序的数。

**【输入样例】**

123

**【输出样例】**

321

**【源程序】**

```
#include<iostream>
using namespace std;
void calculate(int n);

int main()
{
    int n;
    cin>>n;//输入 n 的值
    calculate(n);//调用函数计算并输出 n 的倒序数
    cout<<endl;
    return 0;
}

void calculate(int n)
{
    cout<<n%10;//输出 n 的最后一位
    if(n>=10) calculate(n/10);//判断是否到达递归边界 x<10
}

点赞
```

**1.1.1.6.2.4 转进制（信息学奥赛一本通-T1161）****转进制（信息学奥赛一本通-T1161）**

原创 Alex\_McAvoy 发布于 2018-03-11 15:52:47 阅读数 3656 收藏  
展开

**【题目描述】**

用递归算法将一个十进制数  $X$  转换成任意进制数  $M$  ( $M \leq 16$ )。

**【输入】**

一行两个数，第一个十进制数  $X$ ，第二个为进制  $M$ 。

**【输出】**

输出结果。

**【输入样例】**

31 16 {将十进制 31 转化为十六进制数}

**【输出样例】**

1F

**【源程序】**

```

#include<iostream>
using namespace std;
void calculate(int n,int k);

int main()
{
    int x,m;
    cin>>x>>m;//输入十进制数 x 与要转换成的进制 m
    calculate(x,m);//调用函数进行转换
    cout<<endl;
    return 0;
}

void calculate(int n,int k)
{
    char num[16]= {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
    int temp;
    temp=n%k;
    n/=k;
    if(n!=0)//判断是否到达边界 n=0
        calculate(n,k);
    cout<<num[temp];
}

```

点赞

**1.1.1.6.2.5 字符串逆序（信息学奥赛一本通-T1162）****字符串逆序（信息学奥赛一本通-T1162）**

原创 Alex\_McAvoy 发布于 2018-03-11 15:57:59 阅读数 3431 收藏  
展开

**【题目描述】**

输入一串以‘!’结束的字符，按逆序输出。

**【输入】**

如题述。

**【输出】**

如题述。

**【输入样例】**

abc!

**【输出样例】**

cba

**【源程序】**

```

#include<iostream>
#include<cstdio>
using namespace std;
void calculate(int n);
char ch[10000];
int main()
{
    gets(ch);//获取一行字符串
    calculate(0);//调用函数
    cout<<endl;
    return 0;
}

void calculate(int n)
{
    if(ch[n]!='!')//判断是否到达边界 ch[x]!=
        calculate(n+1);
    else return;
    cout<<ch[n];
}

```

点赞

### 1.1.1.6.2.6 阿克曼(Ackmann)函数（信息学奥赛一本通-T1163）

#### 阿克曼(Ackmann)函数（信息学奥赛一本通-T1163）

原创 Alex\_McAvoy 发布于 2018-03-11 22:56:23 阅读数 2592 收藏  
展开

**【题目描述】**

阿克曼(Ackmann)函数  $A(m, n)$  中,  $m, n$  定义域是非负整数( $m \leq 3, n \leq 10$ ), 函数值定义为:

$$akm(m, n) = n + 1; \quad (m = 0 \text{ 时})$$



$\text{akm}(m,n) = \text{akm}(m-1,1); \quad (m>0,n=0 \text{ 时})$

$\text{akm}(m,n) = \text{akm}(m-1,\text{akm}(m,n-1)); \quad (m,n>0 \text{ 时})$

**【输入】**

输入  $m$  和  $n$ 。

**【输出】**

函数值

**【输入样例】**

2 3

**【输出样例】**

9

**【源程序】**

```
#include<iostream>
using namespace std;
int ack(int m,int n);

int main()
{
    int m,n;
    cin>>m>>n;
    cout<<ack(m,n)<<endl;
    return 0;
}

int ack(int m,int n)
{
    if(m==0)//判断是否到达递归边界 m=0
        return (n+1);
    else if(n==0)//判断是否到达递归边界 n=0
        return (ack(m-1,1));
    else//否则继续递归
        return (ack(m-1,ack(m,n-1)));
}
```

### 1.1.1.6.2.7 digit 函数（信息学奥赛一本通-T1164）

**digit 函数（信息学奥赛一本通-T1164）**

原创 Alex\_McAvoy 发布于 2018-03-11 23:02:51 阅读数 2868 收藏  
展开

**【题目描述】**

在程序中定义一函数 `digit(n,k)`，它能分离出整数 `n` 从右边数第 `k` 个数字。

**【输入】**

正整数 `n` 和 `k`。

**【输出】**

一个数字。

**【输入样例】**

31859 3

**【输出样例】**

8

**【源程序】**

```
#include<iostream>
using namespace std;
int calculate(int n,int k);

int main()
{
    int n,k;
    cin>>n>>k;
    cout<<calculate(n,k)<<endl;
    return 0;
}

int calculate(int n,int k)
{
    k--;
    if(k==0)//判断是否到达递归边界 k=0
        return n%10;
    else//否则继续递归
        return calculate(n/10,k);
}
```

点赞 2

## 1.1.1.6.2.8 Hermite 多项式（信息学奥赛一本通-T1165）

### Hermite 多项式（信息学奥赛一本通-T1165）

原创 Alex\_McAvoy 发布于 2018-03-11 23:03:34 阅读数 3115 收藏  
展开

#### 【题目描述】

用递归的方法求 Hermite 多项式的值

对给定的  $x$  和正整数  $n$ ，求多项式的值，并保留两位小数。

#### 【输入】

给定的  $n$  和正整数  $x$ 。

#### 【输出】

多项式的值。

#### 【输入样例】

1 2

#### 【输出样例】

4.00

#### 【源程序】

```
#include<iostream>
#include<cstdio>
using namespace std;
double calculate(int n,int x);

int main()
{
    int n,x;
    double result;
    cin>>n>>x;
    result=calculate(n,x);
    printf("%.2lf\n",result);
    return 0;
}

double calculate(int n,int x)
{
    if(n==0)//判断是否到达递归边界 n=0
```

```

    return 1;
else if(n==1)//判断是否到达递归边界 n=1
    return 2*x;
else//否则继续递归
    return ( 2*x*calculate(n-1,x)-2*(n-1)*calculate(n-2,x) );
}

```

### 1.1.1.6.2.9 求 $f(x,n)$ （信息学奥赛一本通-T1166）

#### 求 $f(x,n)$ （信息学奥赛一本通-T1166）

原创 Alex\_McAvoy 发布于 2018-03-11 23:04:21 阅读数 3161 收藏  
展开

##### 【题目描述】

已知

计算  $f$  的值。

##### 【输入】

输入  $x$  和  $n$ 。

##### 【输出】

函数值，保留两位小数。

##### 【输入样例】

4.2 10

##### 【输出样例】

3.68

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;
double calculate(double x,double n);

int main()
{
    double n,x;
    double result;
    cin>>x>>n;

```

```

    result=calculate(x,n);
    printf("%.2lf\n",result);
    return 0;
}

double calculate(double x,double n)
{
    if(n==1)//判断是否到达递归边界 n=1
        return sqrt(1+x);
    else//否则继续递归
        return sqrt(n+calculate(x,n-1));
}

```

点赞

### 1.1.1.6.2.10 再求 $f(x,n)$ (信息学奥赛一本通-T1167)

#### 再求 $f(x,n)$ (信息学奥赛一本通-T1167)

原创 Alex\_McAvoy 发布于 2018-03-11 23:04:22 阅读数 2432 收藏  
展开

##### 【题目描述】

已知

用递归函数求解。

##### 【输入】

第一数是  $x$  的值，第二个数是  $n$  的值。

##### 【输出】

函数值，保留两位小数。

##### 【输入样例】

1 2

##### 【输出样例】

0.40

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cmath>
using namespace std;

```

```
double calculate(double x,double n);
```

```
int main()
{
    double n,x;
    double result;
    cin>>x>>n;
    result=calculate(x,n);
    printf("%.2lf\n",result);
    return 0;
}
```

```
double calculate(double x,double n)
{
    if(n==1)//判断是否到达递归边界 n=1
        return x/(1+x);
    else//否则继续递归
        return x/(n+calculate(x,n-1));
}
```

### 1.1.1.6.3 新建项目

## 1.1.2 第二部分 基础算法

### 1.1.2.1 第一章 高精度计算

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第二部分 基础算法](#) > 第一章 高精度计算

#### 1.1.2.1.1 高精度乘法（信息学奥赛一本通-T1307）

#### 高精度乘法（信息学奥赛一本通-T1307）

原创 Alex\_McAvoy 发布于 2018-04-15 13:28:06 阅读数 2933 收藏  
展开

**【题目描述】**

输入两个高精度正整数  $M$  和  $N$  ( $M$  和  $N$  均小于 100 位)。求这两个高精度数的积。

**【输入】**

输入两个高精度正整数  $M$  和  $N$ 。

**【输出】**

求这两个高精度数的积。

**【输入样例】**

36

3

**【输出样例】**

108

**【源程序】**

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;
int main()
{
    char str1[256],str2[256];
    int a[256],b[256],c[256];
    int lena,lenb,lenc;
    int x;
    int i,j;

    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    memset(c,0,sizeof(c));

    cin>>str1;//输入乘数 str1
    cin>>str2;//输入乘数 str2

    lena=strlen(str1);
    lenb=strlen(str2);
    for(i=0;i<=lena-1;i++)//乘数 str1 存入数组 a
        a[lena-i]=str1[i]-'0';
    for(i=0;i<=lenb-1;i++)//乘数 str2 存入数组 b
        b[lenb-i]=str2[i]-'0';

    for(i=1;i<=lenb;i++)
    {
        x=0;//用于存放进位
```

```

for(j=1;j<=lena;j++)//对乘数每一位进行处理
{
    c[i+j-1]=a[j]*b[i]+x+c[i+j-1];//当前乘积+上次乘积进位+原数
    x=c[i+j-1]/10;
    c[i+j-1]%=10;
}
c[i+lena]=x;//进位
}
lenc=lena+lenb;
while((c[lenc]==0)&&(lenc>1))//删除前导 0
    lenc--;
for(i=lenc;i>=1;i--)//倒序输出
    cout<<c[i];
cout<<endl;
return 0;
}

```

点赞 3

### 1.1.2.1.2 高精除（信息学奥赛一本通-T1308）

#### 高精除（信息学奥赛一本通-T1308）

原创 Alex\_McAvoy 发布于 2018-04-15 13:30:46 阅读数 2748 收藏  
展开

##### 【题目描述】

高精除以高精，求它们的商和余数。

##### 【输入】

输入两个低于 300 位的正整数。

##### 【输出】

输出商和余数。

##### 【输入样例】

```

123131231845757768789798764232456786432456787654324567142534675678
6867867867
1231312318767141738178325678412414124141425346756786867867867

```



## 【输出样例】

999999999748590

179780909068307566598992807564736854549985603543237528310337

## 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int a[100],b[100],c[100];
int compare(int a[],int b[])//比较 a、b，若 a>b 为 1；若 a<b 为-1；若 a=b 为 0
{
    int i;
    if(a[0]>b[0])
        return 1;
    if(a[0]<b[0])
        return -1;
    for(i=a[0];i>0;i--)//从高位到低位比较
    {
        if(a[i]>b[i])
            return 1;
        if(a[i]<b[i])
            return -1;
    }
    return 0;
}

void subduction(int a[],int b[])//计算 a=a-b
{
    int flag;
    int i;

    flag=compare(a,b);
    if(flag==0)//相等
    {
        a[0]=0;
        return;
    }
    if(flag==1)//大于
    {
        for(i=1;i<=a[0];i++)
        {
            if(a[i]<b[i])//若不够向上借位
            {
                a[i+1]--;
                a[i]+=10;
            }
        }
    }
}

```

```

    }
    a[i]-=b[i];
}
while(a[0]>0&& a[a[0]]==0)//删除前导 0
    a[0]--;
return;
}
}
int main()
{
    char str1[100],str2[100];
    int i,j;

    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    memset(c,0,sizeof(c));

    cin>>str1>>str2;
    a[0]=strlen(str1);//a[0]存储串 1 的位数
    b[0]=strlen(str2);//b[0]存储串 2 的位数
    for(i=1;i<=a[0];i++)
        a[i]=str1[a[0]-i]-'0';
    for(i=1;i<=b[0];i++)
        b[i]=str2[b[0]-i]-'0';

    int temp[100];
    c[0]=a[0]-b[0]+1;
    for(i=c[0];i>0;i--)
    {
        memset(temp,0,sizeof(temp));

        for(j=1;j<=b[0];j++)//从 i 开始的地方，复制数组 b 到数组 temp
            temp[j+i-1]=b[j];
        temp[0]=b[0]+i-1;

        while(compare(a,temp)>=0)//用减法模拟
        {
            c[i]++;
            subduction(a,temp);
        }
    }

    while(c[0]>0&& c[c[0]]==0)//删除前导 0

```

```

        c[0]--;

//    cout<<"商为: ";
    if(c[0]==0)//输出结果
        cout<<0<<endl;
    else
    {
        for(i=c[0];i>0;i--)
            cout<<c[i];
        cout<<endl;
    }

//    cout<<"余数为: ";
    if(a[0]==0)//输出余数
        cout<<0<<endl;
    else
    {
        for(i=a[0];i>0;i--)
            cout<<a[i];
        cout<<endl;
    }

    return 0;
}
点赞 3

```

### 1.1.2.1.3 回文数（信息学奥赛一本通-T1309）

#### 回文数（信息学奥赛一本通-T1309）

原创 Alex\_McAvoy 发布于 2018-04-15 22:18:34 阅读数 1552 收藏  
展开

##### 【题目描述】

若一个数（首位不为零）从左向右读与从右向左读都是一样，我们就将其称之为回文数。例如：给定一个 10 进制数 56，将 56 加 65（即把 56 从右向左读），得到 121 是一个回文数。又如，对于 10 进制数 87：

STEP1:  $87 + 78 = 165$

STEP2:  $165 + 561 = 726$

STEP3:  $726 + 627 = 1353$

STEP4:  $1353+3531=4884$

在这里的一步是指进行了一次  $N$  进制的加法，上例最少用了 4 步得到回文数 4884。

写一个程序，给定一个  $N$  ( $2 \leq N \leq 10$  或  $N=16$ ) 进制数  $M$ 。求最少经过几步可以得到回文数。如果在 30 步以内(包含 30 步)不可能得到回文数，则输出“Impossible”。

**【输入】**

给定一个  $N$  ( $2 \leq N \leq 10$  或  $N=16$ ) 进制数  $M$ 。

**【输出】**

最少几步。如果在 30 步以内(包含 30 步)不可能得到回文数，则输出“Impossible”。

**【输入样例】**

9 87

**【输出样例】**

6

**【源程序】**

```
#include<iostream>
#include<cstring>
using namespace std;
int n;
int a[500],b[500],step=0;
char str[500];
void add(int a[],int b[])//高精加
{
    int i;
    for(i=1;i<=a[0];i++)
    {
        a[i]+=b[i];
        a[i+1]+=a[i]/n;
        a[i]%=n;
    }
    if(a[a[0]+1]>0)
        a[0]++;
}
bool judge(int a[])//判断回文数
{
    int i;
    for(i=1;i<=a[0]/2;i++)
        if(a[i]!=a[a[0]-i+1])
            return false;
    return true;
}
```

```

void overturn(int a[])//倒序存储
{
    int i,j;
    b[0]=a[0];
    j=1;
    for(i=a[0];i>=1;i--)
    {
        b[j]=a[i];
        j++;
    }
}
void print(int a[])//数据输出
{
    int i;
    for(i=a[0];i>=1;i--)
        cout<<a[i];
    cout<<endl;
}
int main()
{
    int len;
    int i;

    cin>>n>>str;
    len=strlen(str);

    int k=1;
    for(i=len-1;i>=0;i--)//数据处理
    {
        if('0'<=str[i]&&str[i]<='9')
        {
            a[k]=str[i]-'0';
            k++;
        }
        if('a'<=str[i]&&str[i]<='z')
        {
            a[k]=str[i]-'a'+10;
            k++;
        }
        if('A'<=str[i]&&str[i]<='Z')
        {
            a[k]=str[i]-'A'+10;
            k++;
        }
    }
}

```

```

    }

    a[0]=len;
    while(judge(a)==0)
    {
        overturn(a);
        add(a,b);
        step++;
        if(step>30)
            break;
    }

    if(step>30)
        cout<<"Impossible"<<endl;
    else
        cout<<step<<endl;
    return 0;
}

```

点赞

### 1.1.2.1.4 大整数加法（信息学奥赛一本通-T1168）

#### 大整数加法（信息学奥赛一本通-T1168）

原创 Alex\_McAvoy 发布于 2018-04-15 22:20:28 阅读数 3421 收藏  
展开

##### 【题目描述】

求两个不超过 200 位的非负整数的和。

##### 【输入】

有两行，每行是一个不超过 200 位的非负整数，可能有多余的前导 0。

##### 【输出】

一行，即相加后的结果。结果里不能有多余的前导 0，即如果结果是 342，那么就不能输出为 0342。

##### 【输入样例】

```

22222222222222222222
33333333333333333333

```

## 【输出样例】

55555555555555555555

## 【源程序】

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;
int main()
{
    string str1,str2;
    int a[210],b[210],c[210];
    int len1,len2;
    int i;

    cin>>str1;
    cin>>str2;
    len1=str1.length();
    len2=str2.length();

    /*补 0 工作，保证两个字符串相同*/
    if(len1<len2)
        for(i=1;i<=len2-len1;i++)
            str1="0"+str1;
    else
        for(i=1;i<=len1-len2;i++)
            str2="0"+str2;

    len1=str1.length();
    len2=str2.length();

    for(i=0;i<=len1-1;i++)//加数 str1 存入数组 a
        a[len1-i]=str1[i]-'0';
    for(i=0;i<=len2-1;i++)//加数 str2 存入数组 b
        b[len2-i]=str2[i]-'0';

    int x=0;
    int lenc=1;
    while( (lenc<=len1) || (lenc<=len2) )
    {
        c[lenc]=a[lenc]+b[lenc]+x;//第 i 位相加并加上次的进位
        x=c[lenc]/10;//向高位进位
        c[lenc]%=10;//存储第 i 位的值
        lenc++;//位置下标
    }
}
```

```
/*处理最高进位*/
c[lenc]=x;
while(c[lenc]==0)
    lenc--;

for(i=lenc;i>0;i--)//倒序输出
    cout<<c[i];
cout<<endl;

return 0;
}
```

#### 1.1.2.1.5 大整数减法（信息学奥赛一本通-T1169）

## 大整数减法（信息学奥赛一本通-T1169）

原创 Alex\_McAvoy 发布于 2018-04-15 22:21:17 阅读数 1951 收藏  
展开

### 【题目描述】

求两个大的正整数相减的差。

【输入】

共 2 行，第 1 行是被减数  $a$ ，第 2 行是减数  $b(a > b)$ 。每个大整数不超过 200 位，不会有多余的前导零。

【输出】

一行，即所求的差。

### 【输入样例】

999999999999999999999999999999999999  
9999999999999

### 【输出样例】

999999999999999999999999999999000000000000000

### 【源程序】

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;
int main()
```



```

{
    char str1[256],str2[256],temp[256];
    int a[256],b[256],c[256];
    int lena,lenb,lenc;
    int i;

    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    memset(c,0,sizeof(c));

    cin>>str1;//输入被减数
    cin>>str2;//输入减数

    lena=strlen(str1);
    lenb=strlen(str2);
    if( (lena<lenb) || (lena==lenb&&strcmp(str1,str2)) )//如果被减数小于减数,值为负,
    两者交换
    {
        strcpy(temp,str1);
        strcpy(str1,str2);
        strcpy(str2,temp);
        cout<<"-";//输出-
    }

    lena=strlen(str1);
    lenb=strlen(str2);

    for(i=0;i<=lena-1;i++)//被减数 str1 存入数组 a
        a[lena-i]=str1[i]-'0';
    for(i=0;i<=lenb-1;i++)//减数 str2 存入数组 b
        b[lenb-i]=str2[i]-'0';

    i=1;
    while(i<=lena||i<=lenb)
    {
        if(a[i]<b[i])
        {
            a[i]+=10;//借位
            a[i+1]--;//上一位减 1
        }
        c[i]=a[i]-b[i];//对应位相减
        i++;
    }
    lenc=i;

```

```

while((c[lenc]==0)&&(lenc>1))//删除前导 0
    lenc--;

for(i=lenc;i>=1;i--)//倒序输出
    cout<<c[i];
cout<<endl;
return 0;
}

```

### 1.1.2.1.6 计算 2 的 N 次方(信息学奥赛一本通-T1170)

#### 计算 2 的 N 次方（信息学奥赛一本通-T1170）

原创 Alex\_McAvoy 发布于 2018-04-15 22:29:07 阅读数 2674 收藏  
展开

##### 【题目描述】

任意给定一个正整数  $N(N \leq 100)$ ，计算 2 的  $n$  次方的值。

##### 【输入】

输入一个正整数  $N$ 。

##### 【输出】

输出 2 的  $N$  次方的值。

##### 【输入样例】

5

##### 【输出样例】

32

##### 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    int n;
    int i,j;
    int a[110];
    memset(a,0,sizeof(a));
    cin>>n;
    a[0]=1,a[1]=1;

```

```

for(i=1;i<=n;i++)
{
    for(j=1;j<=a[0];j++)//计算
        a[j]*=2;
    for(j=1;j<=a[0];j++)//转为高精加
    {
        a[j+1]+=a[j]/10;
        a[j]%=10;
    }
    if(a[a[0]+1]>0)//进位
        a[0]+=1;
}
for(i=a[0];i>=1;i--)//倒序输出
    cout<<a[i];
return 0;
}

```

点赞

### 1.1.2.1.7 大整数的因子（信息学奥赛一本通-T1171）

#### 大整数的因子（信息学奥赛一本通-T1171）

原创 Alex\_McAvoy 发布于 2018-04-15 22:29:10 阅读数 1748 收藏  
展开

##### 【题目描述】

已知正整数  $k$  满足  $2 \leq k \leq 9$ ，现给出长度最大为 30 位的十进制非负整数  $c$ ，求所有能整除  $c$  的  $k$ 。

##### 【输入】

一个非负整数  $c$ ， $c$  的位数  $\leq 30$ 。

##### 【输出】

若存在满足  $c \% k == 0$  的  $k$ ，从小到大输出所有这样的  $k$ ，相邻两个数之间用单个空格隔开；若没有这样的  $k$ ，则输出 "none"。

##### 【输入样例】

30

##### 【输出样例】

2 3 5 6

## 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int a[50],b[50];
char str[50];
int divide(int x)//高精除
{
    int i=a[0],res=0;
    while(i>0)
    {
        res*=10;
        res+=a[i];
        res%=x;
        i--;
    }
    return res;
}
int main()
{
    int len;
    bool flag=false;
    int i;

    cin>>str;
    len=strlen(str);

    int k=1;
    for(i=len-1;i>=0;i--)//数据处理
    {
        a[k]=str[i]-'0';
        k++;
    }

    a[0]=len;
    for(i=2;i<=9;i++)
    {
        if(divide(i)==false)//判断是否能整除，可以就输出
        {
            flag=true;
            cout<<i<<" ";
        }
    }
    if(flag==false)

```

```

        cout<<"none"<<endl;
    return 0;
}

```

点赞 2

### 1.1.2.1.8 求 10000 以内 $n$ 的阶乘（信息学奥赛一本通-T1172）

#### 求 10000 以内 $n$ 的阶乘（信息学奥赛一本通-T1172）

原创 Alex\_McAvoy 发布于 2018-04-15 22:19:22 阅读数 2971 收藏  
展开

##### 【题目描述】

求 10000 以内  $n$  的阶乘。

##### 【输入】

只有一行输入，整数  $n$  ( $0 \leq n \leq 10000$ )。

##### 【输出】

一行，即  $n!$  的值。

##### 【输入样例】

4

##### 【输出样例】

24

##### 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int a[100000];
int main()
{
    int n;
    int i,j;

    cin>>n;
    a[0]=1,a[1]=1;
    for(i=1;i<=n;i++)//高精乘
    {

```

```

for(j=1;j<=a[0];j++)
    a[j]*=i;
for(j=1;j<=a[0];j++)
{
    a[j+1]+=a[j]/10;
    a[j]%=10;
}
if(a[j]>0)//进位处理
{
    while(a[j]>10)
    {
        a[j+1]+=a[j]/10;
        a[j]%=10;
        j++;
    }
    a[0]=j;
}
}
for(i=a[0];i>=1;i--)
    cout<<a[i];
return 0;
}

```

点赞 4

### 1.1.2.1.9 阶乘和（信息学奥赛一本通-T1173）

#### 阶乘和（信息学奥赛一本通-T1173）

原创 Alex\_McAvoy 发布于 2018-04-16 22:30:49 阅读数 1997 收藏  
展开

##### 【题目描述】

用高精度计算出  $S=1!+2!+3!+\dots+n!$  ( $n\leq 50$ ), 其中“!”表示阶乘, 例如:  $5!=5\times 4\times 3\times 2\times 1$ 。

输入正整数  $n$ , 输出计算结果  $S$ 。

##### 【输入】

一个正整数  $n$ 。

##### 【输出】

计算结果  $S$ 。

## 【输入样例】

5

## 【输出样例】

153

## 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int a[500],sum[500];
void mul(int x)//高精乘
{
    int i;
    for(i=1;i<=a[0];i++)
        a[i]*=x;
    for(i=1;i<=a[0];i++)
    {
        a[i+1]+=a[i]/10;
        a[i]%=10;
    }

    i=a[0];
    while(a[i+1]>0)
        i++;

    a[0]=i;
    i=a[0];
    while(a[i]>10)
    {
        a[i+1]+=a[i]/10;
        a[i]%=10;
        i++;
    }
    a[0]=i;
}
void add()//高精加
{
    int i;
    if(sum[0]>a[0])
        sum[0]=sum[0];
    else
        sum[0]=a[0];
    for(i=1;i<=sum[0];i++)
    {

```

```

        sum[i]+=a[i];
        sum[i+1]+=sum[i]/10;
        sum[i]%=10;
    }
    if(sum[sum[0]+1]>0)
        sum[0]+=1;
}
int main()
{
    int n;
    int i;

    cin>>n;
    a[0]=1;a[1]=1;sum[0]=1;sum[1]=0;
    for(i=1;i<=n;i++)
    {
        mul(i);//计算阶乘
        add();//计算阶乘和
    }
    for(i=sum[0];i>=1;i--)
        cout<<sum[i];
    cout<<endl;
    return 0;
}

```

点赞 3

### 1.1.2.1.10 大整数乘法（信息学奥赛一本通-T1174）

#### 大整数乘法（信息学奥赛一本通-T1174）

原创 Alex\_McAvoy 发布于 2018-04-15 22:38:05 阅读数 1194 收藏  
展开

##### 【题目描述】

求两个不超过 200 位的非负整数的积。

##### 【输入】

有两行，每行是一个不超过 200 位的非负整数，没有多余的前导 0。



**【输出】**

一行，即相乘后的结果。结果里不能有多余的前导 0，即如果结果是 342，那么就不能输出为 0342。

**【输入样例】**

12345678900

98765432100

**【输出样例】**

1219326311126352690000

**【源程序】**

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;
int main()
{
    char str1[256],str2[256];
    int a[256],b[256],c[256];
    int lena,lenb,lenc;
    int x;
    int i,j;

    memset(a,0,sizeof(a));
    memset(b,0,sizeof(b));
    memset(c,0,sizeof(c));

    cin>>str1;//输入乘数 str1
    cin>>str2;//输入乘数 str2

    lena=strlen(str1);
    lenb=strlen(str2);
    for(i=0;i<=lena-1;i++)//乘数 str1 存入数组 a
        a[lena-i]=str1[i]-'0';
    for(i=0;i<=lenb-1;i++)//乘数 str2 存入数组 b
        b[lenb-i]=str2[i]-'0';

    for(i=1;i<=lenb;i++)
    {
        x=0;//用于存放进位
        for(j=1;j<=lena;j++)//对乘数每一位进行处理
        {
            c[i+j-1]=a[j]*b[i]+x+c[i+j-1];//当前乘积+上次乘积进位+原数
            x=c[i+j-1]/10;
```

```

        c[i+j-1]%=10;
    }
    c[i+lenc]=x;//进位
}
lenc=lenc+1;
while((c[lenc]==0)&&(lenc>1))//删除前导 0
    lenc--;
for(i=lenc;i>=1;i--)//倒序输出
    cout<<c[i];
cout<<endl;
return 0;
}

```

点赞

### 1.1.2.1.11 除以 13（信息学奥赛一本通-T1175）

#### 除以 13（信息学奥赛一本通-T1175）

原创 Alex\_McAvoy 发布于 2018-04-15 22:37:01 阅读数 1718 收藏  
展开

##### 【题目描述】

输入一个大于 0 的大整数 N，长度不超过 100 位，要求输出其除以 13 得到的商和余数。

##### 【输入】

一个大于 0 的大整数，长度不超过 100 位。

##### 【输出】

两行，分别为整数除法得到的商和余数。

##### 【输入样例】

2132104848488485

##### 【输出样例】

164008065268345  
0

##### 【源程序】

```

#include<iostream>
#include<cstring>

```

```

using namespace std;
int main()
{
    int a[110];
    char str[110];
    int len;
    int i,k=0,b=0,c;

    cin>>str;
    len=strlen(str);

    for(i=len-1;i>=0;i--)
        a[++k]=str[i]-'0';
    a[0]=len;

    i=a[0];//首项处理
    while(b<13)
    {
        b*=10;
        b+=a[i];
        i--;
    }
    cout<<b/13;

    b%=13;
    c=b;
    while(i>=1)//高精除
    {
        b*=10;
        b+=a[i];
        i--;
        cout<<b/13;
        b%=13;
        c=b;
    }
    cout<<endl;
    cout<<c<<endl;
    return 0;
}

```

点赞 2

## 1.1.2.2 第二章 数据排序

### 1.1.2.2.1 车厢重组（信息学奥赛一本通-T1310）

#### 车厢重组（信息学奥赛一本通-T1310）

原创 Alex\_McAvoy 发布于 2018-04-21 14:30:40 阅读数 2074 收藏  
展开

##### 【题目描述】

在一个旧式的火车站旁边有一座桥，其桥面可以绕河中心的桥墩水平旋转。一个车站的职工发现桥的长度最多能容纳两节车厢，如果将桥旋转  $180$  度，则可以把相邻两节车厢的位置交换，用这种方法可以重新排列车厢的顺序。于是他就负责用这座桥将进站的车厢按车厢号从小到大排列。他退休后，火车站决定将这一工作自动化，其中一项重要的工作是编一个程序，输入初始的车厢顺序，计算最少用多少步就能将车厢排序。

##### 【输入】

有两行数据，第一行是车厢总数  $N$ （不大于  $10000$ ），第二行是  $N$  个不同的数表示初始的车厢顺序。

##### 【输出】

一个数据，是最少的旋转次数。

##### 【输入样例】

```
4
4 3 2 1
```

##### 【输出样例】

```
6
```

##### 【源程序】

```
#include<iostream>
using namespace std;
int a[10100];
int main()
{
    int n;
    int i,j;
    int temp,sum=0;

    cin>>n;
    for(i=1;i<=n;i++)
        cin>>a[i];
    for(i=1;i<=n;i++)
```

```

    for(j=i+1;j<=n;j++)
        if(a[i]>a[j])
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
            sum++;
        }
    cout<<sum<<endl;
    return 0;
}

```

点赞

## 1.1.2.2.2 求逆序对（信息学奥赛一本通-T1311）

### 求逆序对（信息学奥赛一本通-T1311）

原创 Alex\_McAvoy 发布于 2018-04-21 14:32:11 阅读数 2752 收藏  
展开

#### 【题目描述】

给定一个序列  $a_1, a_2, \dots, a_n$ ，如果存在  $i < j$  并且  $a_i > a_j$ ，那么我们称之为逆序对，求逆序对的数目。

#### 【输入】

第一行为  $n$ ，表示序列长度，接下来的  $n$  行，第  $i+1$  行表示序列中的第  $i$  个数。

#### 【输出】

所有逆序对总数。

#### 【输入样例】

```

4
3
2
3
2

```

#### 【输出样例】

```

3

```

#### 【源程序】

```

#include<iostream>
#include<cstdio>

```

```

#include<cstring>
using namespace std;
int a[100100],n,b[100100];
long long sum=0;
void msort(int left,int right)
{
    if(left>=right)
        return;

    int mid=(left+right)/2;

    msort(left,mid);
    msort(mid+1,right);

    int i=left,j=mid+1,k=left;
    while(i<=mid&&j<=right)
    {
        if(a[i]>a[j])
        {
            b[k++]=a[j++];
            sum+=mid-i+1;
        }
        else
            b[k++]=a[i++];
    }
    while(i<=mid)
        b[k++]=a[i++];
    while(j<=right)
        b[k++]=a[j++];
    for(i=left;i<=right;i++)
        a[i]=b[i];
}
int main()
{
    int i,j;
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>a[i];
    msort(1,n);
    cout<<sum<<endl;
    return 0;
}

```

点赞 2

### 1.1.2.2.3 谁考了第 k 名（信息学奥赛一本通-T1176）

#### 谁考了第 k 名（信息学奥赛一本通-T1176）

原创 Alex\_McAvoy 发布于 2018-04-21 14:33:36 阅读数 3575 收藏  
展开

##### 【题目描述】

在一次考试中，每个学生的成绩都不相同，现知道了每个学生的学号和成绩，求考第 k 名学生的学号和成绩。

##### 【输入】

其后有 n 行数据，每行包括一个学号（整数）和一个成绩（浮点数），中间用一个空格分隔。

##### 【输出】

输出第 k 名学生的学号和成绩，中间用空格分隔。（注：请用 %g 输出成绩）

##### 【输入样例】

```
5 3
90788001 67.8
90788002 90.3
90788003 61
90788004 68.4
90788005 73.9
```

##### 【输出样例】

```
90788004 68.4
```

##### 【源程序】

```
#include<iostream>
using namespace std;
struct node{
    int id;
    double score;
}stu[110],temp;
int main()
{
    int n,k;
    int i,j;

    cin>>n>>k;
    for(i=1;i<=n;i++)
```

```

cin>>stu[i].id>>stu[i].score;

for(i=1;i<=n;i++)
    for(j=i+1;j<=n;j++)
        if(stu[i].score<stu[j].score)
        {
            temp=stu[i];
            stu[i]=stu[j];
            stu[j]=temp;
        }

cout<<stu[k].id<<" "<<stu[k].score;
return 0;
}

```

### 1.1.2.2.4 奇数单增序列（信息学奥赛一本通-T1177）

#### 奇数单增序列（信息学奥赛一本通-T1177）

原创 Alex\_McAvoy 发布于 2018-04-21 14:39:47 阅读数 2297 收藏  
展开

##### 【题目描述】

给定一个长度为  $N$ （不大于 500）的正整数序列，请将其中的所有奇数取出，并按升序输出。

##### 【输入】

第 1 行为  $N$ ；第 2 行为  $N$  个正整数，其间用空格间隔。

##### 【输出】

增序输出的奇数序列，数据之间以逗号间隔。数据保证至少有一个奇数。

##### 【输入样例】

```

10
1 3 2 6 5 4 9 8 7 10

```

##### 【输出样例】

```

1,3,5,7,9

```

##### 【源程序】

```

#include<iostream>
using namespace std;
int a[510];

```



```

int main()
{
    int n,b;
    int k=0;
    int temp;
    int i,j;

    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>b;
        if(b%2==1)
        {
            k++;
            a[k]=b;
        }
    }
    for(i=1;i<=k;i++)
        for(j=i+1;j<=k;j++)
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }

    cout<<a[1];
    for(i=2;i<=k;i++)
        cout<<" "<<a[i];
    return 0;
}

```

### 1.1.2.2.5 成绩排序（信息学奥赛一本通-T1178）

#### 成绩排序（信息学奥赛一本通-T1178）

原创 Alex\_McAvoy 发布于 2018-04-21 14:41:20 阅读数 3558 收藏  
展开

##### 【题目描述】

给出班里某门课程的成绩单，请你按成绩从高到低对成绩单排序输出，如果有相同分数则名字字典序小的在前。

**【输入】**

第一行为  $n$  ( $0 < n < 20$ ), 表示班里的学生数目;

接下来的  $n$  行, 每行为每个学生的名字和他的成绩, 中间用单个空格隔开。名字只包含字母且长度不超过 20, 成绩为一个不大于 100 的非负整数。

**【输出】**

把成绩单按分数从高到低的顺序进行排序并输出, 每行包含名字和分数两项, 之间有一个空格。

**【输入样例】**

```
4
Kitty 80
Hanmeimei 90
Joey 92
Tim 28
```

**【输出样例】**

```
Joey 92
Hanmeimei 90
Kitty 80
Tim 28
```

**【源程序】**

```
#include<iostream>
using namespace std;
#include <stdio.h>
#include <string.h>
struct node{
    char name[30];
    int score;
}stu[30],temp;
int main()
{
    int n;
    int i,j;

    cin>>n;
    for(i=1;i<=n;i++)
        cin>>stu[i].name>>stu[i].score;
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            if(stu[i].score<stu[j].score)
            {
                temp=stu[i];
                stu[i]=stu[j];
```

```

        stu[j]=temp;
    }
    else if(stu[i].score==stu[j].score&&strcmp(stu[i].name,stu[j].name)>0)
    {
        temp=stu[i];
        stu[i]=stu[j];
        stu[j]=temp;
    }
    for(i=1;i<=n;i++)
        cout<<stu[i].name<<" "<<stu[i].score<<endl;
    return 0;
}

```

点赞 1

### 1.1.2.2.6 奖学金（信息学奥赛一本通-T1179）

#### 奖学金（信息学奥赛一本通-T1179）

原创 Alex\_McAvoy 发布于 2018-04-21 14:44:03 阅读数 2738 收藏  
展开

##### 【题目描述】

某小学最近得到了一笔赞助，打算拿出其中一部分为学习成绩优秀的前 5 名学生发奖学金。期末，每个学生都有 3 门课的成绩:语文、数学、英语。先按总分从高到低排序，如果两个同学总分相同，再按语文成绩从高到低排序，如果两个同学总分和语文成绩都相同，那么规定学号小的同学排在前面，这样，每个学生的排序是唯一确定的。

任务：先根据输入的 3 门课的成绩计算总分，然后按上述规则排序，最后按排名顺序输出前五名名学生的学号和总分。注意，在前 5 名同学中，每个人的奖学金都不相同，因此，你必须严格按上述规则排序。例如，在某个正确答案中，如果前两行的输出数据(每行输出两个数:学号、总分) 是：

7 279

5 279

这两行数据的含义是:总分最高的两个同学的学号依次是 7 号、5 号。这两名同学的总分都是 279 (总分等于输入的语文、数学、英语三科成绩之和) ，但学号为 7 的学生语文成绩更高一些。如果你的前两名的输出数据是：

5 279

7 279

则按输出错误处理，不能得分。

### 【输入】

包含  $n+1$  行：

第 1 行为一个正整数  $n$ （小于 300），表示该校参加评选的学生人数。

第 2 到  $n+1$  行，每行有 3 个用空格隔开的数字，每个数字都在 0 到 100 之间。第  $j$  行的 3 个数字依次表示学号为  $j-1$  的学生的语文、数学、英语的成绩。每个学生的学号按照输入顺序编号为  $1\sim n$ （恰好是输入数据的行号减 1）。

### 【输出】

共有 5 行，每行是两个用空格隔开的正整数，依次表示前 5 名学生的学号和总分。

### 【输入样例】

```
6
90 67 80
87 66 91
78 89 91
88 99 77
67 89 64
78 89 98
```

### 【输出样例】

```
6 265
4 264
3 258
2 244
1 237
```

### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int i,j;
    int math,english,grade;
    int result[500][2]={0},chinese[500]={0};

    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>chinese[i]>>math>>english;
        grade=chinese[i]+math+english;
        result[i][0]=i;
```

```

        result[i][1]=grade;
    }

    for(i=1;i<=n-1;i++)
    {
        for(j=1;j<=n-i;j++)
        {
            if(result[j][1]<result[j+1][1])//按总分排序
            {
                swap(result[j][0],result[j+1][0]);
                swap(result[j][1],result[j+1][1]);
                swap(chinese[j],chinese[j+1]);
            }
            else if( result[j][1]==result[j+1][1] && chinese[j]<chinese[j+1])//按语文成绩排
序
            {
                swap(result[j][0],result[j+1][0]);
                swap(result[j][1],result[j+1][1]);
                swap(chinese[j],chinese[j+1]);
            }
            else if( result[j][1]==result[j+1][1] && chinese[j]==chinese[j+1] &&
result[j][0]>result[j+1][0] )//按学号排序
            {
                swap(result[j][0],result[j+1][0]);
                swap(result[j][1],result[j+1][1]);
                swap(chinese[j],chinese[j+1]);
            }
        }
    }

    for(i=1;i<=5;i++)
        cout<<result[i][0]<<" "<<result[i][1]<<endl;
    return 0;
}

```

点赞 2

### 1.1.2.2.7 分数线划定（信息学奥赛一本通-T1180）

#### 分数线划定（信息学奥赛一本通-T1180）

原创 Alex\_McAvoy 发布于 2018-04-21 14:48:25 阅读数 2271 收藏  
展开

**【题目描述】**

世博会志愿者的选拔工作正在 A 市如火如荼的进行。为了选拔最合适的人才，A 市对所有报名的选手进行了笔试，笔试分数达到面试分数线的选手方可进入面试。面试分数线根据计划录取人数的 150% 划定，即如果计划录取  $m$  名志愿者，则面试分数线为排名第  $m \times 150\%$ （向下取整）名的选手的分数，而最终进入面试的选手为笔试成绩不低于面试分数线的所有选手。

现在就请你编写程序划定面试分数线，并输出所有进入面试的选手的报名号和笔试成绩。

**【输入】**

第一行，两个整数  $n, m$  ( $5 \leq n \leq 5000, 3 \leq m \leq n$ )，中间用一个空格隔开，其中  $n$  表示报名参加笔试的选手总数， $m$  表示计划录取的志愿者人数。输入数据保证  $m \times 150\%$  向下取整后小于等于  $n$ 。

第二行到第  $n+1$  行，每行包括两个整数，中间用一个空格隔开，分别是选手的报名号  $k$  ( $1000 \leq k \leq 9999$ ) 和该选手的笔试成绩  $s$  ( $1 \leq s \leq 100$ )。数据保证选手的报名号各不相同。

**【输出】**

第一行，有两个整数，用一个空格隔开，第一个整数表示面试分数线；第二个整数为进入面试的选手的实际人数。

从第二行开始，每行包含两个整数，中间用一个空格隔开，分别表示进入面试的选手的报名号和笔试成绩，按照笔试成绩从高到低输出，如果成绩相同，则按报名号由小到大的顺序输出。

**【输入样例】**

```
6 3
1000 90
3239 88
2390 95
7231 84
1005 95
1001 88
```

**【输出样例】**

```
88 5
1005 95
2390 95
1000 90
1001 88
3239 88
```

**【源程序】**

```
#include<iostream>
```

```

#include<algorithm>
using namespace std;
int main()
{
    int n,m;
    int number[10000],score[10000];
    int i,j;
    int interview_score,interview_people=0;

    cin>>n>>m;
    for(i=0;i<n;i++)
        cin>>number[i]>>score[i];

    for(i=1;i<=n-1;i++)
    {
        for(j=0;j<=n-i;j++)
        {
            if(score[j]<score[j+1])//按面试分数排序
            {
                swap(score[j],score[j+1]);
                swap(number[j],number[j+1]);
            }
            if(score[j]==score[j+1])//成绩相同按报名号排序
            {
                if(number[j]>number[j+1])
                {
                    swap(score[j],score[j+1]);
                    swap(number[j],number[j+1]);
                }
            }
        }
    }

    m=double(m*1.5);//按计划录取人数的 150%划定选手
    interview_score=score[m-1];

    for(i=0;i<n;i++)
        if(score[i]>=interview_score)    interview_people++;
        else    break;

    cout<<interview_score<<" "<<interview_people<<endl;
    for(i=0;i<interview_people;i++)
        cout<<number[i]<<" "<<score[i]<<endl;
}

```

```

    return 0;
}

```

## 1.1.2.2.8 整数奇偶排序（信息学奥赛一本通-T1181）

### 整数奇偶排序（信息学奥赛一本通-T1181）

原创 Alex\_McAvoy 发布于 2018-04-21 14:46:52 阅读数 1690 收藏  
展开

#### 【题目描述】

给定 10 个整数的序列，要求对其重新排序。排序要求：

奇数在前，偶数在后；  
奇数按从大到小排序；  
偶数按从小到大排序。

#### 【输入】

输入一行，包含 10 个整数，彼此以一个空格分开，每个整数的范围是大于等于 0，小于等于 100。

#### 【输出】

按照要求排序后输出一行，包含排序后的 10 个整数，数与数之间以一个空格分开。

#### 【输入样例】

4 7 3 13 11 12 0 47 34 98

#### 【输出样例】

47 13 11 7 3 0 4 12 34 98

#### 【源程序】

```

#include<iostream>
using namespace std;
int main()
{
    int d;
    int a[20],b[20];
    int m=0,n=0;
    int temp;
    int i,j;

    while(cin>>d)
    {
        if(d%2==1)

```



```

        a[m++]=d;
    else
        b[n++]=d;
}

for(i=0;i<m;i++)
    for(j=i+1;j<m;j++)
        if(a[i]<a[j])
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }

for(i=0;i<n;i++)
    for(j=i+1;j<n;j++)
        if(b[i]>b[j])
        {
            temp=b[i];
            b[i]=b[j];
            b[j]=temp;
        }

for(i=0;i<m;i++)
    cout<<a[i]<<" ";
for(i=0;i<n;i++)
    cout<<b[i]<<" ";
cout<<endl;
return 0;
}

```

### 1.1.2.2.9 合影效果（信息学奥赛一本通-T1182）

#### 合影效果（信息学奥赛一本通-T1182）

原创 Alex\_McAvoy 发布于 2018-04-21 14:45:40 阅读数 2426 收藏  
展开

##### 【题目描述】

小云和朋友们去爬香山，为美丽的景色所陶醉，想合影留念。如果他们站成一排，男生全部在左（从拍照者的角度），并按照从矮到高的顺序从左到右排，女生全部在右，

并按照从高到矮的顺序从左到右排，请问他们合影的效果是什么样的（所有人的身高都不同）？

#### 【输入】

第一行是人数  $n$  ( $2 \leq n \leq 40$ ，且至少有 1 个男生和 1 个女生)。后面紧跟  $n$  行，每行输入一个人的性别（男 **male** 或女 **female**）和身高（浮点数，单位米），两个数据之间以空格分隔。

#### 【输出】

$n$  个浮点数，模拟站好队后，拍照者眼中从左到右每个人的身高。每个浮点数需保留到小数点后 2 位，相邻两个数之间用单个空格隔开。

#### 【输入样例】

```
6
male 1.72
male 1.78
female 1.61
male 1.65
female 1.70
female 1.56
```

#### 【输出样例】

```
1.65 1.72 1.78 1.70 1.61 1.56
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    int n;
    double d;
    char s[20];
    double a[50],b[50];
    int p=0,q=0;
    double temp;
    int i,j;

    cin>>n;
    for(i=0;i<n;i++)
    {
        cin>>s>>d;
        if(strcmp(s,"male")==0)
            a[p++]=d;
```

```

        else
            b[q++]=d;
    }

    for(i=0;i<p;i++)
        for(j=i+1;j<p;j++)
            if(a[i]>a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }

    for(i=0;i<q;i++)
        for(j=i+1;j<q;j++)
            if(b[i]<b[j])
            {
                temp=b[i];
                b[i]=b[j];
                b[j]=temp;
            }

    for(i=0;i<p;i++)
        printf("%.2lf ",a[i]);
    for(i=0;i<q;i++)
        printf("%.2lf ",b[i]);
    return 0;
}

```

### 1.1.2.2.10 病人排队（信息学奥赛一本通-T1183）

#### 病人排队（信息学奥赛一本通-T1183）

原创 Alex\_McAvoy 发布于 2018-04-22 21:31:22 阅读数 2925 收藏  
展开

##### 【题目描述】

病人登记看病，编写一个程序，将登记的病人按照以下原则排出看病的先后顺序：

- 1.老年人（年龄  $\geq 60$  岁）比非老年人优先看病。
- 2.老年人按年龄从大到小的顺序看病，年龄相同的按登记的先后顺序排序。

3.非老年人按登记的先后顺序看病。

**【输入】**

第 1 行，输入一个小于 100 的正整数，表示病人的个数；

后面按照病人登记的先后顺序，每行输入一个病人的信息，包括：一个长度小于 10 的字符串表示病人的 ID（每个病人的 ID 各不相同且只含数字和字母），一个整数表示病人的年龄，中间用单个空格隔开。

**【输出】**

按排好的看病顺序输出病人的 ID，每行一个。

**【输入样例】**

```
5
021075 40
004003 15
010158 67
021033 75
102012 30
```

**【输出样例】**

```
021033
010158
021075
004003
102012
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
struct node{
    char id[20];
    int age;
    int seq;
}a[110],b[110],temp;
int main()
{
    int n,d;
    char s[20];
    int p=0,q=0,k=0;
    int i,j;

    cin>>n;
    for(i=1;i<=n;i++)
```

```

{
    cin>>s>>d;
    k++;
    if(d>=60)
    {
        strcpy(a[p].id,s);
        a[p].age=d;
        a[p].seq=k;
        p++;
    }
    else
    {
        strcpy(b[q].id,s);
        b[q].age=d;
        b[q].seq=k;
        q++;
    }
}

for(i=0;i<p;i++)
    for(j=i+1;j<p;j++)
        if(a[i].age<a[j].age)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
        else if(a[i].age==a[j].age&& a[i].seq>a[j].seq)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }

for(i=0;i<q;i++)
    for(j=i+1;j<q;j++)
        if(b[i].seq>b[j].seq)
        {
            temp=b[i];
            b[i]=b[j];
            b[j]=temp;
        }

for(i=0;i<p;i++)

```

```

        cout<<a[i].id<<endl;
    for(i=0;i<q;i++)
        cout<<b[i].id<<endl;

    return 0;
}

```

点赞

### 1.1.2.2.11 明明的随机数（信息学奥赛一本通-T1184）

#### 明明的随机数（信息学奥赛一本通-T1184）

原创 Alex\_McAvoy 发布于 2018-04-22 21:32:52 阅读数 1958 收藏  
展开

##### 【题目描述】

明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了  $N$  个 1 到 1000 之间的随机整数 ( $N \leq 100$ )，对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

##### 【输入】

有 2 行，第 1 行为 1 个正整数，表示所生成的随机数的个数：  $N$ ；

第 2 行有  $N$  个用空格隔开的正整数，为所产生的随机数。

##### 【输出】

也是 2 行，第 1 行为 1 个正整数  $M$ ，表示不相同的随机数的个数。第 2 行为  $M$  个用空格隔开的正整数，为从小到大排好序的不相同的随机数。

##### 【输入样例】

```

10
20 40 32 67 40 20 89 300 400 15

```

##### 【输出样例】

```

8
15 20 32 40 67 89 300 400

```

##### 【源程序】

```

#include<iostream>
using namespace std;
int main()

```

```

{
    int n;
    int num[1001]={0},k;
    int result=0;
    int i;

    cin>>n;

    for(i=1;i<=n;i++) //桶排序
    {
        cin>>k;
        if(num[k]==0) result++;//去重
        num[k]++;
    }

    cout<<result<<endl;
    for(i=1;i<=1000;i++)
        if(num[i]) cout<<i<<" ";//桶号即随机数
    cout<<endl;
    return 0;
}

```

### 1.1.2.2.12 单词排序（信息学奥赛一本通-T1185）

#### 单词排序（信息学奥赛一本通-T1185）

原创 Alex\_McAvoy 发布于 2018-04-22 21:33:54 阅读数 3160 收藏  
展开

##### 【题目描述】

输入一行单词序列，相邻单词之间由 1 个或多个空格间隔，请按照字典序输出这些单词，要求重复的单词只输出一次。（区分大小写）

##### 【输入】

一行单词序列，最少 1 个单词，最多 100 个单词，每个单词长度不超过 50，单词之间用至少 1 个空格间隔。数据不含除字母、空格外的其他字符。

##### 【输出】

按字典序输出这些单词，重复的单词只输出一次。

##### 【输入样例】

She wants to go to Peking University to study Chinese

**【输出样例】**

Chinese  
Peking  
She  
University  
go  
study  
to  
wants

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<string>
#include<algorithm>
using namespace std;
int main()
{
    string a[100];
    int k=0;
    bool flag;
    int i;

    while(cin>>a[k])
    {
        flag=false;
        for(i=0;i<k;i++)
        {
            if(a[i].compare(a[k])==0)
            {
                flag=true;
                break;
            }
        }
        if(!flag)
            k++;
    }
    sort(a,a+k);
    for(i=0;i<k;i++)
        cout<<a[i]<<endl;
    return 0;
}
```



点赞

### 1.1.2.2.13 出现次数超过一半的数（信息学奥赛一本通-T1186）

#### 出现次数超过一半的数（信息学奥赛一本通-T1186）

原创 Alex\_McAvoy 发布于 2018-04-22 21:35:12 阅读数 2589 收藏  
展开

##### 【题目描述】

给出一个含有  $n$  ( $0 < n \leq 1000$ ) 个整数的数组，请找出其中出现次数超过一半的数。数组中的数大于-50 且小于 50。

##### 【输入】

第一行包含一个整数  $n$ ，表示数组大小；

第二行包含  $n$  个整数，分别是数组中的每个元素，相邻两个元素之间用单个空格隔开。

##### 【输出】

如果存在这样的数，输出这个数；否则输出 no。

##### 【输入样例】

```
3
1 2 2
```

##### 【输出样例】

```
2
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    int a[101]={0};
    int n,b;
    int i;
    bool flag=false;

    cin>>n;
```

```

for(i=0;i<n;i++)
{
    cin>>b;
    a[b+50]++;
}
for(i=0;i<100;i++)
    if(a[i]>=(n+1)/2)
    {
        flag=true;
        cout<<i-50<<endl;
    }
if(flag==0)
    cout<<"no";
cout<<endl;
return 0;
}

```

### 1.1.2.2.14 统计字符数（信息学奥赛一本通-T1187）

#### 统计字符数（信息学奥赛一本通-T1187）

原创 Alex\_McAvoy 发布于 2018-04-22 21:36:19 阅读数 2067 收藏  
展开

##### 【题目描述】

给定一个由 **a-z** 这 26 个字符组成的字符串，统计其中哪个字符出现的次数最多。

##### 【输入】

输入包含一行，一个字符串，长度不超过 1000。

##### 【输出】

输出一行，包括出现次数最多的字符和该字符出现的次数，中间以一个空格分开。如果有多个字符出现的次数相同且最多，那么输出 **ascii** 码最小的那一个字符。

##### 【输入样例】

abbccc

##### 【输出样例】

c 3

##### 【源程序】

```

#include<iostream>
#include<cstdio>

```

```

#include<cstring>
using namespace std;
char str[1100];
int main()
{
    int a[30]={0};
    int len,max=-1;
    int i,j;

    cin>>str;
    len=strlen(str);
    for(i=0;i<len;i++)
        a[str[i]-'a']++;

    for(i=0;i<26;i++)
        if(a[i]>max)
        {
            j=i;
            max=a[i];
        }

    cout<<char('a'+j)<<" "<<max<<endl;
    return 0;
}

```

点赞

### 1.1.2.3 第三章 递推算法

#### 1.1.2.3.1 昆虫繁殖（信息学奥赛一本通-T1312）

##### 昆虫繁殖（信息学奥赛一本通-T1312）

原创 Alex\_McAvoy 发布于 2018-04-25 20:50:19 阅读数 2660 收藏  
展开

##### 【题目描述】

科学家在热带森林中发现了一种特殊的昆虫，这种昆虫的繁殖能力很强。每对成虫过  $x$  个月产  $y$  对卵，每对卵要过两个月长成成虫。假设每个成虫不死，第一个月只有一

对成虫，且卵长成成虫后的第一个月不产卵(过  $X$  个月产卵)，问过  $Z$  个月以后，共有成虫多少对？ $0 \leq X \leq 20, 1 \leq Y \leq 20, X \leq Z \leq 50$ 。

**【输入】**

$x, y, z$  的数值。

**【输出】**

过  $Z$  个月以后，共有成虫对数。

**【输入样例】**

1 2 8

**【输出样例】**

37

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
long long a[60],b[60];
int main()
{
    int x,y,z;
    int i;

    cin>>x>>y>>z;
    for(i=1;i<=x;i++)
    {
        a[i]=1;//a[i]第 i 个月的成虫
        b[i]=0;//b[i]第 i 个月的卵
    }
    for(i=x+1;i<=z+1;i++)
    {
        b[i]=a[i-x]*y;//第 i-x 月的成虫在 x 个月后产下 y 个卵
        a[i]=a[i-1]+b[i-2]//第 i 个月的成虫等于第 i-1 个月的成虫数加上第 i-2 个月的卵
    }

    cout<<a[z+1]<<endl;
    return 0;
}
```

点赞 1

## 1.1.2.3.2 位数问题（信息学奥赛一本通-T1313）

### 位数问题（信息学奥赛一本通-T1313）

原创 Alex\_McAvoy 发布于 2018-04-25 20:51:58 阅读数 2129 收藏  
展开

#### 【题目描述】

在所有的  $N$  位数中，有多少个数中有偶数个数字 3？由于结果可能很大，你只需要输出这个答案对 12345 取余的值。

#### 【输入】

输入包含一行，一个字符串，长度不超过 1000。读入一个数  $N$ 。

#### 【输出】

输出有多少个数中有偶数个数字 3。

#### 【输入样例】

2

#### 【输出样例】

73

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main()
{
    int f[1001][2]; //f[i][0]表示前 i 位取偶数个 3 的情况，f[i][1]表示前 i 位取奇数个 3 的情况
    int n,x;
    int i;

    /*边界条件*/
    f[1][1]=1;
    f[1][0]=9;

    cin>>n;
    for(i=2;i<=n;i++)
    {
        x=f[1][0];
```

```

    if(i==n)
        x--;
    f[i][0]=(f[i-1][0]*x+f[i-1][1])%12345;
    f[i][1]=(f[i-1][1]*x+f[i-1][0])%12345;
}
cout<<f[n][0]<<endl;
return 0;
}

```

### 1.1.2.3.3 过河卒（信息学奥赛一本通-T1314）

#### 过河卒（信息学奥赛一本通-T1314）

原创 Alex\_McAvoy 发布于 2018-04-25 20:53:00 阅读数 1625 收藏  
展开

##### 【题目描述】

棋盘上 A 点有一个过河卒，需要走到目标 B 点。卒行走的规则：可以向下、或者向右。同时在棋盘上的某一点有一个对方的马（如 C 点），该马所在的点和所有跳跃一步可达的点称为对方马的控制点，如图 3-1 中的 C 点和 P1，……，P8，卒不能通过对方马的控制点。棋盘用坐标表示，A 点(0,0)、B 点(n, m) (n,m 为不超过 20 的整数),同样马的位置坐标是需要给出的，C≠A 且 C≠B。现在要求你计算出卒从 A 点能够到达 B 点的路径的条数。

##### 【输入】

给出 n、m 和 C 点的坐标。

##### 【输出】

从 A 点能够到达 B 点的路径的条数。

##### 【输入样例】

8 6 0 4

##### 【输出样例】

1617

##### 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
long long a[30][30];

```

```

int vis[30][30];
int next[][2]={{2,1},{1,2},{-1,2},{-2,1},{-2,-1},{-1,-2},{1,-2},{2,-1}};
int main()
{
    int n,m;
    int x,y;
    int nx,ny;
    int i,j;

    memset(vis,0,sizeof(vis));
    cin>>n>>m>>x>>y;

    a[0][0]=0;//处理 A=B 的情况
    vis[x][y]=1;//设置马管辖的位置
    a[x][y]=0;
    for(i=0;i<8;i++)
    {
        nx=x+next[i][0];
        ny=y+next[i][1];
        if(0<=nx&&nx<=n&&0<=ny&&ny<=m)
        {
            vis[nx][ny]=1;
            a[nx][ny]=0;
        }
    }
    for(i=0;i<=n;i++)
    {
        if(vis[i][0]==1)
            while(i<=n)
            {
                i++;
                a[i][0]=0;
            }
        else
            a[i][0]=1;
    }
    for(j=0;j<=m;j++)
    {
        if(vis[0][j]==1)
            while(j<=m)
            {
                j++;
                a[0][j]=0;
            }
    }
}

```

```

else
    a[0][j]=1;
}
for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
        if(vis[i][j]==0)
            a[i][j]=a[i][j-1]+a[i-1][j];
cout<<a[n][m]<<endl;
return 0;
}

```

点赞 1

### 1.1.2.3.4 菲波那契数列（信息学奥赛一本通-T1188）

#### 菲波那契数列（信息学奥赛一本通-T1188）

原创 Alex\_McAvoy 发布于 2018-04-25 20:55:07 阅读数 3021 收藏  
展开

##### 【题目描述】

菲波那契数列是指这样的数列：数列的第一个和第二个数都为 1，接下来每个数都等于前面 2 个数之和。

给出一个正整数  $a$ ，要求菲波那契数列中第  $a$  个数对 1000 取模的结果是多少。

##### 【输入】

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $a(1 \leq a \leq 1000000)$ 。

##### 【输出】

$n$  行，每行输出对应一个输入。输出应是一个正整数，为菲波那契数列中第  $a$  个数对 1000 取模得到的结果。

##### 【输入样例】

```

4
5
2
19
1

```



## 【输出样例】

```
?5
1
181
1
```

## 【源程序】

```
#include<iostream>
using namespace std;
int a[1000100];
int main()
{
    int n,x,i;
    a[1]=1;
    a[2]=1;
    for(i=3;i<=1000000;i++)
        a[i]=(a[i-1]+a[i-2])%1000;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>x;
        cout<<a[x]<<endl;
    }
    return 0;
}
```

## 1.1.2.3.5 Pell 数列（信息学奥赛一本通-T1189）

### Pell 数列（信息学奥赛一本通-T1189）

原创 Alex\_McAvoy 发布于 2018-04-25 20:56:14 阅读数 1899 收藏  
展开

## 【题目描述】

Pell 数列  $a_1, a_2, a_3, \dots$  的定义是这样的， $a_1=1, a_2=2, \dots, a_n=2a_{n-1}+a_{n-2} (n>2)$ 。

给出一个正整数  $k$ ，要求 Pell 数列的第  $k$  项模上 32767 是多少。

## 【输入】

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $k (1 \leq k < 1000000)$ 。

**【输出】**

$n$  行，每行输出对应一个输入。输出应是一个非负整数。

**【输入样例】**

```
2
1
8
```

**【输出样例】**

```
1
408
```

**【源程序】**

```
#include<iostream>
using namespace std;
int a[1000100];
int main()
{
    int n,k,i;
    a[1]=1;
    a[2]=2;
    for(i=3;i<=1000000;i++)
        a[i]=(2*a[i-1]+a[i-2])%32767;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>k;
        cout<<a[k]<<endl;
    }
    return 0;
}
```

## 1.1.2.3.6 上台阶（信息学奥赛一本通-T1190）

### 上台阶（信息学奥赛一本通-T1190）

原创 Alex\_McAvoy 发布于 2018-05-03 22:45:35 阅读数 3570 收藏  
展开

**【题目描述】**

楼梯有  $n$  ( $71 > n > 0$ ) 阶台阶,上楼时可以一步上 1 阶,也可以一步上 2 阶,也可以一步上 3 阶,编程计算共有多少种不同的走法。

**【输入】**

输入的每一行包括一组测试数据，即为台阶数  $n$ 。最后一行为 0，表示测试结束。

**【输出】**

每一行输出对应一行输入的结果，即为走法的数目。

**【输入样例】**

```
1
2
3
4
0
```

**【输出样例】**

```
1
2
4
7
```

**【源程序】**

```
#include<stdio>
long long d[110]= {0};
int main()
{
    d[1]=1;
    d[2]=2;
    d[3]=4;
    for(int i=4; i<=100; i++)
        d[i]=d[i-1]+d[i-2]+d[i-3];
    int a;
    while(scanf("%d",&a)==1&&a)
        printf("%lld\n",d[a]);
    return 0;
}
```

## 1.1.2.3.7 流感传染（信息学奥赛一本通-T1191）

### 流感传染（信息学奥赛一本通-T1191）

原创 Alex\_McAvoy 发布于 2018-05-03 22:42:19 阅读数 2635 收藏  
展开

**【题目描述】**

有一批易感人群住在网格状的宿舍区内，宿舍区为  $n*n$  的矩阵，每个格点为一个房间，房间里可能住人，也可能空着。在第一天，有些房间里的人得了流感，以后每天，得流感的人会使其邻居传染上流感，（已经得病的不变），空房间不会传染。请输出第  $m$  天得流感的人数。

**【输入】**

第一行一个数字  $n$ ， $n$  不超过 100，表示有  $n*n$  的宿舍房间。

接下来的  $n$  行，每行  $n$  个字符，'.'表示第一天该房间住着健康的人，'#'表示该房间空着，'@'表示第一天该房间住着得流感的人。

接下来一行是一个整数  $m$ ， $m$  不超过 100。

**【输出】**

输出第  $m$  天，得流感的人数。

**【输入样例】**

```
5
...#
.#.@.
.#@..
#....
.....
4
```

**【输出样例】**

```
16
```

**【源程序】**

```
#include<iostream>
#include<cstring>
using namespace std;
char s[110];
int a[110][110];
struct node{
    int r;//r 行
    int c;//c 列
    int d;//d 天
}q[10100];
int nextt[][2]={{-1,0},{1,0},{0,-1},{0,1}};
int main()
{
    int n,m;
    int h=1,t=1;
    int r,c,d;
    int nr,nc,cnt=0;
    int i,j;
```

```

cin>>n;
for(i=0;i<n;i++)
{
    cin>>s;
    for(j=0;j<n;j++)
        if(s[j]=='.')
            a[i][j]=0;
        else if(s[j]=='@')
        {
            a[i][j]=1;
            q[t].r=i;
            q[t].c=j;
            q[t].d=1;
            t++;
        }
        else if(s[j]=='#')
            a[i][j]=2;
    }
cin>>m;
while(h<t)
{
    r=q[h].r;
    c=q[h].c;
    d=q[h].d;
    if(d==m)
        break;
    for(i=0;i<4;i++)
    {
        nr=r+nextt[i][0];
        nc=c+nextt[i][1];
        if(a[nr][nc]==0&&0<=nr&&nr<n&&0<=nc&&nc<n)
        {
            q[t].r=nr;
            q[t].c=nc;
            q[t].d=d+1;
            a[nr][nc]=1;
            t++;
        }
    }
    h++;
}
for(i=0;i<n;i++)
    for(j=0;j<n;j++)
        if(a[i][j]==1)

```

```

        cnt++;
    cout<<cnt<<endl;
    return 0;
}

```

点赞 1

### 1.1.2.3.8 放苹果（信息学奥赛一本通-T1192）

#### 放苹果（信息学奥赛一本通-T1192）

原创 Alex\_McAvoy 发布于 2018-05-03 22:31:37 阅读数 2642 收藏  
展开

##### 【题目描述】

把  $M$  个同样的苹果放在  $N$  个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？（用  $K$  表示）5，1，1 和 1，5，1 是同一种分法。

##### 【输入】

第一行是测试数据的数目  $t$  ( $0 \leq t \leq 20$ )。以下每行均包含二个整数  $M$  和  $N$ ，以空格分开。 $1 \leq M, N \leq 10$ 。

##### 【输出】

对输入的每组数据  $M$  和  $N$ ，用一行输出相应的  $K$ 。

##### 【输入样例】

```

1
7 3

```

##### 【输出样例】

```

8

```

##### 【源程序】

```

#include<iostream>
#include<cstring>
using namespace std;
int a[20][20];
int f(int m,int n)
{
    int i,j;
    for(i=1;i<=n;i++)//0 个苹果
        a[0][i]=1;
}

```

```

for(i=1;i<=m;i++)//1 个盘子
    a[i][1]=1;
for(i=1;i<=m;i++)
    for(j=2;j<=n;j++)
        if(i<j)
            a[i][j]=a[i][i];
        else
            a[i][j]=a[i][j-1]+a[i-j][j];
}
int main()
{
    int m,n,i,j,k;
    cin>>k;
    for(i=1;i<=k;i++)
    {
        cin>>m>>n;
        f(m,n);
        cout<<a[m][n]<<endl;
    }
    return 0;
}

```

### 1.1.2.3.9 吃糖果（信息学奥赛一本通-T1193）

#### 吃糖果（信息学奥赛一本通-T1193）

原创 Alex\_McAvoy 发布于 2018-05-04 16:08:14 阅读数 1101 收藏  
展开

##### 【题目描述】

名名的妈妈从外地出差回来，带了一盒好吃又精美的巧克力给名名(盒内共有  $N$  块巧克力， $0 < N < 20$ )。妈妈告诉名名每天可以吃一块或者两块巧克力。假设名名每天都吃巧克力，问名名共有多少种不同的吃完巧克力的方案。例如：如果  $N=1$ ，则名名第 1 天就吃掉它，共有 1 种方案；如果  $N=2$ ，则名名可以第 1 天吃 1 块，第 2 天吃 1 块，也可以第 1 天吃 2 块，共有 2 种方案；如果  $N=3$ ，则名名第 1 天可以吃 1 块，剩 2 块，也可以第 1 天吃 2 块剩 1 块，所以名名共有  $2+1=3$  种方案；如果  $N=4$ ，则名名可以第 1 天吃 1 块，剩 3 块，也可以第 1 天吃 2 块，剩 2 块，共有  $3+2=5$  种方案。现在给定  $N$ ，请你写程序求出名名吃巧克力的方案数目。

##### 【输入】

输入只有 1 行，即整数  $N$ 。

##### 【输出】

输出只有 1 行，即名名吃巧克力的方案数。

## 【输入样例】

4

## 【输出样例】

5

## 【源程序】

```
#include<iostream>
using namespace std;
int a[30];
int main()
{
    int n,i;
    a[1]=1;
    a[2]=1+a[2-1];
    for(i=3;i<20;i++)
        a[i]=a[i-1]+a[i-2];
    cin>>n;
    cout<<a[n]<<endl;
    return 0;
}
```

### 1.1.2.3.10 移动路线（信息学奥赛一本通-T1194）

#### 移动路线（信息学奥赛一本通-T1194）

原创 Alex\_McAvoy 发布于 2018-05-04 16:11:24 阅读数 1667 收藏  
展开

## 【题目描述】

X 桌子上有一个  $m$  行  $n$  列的方格矩阵，将每个方格用坐标表示，行坐标从下到上依次递增，列坐标从左至右依次递增，左下角方格的坐标为(1,1)，则右上角方格的坐标为(m,n)。

小明是个调皮的孩子，一天他捉来一只蚂蚁，不小心把蚂蚁的右脚弄伤了，于是蚂蚁只能向上或向右移动。小明把这只蚂蚁放在左下角的方格中，蚂蚁从左下角的方格中移动到右上角的方格中，每步移动一个方格。蚂蚁始终在方格矩阵内移动，请计算出不同的移动路线的数目。

对于 1 行 1 列的方格矩阵，蚂蚁原地移动，移动路线数为 1；对于 1 行 2 列（或 2 行 1 列）的方格矩阵，蚂蚁只需一次向右（或向上）移动，移动路线数也为 1.....对于一个 2 行 3 列的方格矩阵，如下所示：



(2,1) (2,2) (2,3)

(1,1) (1,2) (1,3)

蚂蚁共有 3 种移动路线:

路线 1: (1,1) → (1,2) → (1,3) → (2,3)

路线 2: (1,1) → (1,2) → (2,2) → (2,3)

路线 3: (1,1) → (2,1) → (2,2) → (2,3)

#### 【输入】

输入只有一行，包括两个整数  $m$  和  $n$  ( $0 < m+n \leq 20$ )，代表方格矩阵的行数和列数， $m$ 、 $n$  之间用空格隔开。

#### 【输出】

输出只有一行，为不同的移动路线的数目。

#### 【输入样例】

2 3

#### 【输出样例】

3

#### 【源程序】

```
#include<iostream>
using namespace std;
int a[30][30];
int main()
{
    int m,n;
    int i,j;
    cin>>m>>n;
    for(i=1;i<=m;i++)
        a[i][1]=1;
    for(j=1;j<=n;j++)
        a[1][j]=1;
    for(i=2;i<=m;i++)
        for(j=2;j<=n;j++)
            a[i][j]=a[i-1][j]+a[i][j-1];
    cout<<a[m][n]<<endl;
    return 0;
}
```

### 1.1.2.3.11 判断整除（信息学奥赛一本通-T1195）

#### 判断整除（信息学奥赛一本通-T1195）

原创 Alex\_McAvoy 发布于 2018-05-04 16:17:50 阅读数 2692 收藏  
展开

##### 【题目描述】

一个给定的正整数序列，在每个数之前都插入+号或-号后计算它们的和。比如序列：  
1、2、4 共有 8 种可能的序列：

$$(+1) + (+2) + (+4) = 7$$

$$(+1) + (+2) + (-4) = -1$$

$$(+1) + (-2) + (+4) = 3$$

$$(+1) + (-2) + (-4) = -5$$

$$(-1) + (+2) + (+4) = 5$$

$$(-1) + (+2) + (-4) = -3$$

$$(-1) + (-2) + (+4) = 1$$

$$(-1) + (-2) + (-4) = -7$$

所有结果中至少有一个可被整数  $k$  整除，我们则称此正整数序列可被  $k$  整除。例如上述序列可以被 3、5、7 整除，而不能被 2、4、6、8……整除。注意：0、-3、-6、-9……都可以认为是 3 的倍数。

##### 【输入】

输入的第一行包含两个数： $N(2 < N < 10000)$  和  $k(2 < k < 100)$ ，其中  $N$  代表一共有  $N$  个数， $k$  代表被除数。第二行给出序列中的  $N$  个整数，这些整数的取值范围都 0 到 10000 之间（可能重复）。

##### 【输出】

如果此正整数序列可被  $k$  整除，则输出 YES，否则输出 NO。（注意：都是大写字母）

##### 【输入样例】

```
3 2
1 2 4
```

##### 【输出样例】

```
NO
```

**【源程序】**

```

#include<iostream>
#include<cstring>
using namespace std;
int a[10100];
int dp[10100][110];
int main()
{
    int n,k,i,j;
    memset(dp,0,sizeof(dp));

    cin>>n>>k;
    for(i=1;i<=n;i++)
        cin>>a[i];
    dp[0][0]=1;
    for(i=1;i<=n;i++)
        for(j=0;j<k;j++)
            dp[i][j]=dp[i-1][(j+a[i])%k]||dp[i-1][(j-a[i]%k+k)%k];
    if(dp[n][0]==0)
        cout<<"NO"<<endl;
    else
        cout<<"YES"<<endl;
    return 0;
}

```

点赞

## 1.1.2.3.12 踩方格（信息学奥赛一本通-T1196）

### 踩方格（信息学奥赛一本通-T1196）

原创 Alex\_McAvoy 发布于 2018-05-04 16:20:06 阅读数 1933 收藏  
展开

**【题目描述】**

有一个方格矩阵，矩阵边界在无穷远处。我们做如下假设：

- a、每走一步时，只能从当前方格移动一格，走到某个相邻的方格上；
- b、走过的格子立即塌陷无法再走第二次；
- c、只能向北、东、西三个方向走；

请问：如果允许在方格矩阵上走  $n$  步，共有多少种不同的方案。2 种走法只要有一步不一样，即被认为是不同的方案。

**【输入】**

允许在方格上行走的步数  $n(n \leq 20)$ 。

**【输出】**

计算出的方案数量。

**【输入样例】**

2

**【输出样例】**

7

**【源程序】**

```
#include<iostream>
#include<cstring>
using namespace std;
int a[30];
int main()
{
    int n,i;
    cin>>n;
    a[1]=3;
    a[2]=7;
    for(i=3;i<=n;i++)
        a[i]=2*a[i-1]+a[i-2];
    cout<<a[n]<<endl;
    return 0;
}
```

### 1.1.2.3.13 山区建小学（信息学奥赛一本通-T1197）

#### 山区建小学（信息学奥赛一本通-T1197）

原创 Alex\_McAvoy 发布于 2018-05-04 16:26:11 阅读数 1856 收藏  
展开

**【题目描述】**

政府在某山区修建了一条道路，恰好穿越总共  $m$  个村庄的每个村庄一次，没有回路或交叉，任意两个村庄只能通过这条路来往。已知任意两个相邻的村庄之间的距离为  $d_i$  ( $d_i$  为正整数)，其中， $0 < i < m$ 。为了提高山区的文化素质，政府又决定从  $m$  个村中选

择  $n$  个村建小学(设  $0 < n \leq m < 500$ )。请根据给定的  $m$ 、 $n$  以及所有相邻村庄的距离, 选择在哪些村庄建小学, 才使得所有村到最近小学的距离总和最小, 计算最小值。

### 【输入】

第 1 行为  $m$  和  $n$ , 其间用空格间隔

2 行为  $m-1$  个整数, 依次表示从一端到另一端的相邻村庄的距离, 整数之间以空格间隔。

例如:

10 3

2 4 6 5 2 4 3 1 3

表示在 10 个村庄建 3 所学校。第 1 个村庄与第 2 个村庄距离为 2, 第 2 个村庄与第 3 个村庄距离为 4, 第 3 个村庄与第 4 个村庄距离为 6, ..., 第 9 个村庄到第 10 个村庄的距离为 3。

### 【输出】

各村庄到最近学校的距离之和的最小值。

### 【输入样例】

10 2  
3 1 3 1 1 1 1 1 3

### 【输出样例】

18

### 【源程序】

```
#include<iostream>
#include<cstring>
#include<algorithm>
using namespace std;
int dis[510];
int a[510][510];
int dp[510][510];
int main()
{
    int m,n;
    int i,j,k;

    memset(a,0,sizeof(a));

    cin>>m>>n;
    dis[1]=0;
    for(i=2;i<=m;i++)
```

```

{
    cin>>dis[i];
    dis[i]+=dis[i-1];
}

for(i=1;i<=m;i++)//学校建在 i 村 j 村的中间村落
    for(j=i+1;j<=m;j++)
        a[i][j]=a[i][j-1]+dis[j]-dis[(i+j)/2];

for(i=1;i<=m;i++)
    for(j=1;j<=i&&j<=n;j++)
        dp[i][j]=999999999;
for(i=1;i<=m;i++)//dp[i][j]前 i 个村建 j 所学校,最短距离
    dp[i][1]=a[1][i];

for(i=2;i<=m;i++)
    for(j=2;j<=i&&j<=n;j++)
        for(k=j-1;k<i;k++)
            dp[i][j]=min(dp[i][j],dp[k][j-1]+a[k+1][i]);

cout<<dp[m][n]<<endl;
return 0;
}

```

## 1.1.2.4 第四章 递归算法

### 1.1.2.4.1 逆波兰表达式（信息学奥赛一本通-T1198）

#### 逆波兰表达式（信息学奥赛一本通-T1198）

原创 Alex\_McAvoy 发布于 2018-05-09 23:45:05 阅读数 2559 收藏  
展开

##### 【题目描述】

逆波兰表达式是一种把运算符前置的算术表达式，例如普通的表达式  $2 + 3$  的逆波兰表示法为  $+ 2 3$ 。逆波兰表达式的优点是运算符之间不必有优先级关系，也不必用括号改变运算次序，例如  $(2 + 3) * 4$  的逆波兰表示法为  $* + 2 3 4$ 。本题求解逆波兰表达式的值，其中运算符包括  $+ - * /$  四个。

**【输入】**

输入为一行，其中运算符和运算数之间都用空格分隔，运算数是浮点数。

**【输出】**

输出为一行，表达式的值。

可直接用 `printf("%f\n", v)` 输出表达式的值 `v`。

**【输入样例】**

```
* + 11.0 12.0 + 24.0 35.0
```

**【输出样例】**

```
1357.000000
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
using namespace std;
char a[55];
double calculate()
{
    scanf("%s",a);
    if(a[0]=='+')
        return calculate()+calculate();
    else if(a[0]=='-')
        return calculate()-calculate();
    else if(a[0]=='*')
        return calculate()*calculate();
    else if(a[0]=='/')
        return calculate()/calculate();
    else
        return atof(a);
}
int main()
{
    printf("%f\n",calculate());
    return 0;
}
```

## 1.1.2.4.2 全排列（信息学奥赛一本通-T1199）

### 全排列（信息学奥赛一本通-T1199）

原创 Alex\_McAvoy 发布于 2018-05-09 23:47:50 阅读数 2922 收藏  
展开

### 【题目描述】

给定一个由不同的小写字母组成的字符串，输出这个字符串的所有全排列。  
我们假设对于小写字母有'a' < 'b' < ... < 'y' < 'z'，而且给定的字符串中的字母已经按照从小到大的顺序排列。

### 【输入】

只有一行，是一个由不同的小写字母组成的字符串，已知字符串的长度在 1 到 6 之间。

### 【输出】

输出这个字符串的所有排列方式，每行一个排列。要求字母序比较小的排列在前面。  
字母序如下定义：

已知  $S=s_1, s_2 \dots s_k, T=t_1, t_2 \dots t_k$ ，则  $S < T$  等价于，存在  $p(1 \leq p \leq k)$ ，使得  $s_1=t_1, s_2=t_2, \dots, s_{p-1}=t_{p-1}, s_p < t_p$  成立。

### 【输入样例】

abc

### 【输出样例】

abc  
acb  
bac  
bca  
cab  
cba

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
using namespace std;
char str[10],temp[10];
int n;
bool vis[10];
void dfs(int step)
{
    if(step==n)
    {
        temp[step]='\0';
        cout<<temp<<endl;
    }
}
```



```

for(int i=0;i<n;i++)
    if(vis[i]==0)
    {
        vis[i]=1;
        temp[step]=str[i];
        dfs(step+1);
        vis[i]=0;
    }
}
int main()
{
    memset(vis,0,sizeof(vis));
    scanf("%s",str);
    n=strlen(str);
    dfs(0);
    return 0;
}

```

点赞 1

### 1.1.2.4.3 分解因数（信息学奥赛一本通-T1200）

#### 分解因数（信息学奥赛一本通-T1200）

原创 Alex\_McAvoy 发布于 2018-05-09 23:56:20 阅读数 2204 收藏  
展开

##### 【题目描述】

给出一个正整数  $a$ ，要求分解成若干个正整数的乘积，即  $a=a_1 \times a_2 \times a_3 \times \dots \times a_n$ ，并且  $1 < a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n$ ，问这样的分解的种数有多少。注意到  $a=a$  也是一种分解。

##### 【输入】

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $a$  ( $1 < a < 32768$ )。

##### 【输出】

$n$  行，每行输出对应一个输入。输出应是一个正整数，指明满足要求的分解的种数。

##### 【输入样例】

2

2  
20

【输出样例】

1  
4

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
int a[32768];
int tot;
int x;
void calculate(int n,int step)
{
    int ans=1;
    for(int i=1;i<=step-1;i++)
        ans*=a[i];
    if(ans>x)
        return;
    if(ans==x)
    {
        tot++;
        return;
    }
    for(int i=a[step-1];i<=x;i++)
        if(n%i==0)
        {
            n/=i;
            a[step]=i;
            calculate(n,step+1);
            n*=i;
        }
}
int main()
{
    int n;
    cin>>n;
    while(n--)
    {
        tot=0;
```

```

    cin>>x;
    a[0]=2;
    calculate(x,1);
    cout<<tot<<endl;
}
return 0;
}

```

点赞

## 1.1.2.4.4 菲波那契数列（信息学奥赛一本通-T1201）

### 菲波那契数列（信息学奥赛一本通-T1201）

原创 Alex\_McAvoy 发布于 2018-05-09 23:57:26 阅读数 1964 收藏  
展开

#### 【题目描述】

菲波那契数列是指这样的数列：数列的第一个和第二个数都为 1，接下来每个数都等于前面 2 个数之和。

给出一个正整数  $a$ ，要求菲波那契数列中第  $a$  个数是多少。

#### 【输入】

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $a(1 \leq a \leq 20)$ 。

#### 【输出】

输出有  $n$  行，每行输出对应一个输入。输出应是一个正整数，为菲波那契数列中第  $a$  个数的大小。

#### 【输入样例】

```

?4
5
2
19
1

```

#### 【输出样例】

```

?5
1
4181
1

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
using namespace std;
int Fibonacci(int n)
{
    if(n==1)
        return 1;
    if(n==2)
        return 1;
    return Fibonacci(n-1)+Fibonacci(n-2);
}
int main()
{
    int n,x;
    cin>>n;
    while(n--)
    {
        cin>>x;
        cout<<Fibonacci(x)<<endl;
    }
    return 0;
}

```

点赞

## 1.1.2.4.5 Pell 数列（信息学奥赛一本通-T1202）

### Pell 数列（信息学奥赛一本通-T1202）

原创 Alex\_McAvoy 发布于 2018-05-09 23:59:43 阅读数 2386 收藏  
展开

**【题目描述】**

Pell 数列  $a_1, a_2, a_3, \dots$  的定义是这样的， $a_1=1, a_2=2, \dots, a_n=2a_{n-1}+a_{n-2} (n>2)$ 。  
给出一个正整数  $k$ ，要求 Pell 数列的第  $k$  项模上 32767 是多少。

**【输入】**

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $k (1 \leq k < 1000000)$ 。

**【输出】**

n 行，每行输出对应一个输入。输出应是一个非负整数。

**【输入样例】**

2  
1  
8

**【输出样例】**

1  
408

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
#define k 32767
using namespace std;
int a[N];
int f(int x)
{
    if(a[x]!=0) return a[x];
    if(x==1) return 1;
    if(x==2) return 2;
    return a[x]=(2*(f(x-1))%k+(f(x-2))%k)%k;
}
int main()
{
    int n,a;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a;
        cout<<f(a)<<endl;
    }
    return 0;
}
```

点赞

## 1.1.2.4.6 扩号匹配问题（信息学奥赛一本通-T1203）

### 扩号匹配问题（信息学奥赛一本通-T1203）

原创 Alex\_McAvoy 发布于 2018-05-10 23:34:11 阅读数 1754 收藏  
展开

#### 【题目描述】

在某个字符串（长度不超过 100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用"\$"标注,不能匹配的右括号用"?"标注。

#### 【输入】

输入包括多组数据，每组数据一行，包含一个字符串，只包含左右括号和大小写字母，字符串长度不超过 100。

#### 【输出】

对每组输出数据，输出两行，第一行包含原始输入字符，第二行由"\$","?"和空格组成，"\$"和"?"表示与之对应的左括号和右括号不能匹配。

#### 【输入样例】

```
((ABCD(x)
)(rttyy())sss)(
```

#### 【输出样例】

```
((ABCD(x)
$$
)(rttyy())sss)(
?      ?$
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
```

```

#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
char a[101],b[101];
stack<int> S;
int main(){
    while(cin>>a)
    {
        int len=strlen(a);
        for(int i=0;i<len;i++)
        {
            if(a[i]=='(')
            {
                S.push(i);
                b[i]=' ';
            }
            else if(a[i]==')')
            {
                if(!S.empty())//栈不为空
                {
                    S.pop();
                    b[i]=' ';
                }
                else
                    b[i]='?';
            }
            else
                b[i]=' ';
        }
        while(!S.empty())//栈不为空
        {
            b[S.top()='$';
            S.pop();
        }
        b[len]='\0';
    }

```

```

        cout<<a<<endl;
        cout<<b<<endl;
    }
    return 0;
}

```

### 1.1.2.4.7 爬楼梯（信息学奥赛一本通-T1204）

#### 爬楼梯（信息学奥赛一本通-T1204）

原创 Alex\_McAvoy 发布于 2018-05-10 23:32:49 阅读数 2065 收藏  
展开

##### 【题目描述】

树老师爬楼梯，他可以每次走 1 级或者 2 级，输入楼梯的级数，求不同的走法数。

例如：楼梯一共有 3 级，他可以每次都走一级，或者第一次走一级，第二次走两级，也可以第一次走两级，第二次走一级，一共 3 种方法。

##### 【输入】

输入包含若干行，每行包含一个正整数  $N$ ，代表楼梯级数， $1 \leq N \leq 30$ 。

##### 【输出】

不同的走法数，每一行输入对应一行输出。

##### 【输入样例】

```

5
8
10

```

##### 【输出样例】

```

8
34
89

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
int calculate(int n)

```



```

{
    if(n==1)
        return 1;
    if(n==2)
        return 2;
    return calculate(n-1)+calculate(n-2);
}
int main()
{
    int n;
    while(scanf("%d",&n)!=EOF)
        cout<<calculate(n)<<endl;
    return 0;
}

```

### 1.1.2.4.8 汉诺塔问题（信息学奥赛一本通-T1205）

#### 汉诺塔问题（信息学奥赛一本通-T1205）

原创 Alex\_McAvoy 发布于 2018-05-10 23:31:34 阅读数 2753 收藏  
展开

##### 【题目描述】

约 19 世纪末，在欧洲的商店中出售一种智力玩具，在一块铜板上有三根杆，最左边的杆上自上而下、由小到大顺序串着由 64 个圆盘构成的塔。目的是将最左边杆上的盘全部移到中间的杆上，条件是一次只能移动一个盘，且不允许大盘放在小盘的上面。

这是一个著名的问题，几乎所有的教材上都有这个问题。由于条件是一次只能移动一个盘，且不允许大盘放在小盘上面，所以 64 个盘的移动次数是：  
18,446,744,073,709,551,615

这是一个天文数字，若每一微秒可能计算(并不输出)一次移动，那么也需要几乎一百万年。我们仅能找出问题的解决方法并解决较小 N 值时的汉诺塔，但很难用计算机解决 64 层的汉诺塔。

假定圆盘从小到大编号为 1, 2, ...

##### 【输入】

输入为一个整数(小于 20) 后面跟三个单字符字符串。整数为盘子的数目，后三个字符表示三个杆子的编号。

**【输出】**

输出每一步移动盘子的记录。一次移动一行。每次移动的记录为例如 **a->3->b** 的形式，即把编号为 **3** 的盘子从 **a** 杆移至 **b** 杆。

**【输入样例】**

2 a b c

**【输出样例】**

a->1->c  
a->2->b  
c->1->b

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
void Hanoi(int n,char a,char c,char b)
{
    if(n==1)
    {
        printf("%c->%d->%c\n",a,n,b);
        return ;
    }
    Hanoi(n-1,a,b,c);
    printf("%c->%d->%c\n",a,n,b);
    Hanoi(n-1,c,a,b);
}
int main()
{
    int n;
    char a,b,c;
    cin>>n>>a>>b>>c;
    Hanoi(n,a,c,b);
    return 0;
}
```

点赞

## 1.1.2.4.9 放苹果（信息学奥赛一本通-T1206）

### 放苹果（信息学奥赛一本通-T1206）

原创 Alex\_McAvoy 发布于 2018-05-10 23:30:00 阅读数 1887 收藏  
展开

#### 【题目描述】

把  $M$  个同样的苹果放在  $N$  个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？（用  $K$  表示）5，1，1 和 1，5，1 是同一种分法。

#### 【输入】

第一行是测试数据的数目  $t$  ( $0 \leq t \leq 20$ )。以下每行均包含二个整数  $M$  和  $N$ ，以空格分开。 $1 \leq M, N \leq 10$ 。

#### 【输出】

对输入的每组数据  $M$  和  $N$ ，用一行输出相应的  $K$ 。

#### 【输入样例】

```
1
7 3
```

#### 【输出样例】

```
8
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
int calculate(int m,int n)
{
    if(m==0||n==1)
        return 1;
    if(m<n)
        return calculate(m,m);
    return calculate(m,n-1)+calculate(m-n,n);
}
int main()
{
    int k;
```

```

int m,n;
cin>>k;
while(k--)
{
    cin>>m>>n;
    cout<<calculate(m,n)<<endl;
}
return 0;
}

```

点赞

### 1.1.2.4.10 求最大公约数问题（信息学奥赛一本通-T1207）

#### 求最大公约数问题（信息学奥赛一本通-T1207）

原创 Alex\_McAvoy 发布于 2018-05-10 23:28:54 阅读数 2104 收藏  
展开

##### 【题目描述】

给定两个正整数，求它们的最大公约数。

##### 【输入】

输入一行，包含两个正整数(<1,000,000,000)。

##### 【输出】

输出一个正整数，即这两个正整数的最大公约数。

##### 【输入样例】

6 9

##### 【输出样例】

3

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
int calculate(int a,int b)

```

```

{
    if(b==0)
        return a;
    return calculate(b,a%b);
}
int main()
{
    int a,b;
    cin>>a>>b;
    if(a>b)
        cout<<calculate(a,b)<<endl;
    else
        cout<<calculate(b,a)<<endl;
    return 0;
}

```

点赞

### 1.1.2.4.11 2 的幂次方表示(信息学奥赛一本通-T1208)

#### 2 的幂次方表示 (信息学奥赛一本通-T1208)

原创 Alex\_McAvoy 发布于 2018-05-10 23:44:38 阅读数 1982 收藏  
展开

##### 【题目描述】

任何一个正整数都可以用 2 的幂次方表示。例如： $137=2^7+2^3+2^0$

同时约定方次用括号来表示，即  $a^b$  可表示为  $a(b)$ 。

由此可知，137 可表示为： $2(7)+2(3)+2(0)$

进一步： $7=2^2+2+2^0$ （21 用 2 表示）

$3=2+2^0$

所以最后 137 可表示为： $2(2(2)+2+2(0))+2(2+2(0))+2(0)$

又如： $1315=2^{10}+2^8+2^5+2+1$

所以 1315 最后可表示为： $2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)$

##### 【输入】

一个正整数  $n$  ( $n \leq 20000$ )。

**【输出】**

一行，符合约定的  $n$  的 0, 2 表示（在表示中不能有空格）。

**【输入样例】**

137

**【输出样例】**

2(2(2)+2+2(0))+2(2+2(0))+2(0)

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
void calculate(int n,int step)
{
    if(n==0)
        return;
    calculate(n/2,step+1);
    if(n%2)
    {
        if(n/2)
            cout<<"+";
        if(step==1)
            cout<<"2";
        else
        {
            cout<<"2(";
            if(step==0)
                cout<<"0";
            else
                calculate(step,0);
            cout<<")";
        }
    }
}
int main()
{
    int n;
    cin>>n;
    calculate(n,0);
    cout<<endl;
    return 0;
}
```

}

点赞

## 1.1.2.4.12 分数求和（信息学奥赛一本通-T1209）

### 分数求和（信息学奥赛一本通-T1209）

原创 Alex\_McAvoy 发布于 2018-05-10 23:41:55 阅读数 1695 收藏  
展开

#### 【题目描述】

输入  $n$  个分数并对他们求和，并用最简形式表示。所谓最简形式是指：分子分母的最大公约数为 1/1；若最终结果的分母为 1，则直接用整数表示。

如：5/6、10/3 均是最简形式，而 3/6 需要化简为 1/2, 3/1 需要化简为 3。

分子和分母均不为 0，也不为负数。

#### 【输入】

第一行是一个整数  $n$ ，表示分数个数， $1 \leq n \leq 10$ ；

接下来  $n$  行，每行一个分数，用 "p/q" 的形式表示，不含空格， $p, q$  均不超过 10。

#### 【输出】

输出只有一行，即最终结果的最简形式。若为分数，用 "p/q" 的形式表示。

#### 【输入样例】

```
2
1/2
1/3
```

#### 【输出样例】

```
5/6
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
```

```

int a[20],b[20];
int gcd(int a,int b)
{
    if(b==0)
        return a;
    return gcd(b,a%b);
}
int main()
{
    int n;
    int cnt=0;
    int numerator=0,denominator=1;
    int divisor;
    char s[20];

    cin>>n;
    while(n-->0)
    {
        scanf("%d/%d",&a[cnt],&b[cnt]);
        cnt++;
    }
    for(int i=0;i<cnt;i++)
        denominator*=b[i];
    for(int i=0;i<cnt;i++)
        numerator=numerator+denominator*a[i]/b[i];

    divisor=gcd(denominator,numerator);
    denominator/=divisor;
    numerator/=divisor;

    if(denominator==1)
        cout<<numerator<<endl;
    else
        cout<<numerator<<"/"<<denominator<<endl;
    return 0;
}

```

### 1.1.2.4.13 因子分解（信息学奥赛一本通-T1210）

#### 因子分解（信息学奥赛一本通-T1210）

原创 Alex\_McAvoy 发布于 2018-05-10 23:38:52 阅读数 1328 收藏  
展开



**【题目描述】**

输入一个数，输出其素因子分解表达式。

**【输入】**

输入一个整数  $n$  ( $2 \leq n < 100$ )。

**【输出】**

输出该整数的因子分解表达式。

表达式中各个素数从小到大排列。

如果该整数可以分解出因子  $a$  的  $b$  次方，当  $b$  大于 1 时，写做  $a^b$ ；当  $b$  等于 1 时，则直接写成  $a$ 。

**【输入样例】**

60

**【输出样例】**

$2^2 * 3 * 5$

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 10001
using namespace std;
int n;
int a[N],b[N];
void calculate(int x,int y)
{
    if(x==0||y>x)
        return;
    while(x%y==0)
    {
        x/=y;
        a[y]++;
    }
    calculate(x,y+1);
}
int main()
{
    bool flag=false;

    cin>>n;
    calculate(n,2);
```

```

for(int i=2;i<=n;i++)
{
    if(flag&&a[i])
        cout<<"*";
    if(a[i])
        flag=true;
    if(a[i]==1)
        cout<<i;
    else if(a[i]>1)
        cout<<i<<"^"<<a[i];
}
cout<<endl;
return 0;
}

```

点赞 1

### 1.1.2.4.14 断元素是否存在（信息学奥赛一本通-T1211）

#### 断元素是否存在（信息学奥赛一本通-T1211）

原创 Alex\_McAvoy 发布于 2018-05-10 23:37:37 阅读数 2085 收藏  
展开

##### 【题目描述】

有一个集合  $M$  是这样生成的：(1) 已知  $k$  是集合  $M$  的元素；(2) 如果  $y$  是  $M$  的元素，那么， $2y+1$  和  $3y+1$  都是  $M$  的元素；(3) 除了上述二种情况外，没有别的数能够成为  $M$  的一个元素。

问题：任意给定  $k$  和  $x$ ，请判断  $x$  是否是  $M$  的元素。这里的  $k$  是无符号整数， $x$  不大于 100000，如果是，则输出 YES，否则，输出 NO。

##### 【输入】

输入整数  $k$  和  $x$ ，逗号间隔。

##### 【输出】

如果是，则输出 YES，否则，输出 NO。

##### 【输入样例】

0,22

**【输出样例】**

YES

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1000010
using namespace std;
int k;
int judge(int x)
{
    if(x==k)
        return 1;
    if((x-1)%3==0&&(x-1)%2==0)
        return (judge((x-1)/3)||judge((x-1)/2));
    if((x-1)%3==0)
        return judge((x-1)/3);
    if((x-1)%2==0)
        return judge((x-1)/2);
    return 0;
}
int main()
{
    int x;
    scanf("%d,%d",&k,&x);
    if(judge(x))
        cout<<"YES"<<endl;
    else
        cout<<"NO"<<endl;
    return 0;
}

```

## 1.1.2.4.15 集合的划分（信息学奥赛一本通-T1315）

### 集合的划分（信息学奥赛一本通-T1315）

原创 Alex\_McAvoy 发布于 2018-05-09 23:40:38 阅读数 1906 收藏  
展开

**【题目描述】**

设  $S$  是一个具有  $n$  个元素的集合,  $S=\{a_1, a_2, \dots, a_n\}$ , 现将  $S$  划分成  $k$  个满足下列条件的子集合  $S_1, S_2, \dots, S_k$  且满足:

1.  $S_i \neq \emptyset$
2.  $S_i \cap S_j = \emptyset$  ( $1 \leq i, j \leq k, i \neq j$ )
3.  $S_1 \cup S_2 \cup S_3 \cup \dots \cup S_k = S$

则称  $S_1, S_2, \dots, S_k$  是集合  $S$  的一个划分。

它相当于把  $S$  集合中的  $n$  个元素  $a_1, a_2, \dots, a_n$  放入  $k$  个 ( $0 < k \leq n \leq 30$ ) 无标号的盒子中, 使得没有一个盒子为空。请你确定  $n$  个元素  $a_1, a_2, \dots, a_n$  放入  $k$  个无标号盒子中去的划分数  $S(n, k)$ 。

**【输入】**

给出  $n$  和  $k$ 。

**【输出】**

$n$  个元素  $a_1, a_2, \dots, a_n$  放入  $k$  个无标号盒子中去的划分数  $S(n, k)$ 。

**【输入样例】**

10 6

**【输出样例】**

22827

思路: 计算  $C(n, k)$ , 直接递归即可

**【源程序】**

```
#include<iostream>
using namespace std;
long long calculate(long long n, long long k)
{
    if(n < k || k == 0)
        return 0;
    if(n == k || k == 1)
        return 1;
    return calculate(n-1, k-1) + k * calculate(n-1, k);
}
int main()
{
    long long n, k;
    cin >> n >> k;
    cout << calculate(n, k) << endl;
    return 0;
}
```

## 1.1.2.4.16 数的计数（信息学奥赛一本通-T1316）

### 数的计数（信息学奥赛一本通-T1316）

原创 Alex\_McAvoy 发布于 2018-05-09 23:43:39 阅读数 1173 收藏  
展开

#### 【题目描述】

我们要求找出具有下列性质数的个数（包括输入的自然数  $n$ ）。先输入一个自然数  $n$  ( $n \leq 1000$ )，然后对此自然数按照如下方法进行处理：

不作任何处理；

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；  
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。

#### 【输入】

自然数  $n$  ( $n \leq 1000$ )。

#### 【输出】

满足条件的数。

#### 【输入样例】

6

#### 【输出样例】

6

提示：满足条件的数为 6、16、26、126、36、136

#### 【源程序】

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    int a[1100];

    cin>>n;
    a[1]=1;
    a[2]=2;
    for(int i=3;i<=n;i++)
    {
        if(i%2==0)
            a[i]=a[i-1]+a[i/2];
```

```

else
    a[i]=a[i-1];
}
cout<<a[n]<<endl;
return 0;
}

```

## 1.1.2.5 第五章 搜索与回溯算法（DFS）

### 1.1.2.5.1 组合的输出（信息学奥赛一本通-T1317）

#### 组合的输出（信息学奥赛一本通-T1317）

原创 Alex\_McAvoy 发布于 2018-05-11 15:43:37 阅读数 2016 收藏  
展开

##### 【题目描述】

排列与组合是常用的数学方法，其中组合就是从  $n$  个元素中抽出  $r$  个元素(不分顺序且  $r \leq n$ )，我们可以简单地将  $n$  个元素理解为自然数  $1, 2, \dots, n$ ，从中任取  $r$  个数。

现要求你用递归的方法输出所有组合。

例如  $n=5, r=3$ ，所有组合为：1 2 3   1 2 4   1 2 5   1 3 4   1 3 5   1 4 5   2 3 4   2 3 5   2 4 5   3 4 5

##### 【输入】

一行两个自然数  $n, r(1 < n < 21, 1 \leq r \leq n)$ 。

##### 【输出】

所有的组合，每一个组合占一行且其中的元素按由小到大的顺序排列，每个元素占三个字符的位置，所有的组合也按字典顺序。

##### 【输入样例】

5 3

##### 【输出样例】

```

1 2 3
1 2 4
1 2 5
1 3 4
1 3 5

```

```

1 4 5
2 3 4
2 3 5
2 4 5
3 4 5

```

### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 30
using namespace std;
int n,r;
int a[N];
int vis[N];
void dfs(int step)
{
    int i;

    if(step==r+1)
    {
        for(i=1;i<=r;i++)
            cout<<" "<<a[i];
        cout<<endl;
        return;
    }

    for(i=a[step-1];i<=n;i++)
    {
        if(vis[i]==0)
        {
            a[step]=i;
            vis[i]=1;

            dfs(step+1);

            vis[i]=0;
        }
    }
}
int main()
{
    cin>>n>>r;
    a[0]=1;

```

```

dfs(1);
return 0;
}

```

### 1.1.2.5.2 自然数的拆分（信息学奥赛一本通-T1318）

#### 自然数的拆分（信息学奥赛一本通-T1318）

原创 Alex\_McAvoy 发布于 2018-05-11 15:45:09 阅读数 1250 收藏  
展开

##### 【题目描述】

任何一个大于 1 的自然数  $n$ ，总可以拆分成若干个小于  $n$  的自然数之和。

当  $n=7$  共 14 种拆分方法：

$7=1+1+1+1+1+1+1$

$7=1+1+1+1+1+2$

$7=1+1+1+1+3$

$7=1+1+1+2+2$

$7=1+1+1+4$

$7=1+1+2+3$

$7=1+1+5$

$7=1+2+2+2$

$7=1+2+4$

$7=1+3+3$

$7=1+6$

$7=2+2+3$

$7=2+5$

$7=3+4$

##### 【输入】

输入  $n$ 。

##### 【输出】

按字典序输出具体的方案。

##### 【输入样例】

7

##### 【输出样例】

$7=1+1+1+1+1+1+1$

$7=1+1+1+1+1+2$



7=1+1+1+1+3

7=1+1+1+2+2

7=1+1+1+4

7=1+1+2+3

7=1+1+5

7=1+2+2+2

7=1+2+4

7=1+3+3

7=1+6

7=2+2+3

7=2+5

7=3+4

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 3000
using namespace std;
int n,r;
int a[N];
void print(int step)
{
    cout<<n<<"=";
    for(int i=1;i<=step-1;i++)
        cout<<a[i]<<" ";
    cout<<a[step]<<endl;
}
void dfs(int sum,int step)
{
    for(int i=a[step-1];i<=sum;i++)
    {
        if(i<n)
        {
            a[step]=i;
            sum-=i;
            if(sum==0)
                print(step);
            else
                dfs(sum,step+1);
            sum+=i;
        }
    }
}
```

```
int main()
{
    cin>>n;
    a[0]=1;
    dfs(n,1);
    return 0;
}
```

版权声明：本文为 CSDN 博主「Alex\_McAvoy」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011815404/article/details/80282523> to insert a line break. Press Enter to add a paragraph return.)

### 1.1.2.5.3 LETTERS （信息学奥赛一本通-T1212）

#### LETTERS （信息学奥赛一本通-T1212）

原创 Alex\_McAvoy 发布于 2018-05-11 15:46:03 阅读数 3124 收藏  
展开

##### 【题目描述】

给出一个  $roe \times col$  的大写字母矩阵，一开始的位置为左上角，你可以向上下左右四个方向移动，并且不能移向曾经经过的字母。问最多可以经过几个字母。

##### 【输入】

第一行，输入字母矩阵行数  $R$  和列数  $S$ ， $1 \leq R, S \leq 20$ 。

接着输出  $R$  行  $S$  列字母矩阵。

##### 【输出】

最多能走过的不同字母的个数。

##### 【输入样例】

```
3 6
HFDFFB
AJHGDH
DGAGEH
```

##### 【输出样例】

```
6
```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 30
using namespace std;
int r,s;
char a[N][N];
int vis[N][N];
int num[26];
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0}};
int maxx=0;
void dfs(int x,int y,int step)
{
    if(maxx<step)
        maxx=step;
    for(int i=0;i<4;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=0&&nx<r&&ny>=0&&ny<s&&vis[nx][ny]==0&&num[a[nx][ny]-'A']==0)
        {
            vis[nx][ny]=1;
            num[a[nx][ny]-'A']=1;
            dfs(nx,ny,step+1);
            vis[nx][ny]=0;
            num[a[nx][ny]-'A']=0;
        }
    }
}
int main()
{
    cin>>r>>s;
    for(int i=0;i<r;i++)
        for(int j=0;j<s;j++)
            cin>>a[i][j];
    num[a[0][0]-'A']=1;
    vis[0][0]=1;
    dfs(0,0,1);
    cout<<maxx<<endl;
    return 0;
}

```

## 1.1.2.5.4 八皇后问题 （信息学奥赛一本通-T1213）

### 八皇后问题 （信息学奥赛一本通-T1213）

原创 Alex\_McAvoy 发布于 2018-05-11 15:47:42 阅读数 3361 收藏  
展开

#### 【题目描述】

在国际象棋棋盘上放置八个皇后，要求每两个皇后之间不能直接吃掉对方。

#### 【输入】

(无)

#### 【输出】

按给定顺序和格式输出所有八皇后问题的解（见样例）。

#### 【输入样例】

(无)

#### 【输出样例】

No. 1

```
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
```

No. 2

```
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0
```

...以下省略

#### 【源程序】

```
#include<iostream>
```

```

#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 100
using namespace std;
int a[N][N],b[N];
int vis[N][N];
int tot;
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0}};
void dfs(int step)
{
    if(step==8+1)
    {
        tot++;
        for(int i=1;i<=8;i++)
            a[tot][i]=b[i];
        return;
    }
    for(int i=1;i<=8;i++)
    {
        if(vis[0][i]==0&&vis[1][step+i]==0&&vis[2][step-i+8]==0)
        {
            vis[0][i]=1;
            vis[1][i+step]=1;
            vis[2][step-i+8]=1;
            b[step]=i;
            dfs(step+1);
            vis[0][i]=0;
            vis[1][i+step]=0;
            vis[2][step-i+8]=0;
        }
    }
}
int main()
{
    dfs(1);
    for(int t=1;t<=tot;t++)
    {
        printf("No. %d\n",t);
        for(int i=1;i<=8;i++)
        {
            for(int j=1;j<=8;j++)
            {
                if(a[t][j]==i)

```

```

        cout<<"1 ";
    else
        cout<<"0 ";
    }
    cout<<endl;
}
}
return 0;
}

```

### 1.1.2.5.5 八皇后（信息学奥赛一本通-T1214）

#### 八皇后（信息学奥赛一本通-T1214）

原创 Alex\_McAvoy 发布于 2018-05-11 15:49:15 阅读数 2050 收藏  
展开

##### 【题目描述】

会下国际象棋的人都很清楚：皇后可以在横、竖、斜线上不限步数地吃掉其他棋子。如何将 8 个皇后放在棋盘上（有  $8 \times 8$  个方格），使它们谁也不能被吃掉！这就是著名的八皇后问题。

对于某个满足要求的 8 皇后的摆放方法，定义一个皇后串  $a$  与之对应，即  $a=b_1b_2\dots b_8$ ，其中  $b_i$  为相应摆法中第  $i$  行皇后所处的列数。已经知道 8 皇后问题一共有 92 组解（即 92 个不同的皇后串）。

给出一个数  $b$ ，要求输出第  $b$  个串。串的比较是这样的：皇后串  $x$  置于皇后串  $y$  之前，当且仅当将  $x$  视为整数时比  $y$  小。

##### 【输入】

第 1 行是测试数据的组数  $n$ ，后面跟着  $n$  行输入。每组测试数据占 1 行，包括一个正整数  $b(1 \leq b \leq 92)$ 。

##### 【输出】

输出有  $n$  行，每行输出对应一个输入。输出应是一个正整数，是对应于  $b$  的皇后串。

##### 【输入样例】

```

2
1
92

```

## 【输出样例】

15863724

84136275

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 100
using namespace std;
int a[N][N],b[N];
int vis[N][N];
int tot;
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0}};
void dfs(int step)
{
    if(step==8+1)
    {
        tot++;
        for(int i=1;i<=8;i++)
            a[tot][i]=b[i];
        return;
    }
    for(int i=1;i<=8;i++)
    {
        if(vis[0][i]==0&&vis[1][step+i]==0&&vis[2][step-i+8]==0)
        {
            vis[0][i]=1;
            vis[1][i+step]=1;
            vis[2][step-i+8]=1;
            b[step]=i;
            dfs(step+1);
            vis[0][i]=0;
            vis[1][i+step]=0;
            vis[2][step-i+8]=0;
        }
    }
}
int main()
{
    int n;
    cin>>n;
    dfs(1);
    while(n--)

```

```

{
    int i;
    cin>>i;
    for(int j=1;j<=8;j++)
        cout<<a[i][j];
    cout<<endl;
}
return 0;
}

```

## 1.1.2.5.6 迷宫（信息学奥赛一本通-T1215）

### 迷宫（信息学奥赛一本通-T1215）

原创 Alex\_McAvoy 发布于 2018-05-11 15:50:47 阅读数 1787 收藏  
展开

#### 【题目描述】

一天 **Extense** 在森林里探险的时候不小心走入了一个迷宫，迷宫可以看成是由  $n * n$  的格点组成，每个格点只有 2 种状态，.和#，前者表示可以通行后者表示不能通行。同时当 **Extense** 处在某个格点时，他只能移动到东南西北(或者说上下左右)四个方向之一的相邻格点上，**Extense** 想要从点 A 走到点 B，问在不走出迷宫的情况下能不能办到。如果起点或者终点有一个不能通行(为#)，则看成无法办到。

#### 【输入】

第 1 行是测试数据的组数  $k$ ，后面跟着  $k$  组输入。每组测试数据的第 1 行是一个正整数  $n$  ( $1 \leq n \leq 100$ )，表示迷宫的规模是  $n * n$  的。接下来是一个  $n * n$  的矩阵，矩阵中的元素为.或者#。再接下来一行是 4 个整数  $ha, la, hb, lb$ ，描述 A 处在第  $ha$  行，第  $la$  列，B 处在第  $hb$  行，第  $lb$  列。注意到  $ha, la, hb, lb$  全部是从 0 开始计数的。

#### 【输出】

$k$  行，每行输出对应一个输入。能办到则输出“YES”，否则输出“NO”。

#### 【输入样例】

```

2
3
.##
..#
#..
0 0 2 2
5
.....
####.#

```



```
..#..
###..
...#.
0 0 4 0
```

## 【输出样例】

```
YES
NO
```

## 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 101
using namespace std;
char ch;
int n,xa,ya,xb,yb;
bool flag;
int vis[N][N];
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0};
void dfs(int x,int y)
{
    for(int i=0;i<4;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=0&&nx<n&&ny>=0&&ny<n&&vis[nx][ny]==0)
        {
            vis[nx][ny]=1;
            if(nx==xb&&ny==yb)
            {
                cout<<"YES"<<endl;
                flag=true;
                break;
            }
            else
                dfs(nx,ny);
        }
    }
}
int main()
{
    int k;
    cin>>k;
```

```

while(k--)
{
    memset(vis,0,sizeof(vis));
    flag=false;
    cin>>n;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
        {
            cin>>ch;
            if(ch=='#')
                vis[i][j]=1;
        }
    cin>>xa>>ya;
    cin>>xb>>yb;
    if(vis[xa][ya]||vis[xb][yb])
    {
        cout<<"NO"<<endl;
        continue;
    }
    else
        dfs(xa,ya);
    if(!flag)
        cout<<"NO"<<endl;
}
return 0;
}

```

### 1.1.2.5.7 红与黑（信息学奥赛一本通-T1216）

#### 红与黑（信息学奥赛一本通-T1216）

原创 Alex\_McAvoy 发布于 2018-05-11 15:52:09 阅读数 2587 收藏  
展开

##### 【题目描述】

有一间长方形的房子，地上铺了红色、黑色两种颜色的正方形瓷砖。你站在其中一块黑色的瓷砖上，只能向相邻的黑色瓷砖移动。请写一个程序，计算你总共能够到达多少块黑色的瓷砖。

**【输入】**

包括多个数据集合。每个数据集合的第一行是两个整数  $W$  和  $H$ ，分别表示  $x$  方向和  $y$  方向瓷砖的数量。 $W$  和  $H$  都不超过 20。在接下来的  $H$  行中，每行包括  $W$  个字符。每个字符表示一块瓷砖的颜色，规则如下：

- 1) ‘.’：黑色的瓷砖；
- 2) ‘#’：白色的瓷砖；
- 3) ‘@’：黑色的瓷砖，并且你站在这块瓷砖上。该字符在每个数据集合中唯一出现一次。

当在一行中读入的是两个零时，表示输入结束。

**【输出】**

对每个数据集合，分别输出一行，显示你从初始位置出发能到达的瓷砖数(记数时包括初始位置的瓷砖)。

**【输入样例】**

```
6 9
...#.
....#
.....
.....
.....
.....
.....
.....
#@...#
.#..#
0 0
```

**【输出样例】**

```
45
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 1001
using namespace std;
int m,n;
char ch;
int maps[N][N];
int vis[N][N];
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0}};
```

```

int cnt;
void dfs(int x,int y)
{
    for(int i=0;i<4;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=1&&ny>=1&&nx<=n&&ny<=m&&vis[nx][ny]==0&&maps[nx][ny]==1)
        {
            vis[nx][ny]=1;
            cnt++;
            dfs(nx,ny);
        }
    }
}
int main()
{
    while(scanf("%d%d",&m,&n)!=EOF&&m&&n)
    {
        int x,y;
        cnt=1;
        memset(vis,0,sizeof(vis));
        memset(maps,0,sizeof(maps));

        for(int i=1;i<=n;i++)
            for(int j=1;j<=m;j++)
            {
                cin>>ch;
                if(ch=='@')
                {
                    x=i;
                    y=j;
                    maps[i][j]=1;
                }
                if(ch=='.')
                    maps[i][j]=1;
                if(ch=='#')
                    maps[i][j]=0;
            }
        vis[x][y]=1;
        dfs(x,y);
        cout<<cnt<<endl;
    }
    return 0;
}

```

}

点赞 2

## 1.1.2.5.8 棋盘问题（信息学奥赛一本通-T1217）

### 棋盘问题（信息学奥赛一本通-T1217）

原创 Alex\_McAvoy 发布于 2018-05-11 15:53:24 阅读数 2038 收藏  
展开

#### 【题目描述】

在一个给定形状的棋盘（形状可能是不规则的）上面摆放棋子，棋子没有区别。要求摆放时任意的两个棋子不能放在棋盘中的同一行或者同一列，请编程求解对于给定形状和大小的棋盘，摆放  $k$  个棋子的所有可行的摆放方案  $C$ 。

#### 【输入】

输入含有多组测试数据。

每组数据的第一行是两个正整数  $n, k$ ，用一个空格隔开，表示了将在一个  $n \times n$  的矩阵内描述棋盘，以及摆放棋子的数目。（ $n \leq 8, k \leq n$ ）

当为  $0\ 0$  时表示输入结束。

随后的  $n$  行描述了棋盘的形状：每行有  $n$  个字符，其中  $\#$  表示棋盘区域， $.$  表示空白区域（数据保证不出现多余的空白行或者空白列）。

#### 【输出】

对于每一组数据，给出一行输出，输出摆放的方案数目  $C$ （数据保证  $C < 2^{31}$ ）。

#### 【输入样例】

```
2 1
#
.#
4 4
...#
..#.
.#..
#...
-1 -1
```

## 【输出样例】

2

1

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 10
using namespace std;
int n,k;
char maps[N][N];
int vis[N];
int dir[4][2]={0,1},{0,-1},{1,0},{-1,0}};
int cnt;
void dfs(int x,int y)
{
    if(y==k)
    {
        cnt++;
        return;
    }
    for(int i=x;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(maps[i][j]=='#'&&vis[j])
            {
                vis[j]=0;
                dfs(i+1,y+1);
                vis[j]=1;
            }
    return;
}
int main()
{
    while(scanf("%d%d",&n,&k)!=EOF)
    {
        if(n==-1&&k==-1)
            break;

        memset(vis,1,sizeof(vis));

        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                cin>>maps[i][j];
    }
}

```

```

    cnt=0;
    dfs(1,0);
    cout<<cnt<<endl;
}
return 0;
}

```

点赞 1

版权声明：本文为 CSDN 博主「Alex\_McAvoy」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011815404/article/details/80282634> to insert a line break. Press Enter to add a paragraph return.)

## 1.1.2.5.9 取石子游戏（信息学奥赛一本通-T1218）

### 取石子游戏（信息学奥赛一本通-T1218）

原创 Alex\_McAvoy 发布于 2018-05-11 15:55:51 阅读数 1668 收藏  
展开

#### 【题目描述】

有两堆石子,两个人轮流去取。每次取的时候,只能从较多的那堆石子里取,并且取的数目必须是较少的那堆石子数目的整数倍,最后谁能够把一堆石子取空谁就算赢。

比如初始的时候两堆石子的数目是 25 和 7。

25 7 --> 11 7 --> 4 7 --> 4 3 --> 1 3 --> 1 0

选手 1 取 选手 2 取 选手 1 取 选手 2 取 选手 1 取

最后选手 1（先取的）获胜，在取的过程中选手 2 都只有唯一的一种取法。

给定初始时石子的数目，如果两个人都采取最优策略，请问先手能否获胜。

#### 【输入】

输入包含多数数据。每组数据一行，包含两个正整数 **a** 和 **b**，表示初始时石子的数目。

输入以两个 0 表示结束。

#### 【输出】

如果先手胜，输出"win"，否则输出"lose"

#### 【输入样例】

34 12

15 24

0 0

**【输出样例】**

win

lose

思路：使用欧几里得辗转相除法即可

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 101
using namespace std;
int main()
{
    int m,n;
    int temp;
    int flag;
    while(scanf("%d%d",&m,&n)!=EOF&&m&&n)
    {
        flag=1;
        if(m<n)
        {
            temp=m;
            m=n;
            n=temp;
        }
        while(m/n==1)
        {
            temp=m;
            m=n;
            n=temp%n;
            flag=-flag;
        }
        if(flag==1)
            cout<<"win"<<endl;
        else
            cout<<"lose"<<endl;
    }
    return 0;
}
```



## 1.1.2.5.10 马走日（信息学奥赛一本通-T1219）

### 马走日（信息学奥赛一本通-T1219）

原创 Alex\_McAvoy 发布于 2018-05-11 15:57:37 阅读数 2181 收藏  
展开

#### 【题目描述】

马在中国象棋以日字形规则移动。

请编写一段程序，给定  $n \times m$  大小的棋盘，以及马的初始位置  $(x, y)$ ，要求不能重复经过棋盘上的同一个点，计算马可以有多少途径遍历棋盘上的所有点。

#### 【输入】

第一行为整数  $T(T < 10)$ ，表示测试数据组数。

每一组测试数据包含一行，为四个整数，分别为棋盘的大小以及初始位置坐标  $n, m, x, y$ 。  
( $0 \leq x \leq n-1, 0 \leq y \leq m-1, m < 10, n < 10$ )。

#### 【输出】

每组测试数据包含一行，为一个整数，表示马能遍历棋盘的途径总数，0 为无法遍历一次。

#### 【输入样例】

```
1
5 4 0 0
```

#### 【输出样例】

```
32
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 10
using namespace std;
int n,m;
int x0,y0;
char maps[N][N];
int vis[N][N];
int dir[8][2]={{-2,1},{-2,-1},{-1,2},{-1,-2},{2,1},{2,-1},{1,2},{1,-2}};
int cnt;
void dfs(int x,int y,int step)
```

```

{
    if(step==n*m)
    {
        cnt++;
        return;
    }
    for(int i=0;i<8;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=0&&ny>=0&&nx<n&&ny<m&&vis[nx][ny]==0)
        {
            vis[nx][ny]=1;
            dfs(nx,ny,step+1);
            vis[nx][ny]=0;
        }
    }
}
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        cnt=0;
        memset(vis,0,sizeof(vis));

        cin>>n>>m;
        cin>>x0>>y0;

        vis[x0][y0]=1;
        dfs(x0,y0,1);
        cout<<cnt<<endl;
    }
    return 0;
}

```

### 1.1.2.5.11 单词接龙（信息学奥赛一本通-T1220）

#### 单词接龙（信息学奥赛一本通-T1220）

原创 Alex\_McAvoy 发布于 2018-05-12 16:02:54 阅读数 1707 收藏  
展开

### 【题目描述】

单词接龙是一个与我们经常玩的成语接龙相类似的游戏，现在我们已知一组单词，且给定一个开头的字母，要求出以这个字母开头的最长的“龙”（每个单词都最多在“龙”中出现两次），在两个单词相连时，其重合部分合为一部分，例如 **beast** 和 **astonish**，如果接成一条龙则变为 **beastonish**，另外相邻的两部分不能存在包含关系，例如 **at** 和 **atide** 间不能相连。

### 【输入】

输入的第一行为一个单独的整数  $n(n \leq 20)$  表示单词数，以下  $n$  行每行有一个单词（只含有大写或小写字母，长度不超过 20），输入的最后一行为一个单个字符，表示“龙”开头的字母。你可以假定以此字母开头的“龙”一定存在。

### 【输出】

只需输出以此字母开头的最长的“龙”的长度。

### 【输入样例】

```
5
at
touch
cheat
choose
tact
a
```

### 【输出样例】

```
23
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<string>
using namespace std;
int n;
string str[30];
int len_str,sum=0;
int times[30]={0};

void dfs(int x)
{
    int i,j;
    int p,q;
    int num;
```

```

int work;

for(i=1;i<str[x].length();i++)
{
    num=0;
    for(j=1;j<=n;j++)
    {
        if(times[j]<2)
        {
            if(str[x][i]==str[j][0])
            {
                p=i,q=0,work=0;
                while(p<=str[x].length()-1)
                {
                    if(str[x][p]!=str[j][q])
                    {
                        num++;
                        work++;
                        break;
                    }
                    p++;
                    q++;
                }
                if(!work&&q!=str[j].length())
                {
                    len_str+=str[j].length()-q;
                    times[j]++;

                    dfs(j);

                    len_str-=str[j].length()-q;
                    times[j]--;
                }
            }
            else num++;
        }
        else num++;
    }
}

if(num==n&&i==str[x].length()-1)
{
    if(sum<len_str) sum=len_str;
    return;
}

```

```

    }
}
int main()
{
    int i;

    cin>>n;
    for(i=1;i<=n+1;i++)    cin>>str[i];

    for(i=1;i<=n;i++)
        if(str[i][0]==str[n+1][0])
        {
            times[i]++;
            len_str=str[i].length();
            dfs(i);
        }

    cout<<sum<<endl;
    return 0;
}

```

### 1.1.2.5.12 分成互质组 （信息学奥赛一本通-T1221）

#### 分成互质组 （信息学奥赛一本通-T1221）

原创 Alex\_McAvoy 发布于 2018-05-12 16:03:21 阅读数 1754 收藏  
展开

##### 【题目描述】

给定  $n$  个正整数，将它们分组，使得每组中任意两个数互质。至少要分成多少个组？

##### 【输入】

第一行是一个正整数  $n$ 。  $1 \leq n \leq 10$ 。

第二行是  $n$  个不大于 10000 的正整数。

##### 【输出】

一个正整数，即最少需要的组数。

##### 【输入样例】

```

6
14 20 33 117 143 175

```

## 【输出样例】

3

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define N 20
using namespace std;
int n;
int a[N];
int cnt=99999;
long long vis[N];
long long gcd(long long a,long long b)
{
    if(b==0)
        return a;
    return
        gcd(b,a%b);
}
void dfs(int k,int step)
{
    if(step==n+1)
    {
        if(k<cnt)
            cnt=k;
        return;
    }
    for(int i=1;i<=k;i++)
        if(gcd(vis[i],a[step])==1)
        {
            vis[i]*=a[step];
            dfs(k,step+1);
            vis[i]/=a[step];
        }
    vis[k+1]*=a[step];
    dfs(k+1,step+1);
    vis[k+1]/=a[step];
}
int main()
{
    int temp;
    cin>>n;

```

```

for(int i=1;i<=n;i++)
{
    cin>>a[i];
    vis[i]=1;
}
sort(a+1,a+1+n);
dfs(1,1);
cout<<cnt<<endl;
return 0;
}

```

### 1.1.2.5.13 放苹果（信息学奥赛一本通-T1222）

#### 放苹果（信息学奥赛一本通-T1222）

原创 Alex\_McAvoy 发布于 2018-05-12 16:03:15 阅读数 1518 收藏  
展开

##### 【题目描述】

把  $M$  个同样的苹果放在  $N$  个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？（用  $K$  表示）5，1，1 和 1，5，1 是同一种分法。

##### 【输入】

第一行是测试数据的数目  $t$  ( $0 \leq t \leq 20$ )。以下每行均包含二个整数  $M$  和  $N$ ，以空格分开。 $1 \leq M, N \leq 10$ 。

##### 【输出】

对输入的每组数据  $M$  和  $N$ ，用一行输出相应的  $K$ 。

##### 【输入样例】

```

1
7 3

```

##### 【输出样例】

```

8

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#define N 10
using namespace std;

```

```

int max[15];
int n,m;
void setmax()//每个盘最多放多少个
{
    for(int i=0;i<n;i++)
        max[i+1]=m/(n-i);
}
int dfs(int sum,int p,int v)
{
    int ans=0;
    if(v==n&& m-sum>=p)
        return 1;
    if(v==n)
        return 0;
    for(int s=p;s<=max[v];s++)
        ans+=dfs(sum+s,s,v+1);
    return ans;
}
int main()
{
    int t,ans[15][15]={0};
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d",&m,&n);
        setmax();
        if(ans[n][m]==0)
            ans[n][m]=dfs(0,0,1);
        printf("%d\n",ans[n][m]);
    }
}

```

## 1.1.2.6 第六章 贪心算法

### 1.1.2.6.1 排队接水（信息学奥赛一本通-T1319）

排队接水（信息学奥赛一本通-T1319）



原创 Alex\_McAvoy 发布于 2018-05-12 19:12:21 阅读数 2469 收藏  
展开

### 【题目描述】

有  $n$  个人在一个水龙头前排队接水，假如每个人接水的时间为  $T_i$ ，请编程找出这  $n$  个人排队的一种顺序，使得  $n$  个人的平均等待时间最小。

### 【输入】

共两行，第一行为  $n(1 \leq n \leq 1000)$ ；第二行分别表示第 1 个人到第  $n$  个人每人的接水时间  $T_1, T_2, \dots, T_n$ ，每个数据之间有 1 个空格。

### 【输出】

有两行，第一行为一种排队顺序，即 1 到  $n$  的一种排列；第二行为这种排列方案下的平均等待时间(输出结果精确到小数点后两位)。

### 【输入样例】

```
10
56 12 1 99 1000 234 33 55 99 812
```

### 【输出样例】

```
3 2 7 8 1 4 9 6 10 5
291.90
```

### 【源程序】

```
#include<iostream>
#include<algorithm>
#include<iomanip>
using namespace std;
struct people{
    int num;
    int time;
}arr[5000];
bool cmp(people a,people b)
{
    return a.time<b.time;
}
int main()
{
    int n;
    int i;
    double time_sum=0,time_average;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>arr[i].time;
```

```

    arr[i].num=i;
}
sort(arr+1,arr+1+n,cmp);

for(i=1;i<=n;i++)
{
    cout<<arr[i].num<<" ";
    if(n-i>=1) time_sum+=arr[i].time*(n-i);
}
time_average=time_sum/n;

cout<<endl;
cout<<setiosflags(ios::fixed)<<setprecision(2);
cout<<time_average<<endl;

return 0;
}

```

### 1.1.2.6.2 均分纸牌（信息学奥赛一本通-T1320）

#### 均分纸牌（信息学奥赛一本通-T1320）

原创 Alex\_McAvoy 发布于 2018-05-12 19:11:00 阅读数 806 收藏  
展开

##### 【题目描述】

有  $n$  堆纸牌，编号分别为  $1, 2, \dots, n$ 。每堆上有若干张，但纸牌总数必为  $n$  的倍数。可以在任一堆上取若干张纸牌，然后移动。

移牌规则为：在编号为  $1$  的堆上取的纸牌，只能移到编号为  $2$  的堆上；在编号为  $n$  的堆上取的纸牌，只能移到编号为  $n-1$  的堆上；其他堆上取的纸牌，可以移到相邻左边或右边的堆上。

现在要求找出一种移动方法，用最少的移动次数使每堆上纸牌数都一样多。

例如  $n=4$ ，4 堆纸牌数分别为：① 9 ② 8 ③ 17 ④ 6

移动 3 次可达到目的：

从 ③ 取 4 张牌放到④（9 8 13 10）->从③取 3 张牌放到 ②（9 11 10 10）-> 从②取 1 张牌放到①（10 10 10 10）。

**【输入】**

$n$  ( $n$  堆纸牌,  $1 \leq n \leq 100$ )

$a_1 a_2 \dots a_n$  ( $n$  堆纸牌, 每堆纸牌初始数,  $1 \leq a_i \leq 10000$ )。

**【输出】**

一个正整数, 即最少需要的组数。所有堆均达到相等时的最少移动次数。

**【输入样例】**

```
4
9 8 17 6
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<iostream>
#include<algorithm>
using namespace std;
int main()
{
    int n,a[10000];
    int sum=0,average,step=0;
    int i,j;

    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>a[i];
        sum+=a[i];
    }

    average=sum/n;
    for(i=1;i<=n;i++) a[i]-=average;

    for(i=1;a[i]==0&&i<n;)i++;
    for(j=n;a[j]==0&&j>1;)j--;

    for(;i<j;)
    {
        a[i+1]+=a[i];
        a[i]=0;
        i++;
        step++;
        while(a[i]==0&&i<j)
            i++;
    }
}
```

```

}

cout<<step<<endl;

return 0;
}

```

### 1.1.2.6.3 删数问题（信息学奥赛一本通-T1321）

#### 删数问题（信息学奥赛一本通-T1321）

原创 Alex\_McAvoy 发布于 2018-05-12 19:09:44 阅读数 980 收藏  
展开

##### 【题目描述】

输入一个高精度的正整数  $n$ ，去掉其中任意  $s$  个数字后剩下的数字按原左右次序组成一个新的正整数。编程对给定的  $n$  和  $s$ ，寻找一种方案使得剩下的数字组成的新数最小。

输出新的正整数。（ $n$  不超过 240 位）

输入数据均不需判错。

##### 【输入】

$n$  和  $s$

##### 【输出】

一个正整数，即最少需要的组数。

##### 【输入样例】

```

175438
4

```

##### 【输出样例】

```

13

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define N 20
using namespace std;

```

```

int main()
{
    int t;
    char str[260];
    int i,j;

    cin>>str;
    cin>>t;
    int len=strlen(str);
    while(t--)
    {
        for(i=0;i<=len-2;i++)
            if(str[i]>str[i+1])
            {
                for(j=i;j<=len-2;j++)
                    str[j]=str[j+1];
                break;
            }
        len--;
    }
    i=0;
    while(i<=len-1&&str[i]!='0')
        i++;
    if(i==len)
        cout<<"0"<<endl;
    else
        for(j=i;j<=len-1;j++)
            cout<<str[j];
    return 0;
}

```

### 1.1.2.6.4 拦截导弹问题（信息学奥赛一本通-T1322）

#### 拦截导弹问题（信息学奥赛一本通-T1322）

原创 Alex\_McAvoy 发布于 2018-05-12 19:08:16 阅读数 1431 收藏  
展开

##### 【题目描述】

某国为了防御敌国的导弹袭击，开发出一种导弹拦截系统，但是这种拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前

一发的高度。某天，雷达捕捉到敌国的导弹来袭，由于该系统还在试用阶段。所以一套系统有可能不能拦截所有的导弹。

输入导弹依次飞来的高度（雷达给出的高度不大于 30000 的正整数）。计算要拦截所有导弹最小需要配备多少套这种导弹拦截系统。

**【输入】**

$n$  颗依次飞来的高度（ $1 \leq n \leq 1000$ ）。

**【输出】**

要拦截所有导弹最小配备的系统数  $k$ 。

**【输入样例】**

389 207 155 300 299 170 158 65

**【输出样例】**

2

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define N 200
using namespace std;
int main()
{
    int a[N]={0};
    int b[N]={0};
    int n;
    int maxx;

    int t=0;
    while(scanf("%d",&n)!=EOF)
    {
        a[t]=n;
        t++;
    }
    b[0]=1;
    for(int i=1;i<t;i++)
    {
        b[i]=1;
        for(int j=0;j<i;j++)
            if(a[j]<a[i]&& b[j]+1>b[i])
                b[i]=b[j]+1;
    }
}
```

```

}
maxx=1;
for(int i=0;i<t;i++)
    if(maxx<b[i])
        maxx=b[i];
cout<<maxx<<endl;
return 0;
}

```

点赞

### 1.1.2.6.5 活动选择（信息学奥赛一本通-T1323）

#### 活动选择（信息学奥赛一本通-T1323）

原创 Alex\_McAvoy 发布于 2018-05-12 19:07:07 阅读数 1129 收藏  
展开

##### 【题目描述】

学校在最近几天有  $n$  个活动，这些活动都需要使用学校的大礼堂，在同一时间，礼堂只能被一个活动使用。由于有些活动时间上有冲突，学校办公室人员只好让一些活动放弃使用礼堂而使用其他教室。

现在给出  $n$  个活动使用礼堂的起始时间  $begin_i$  和结束时间  $endi$  ( $begin_i < end_i$ )，请你帮助办公室人员安排一些活动来使用礼堂，要求安排的活动尽量多。

##### 【输入】

第一行一个整数  $n$  ( $n \leq 1000$ )；

接下来的  $n$  行，每行两个整数，第一个  $begin_i$ ，第二个是  $endi$  ( $begin_i < end_i \leq 32767$ )。

##### 【输出】

输出最多能安排的活动个数。

##### 【输入样例】

```

11
3 5
1 4
12 14
8 12
0 6
8 11
6 10
5 7

```

3 8  
5 9  
2 13

### 【输出样例】

4

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 1001
using namespace std;
int n;
int beginn[N],endd[N];
void qsort(int x,int y)
{
    int i,j,mid;
    i=x;
    j=y;
    mid=endd[(x+y)/2];
    while(i<=j)
    {
        while(endd[i]<mid)
            i++;
        while(endd[j]>mid)
            j--;
        if(i<=j)
        {
            swap(endd[j],endd[i]);
            swap(beginn[j],beginn[i]);
            i++;
            j--;
        }
    }
    if(x<j)
        qsort(x,j);
    if(i<y)
        qsort(i,y);
}
int main()
{
```



```

cin>>n;
for(int i=1;i<=n;i++)
    cin>>beginn[i]>>endd[i];
qsort(1,n);

int cnt=0;
int temp=-INF;
for(int i=1;i<=n;i++)
    if(beginn[i]>=temp)
    {
        cnt++;
        temp=endd[i];
    }
cout<<cnt<<endl;
return 0;
}

```

### 1.1.2.6.6 整数区间（信息学奥赛一本通-T1324）

#### 整数区间（信息学奥赛一本通-T1324）

原创 Alex\_McAvoy 发布于 2018-05-12 19:05:35 阅读数 1225 收藏  
展开

##### 【题目描述】

请编程完成以下任务：

- 1.从文件中读取闭区间的个数及它们的描述；
- 2.找到一个含元素个数最少的集合,使得对于每一个区间,都至少有一个整数属于该集合，输出该集合的元素个数。

##### 【输入】

首行包括区间的数目  $n$ ,  $1 \leq n \leq 10000$ , 接下来的  $n$  行, 每行包括两个整数  $a, b$ , 被一空格隔开,  $0 \leq a \leq b \leq 10000$ , 它们是某一个区间的开始值和结束值。

##### 【输出】

第一行集合元素的个数, 对于每一个区间都至少有一个整数属于该区间, 且集合所包含元素数目最少。

##### 【输入样例】

```

4
3 6

```

2 4  
0 2  
4 7

### 【输出样例】

2

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 1001
using namespace std;
int n;
int a[N],b[N];
void qsort(int x,int y)
{
    int i,j,mid1,mid2;
    i=x;
    j=y;
    mid1=b[(x+y)/2];
    mid2=a[(x+y)/2];
    while(i<=j)
    {
        while( b[i]<mid1 || (b[i]==mid1&& a[i]<mid2) )
            i++;
        while( b[j]>mid1 || (b[j]==mid1&& a[j]>mid2) )
            j--;
        if(i<=j)
        {
            swap(a[j],a[i]);
            swap(b[j],b[i]);
            i++;
            j--;
        }
    }
    if(x<j)
        qsort(x,j);
    if(i<y)
        qsort(i,y);
}
int main()
```

```

{
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i]>>b[i];
    qsort(1,n);

    int cnt=0;
    int temp=-INF;
    for(int i=1;i<=n;i++)
        if(a[i]>temp)
        {
            cnt++;
            temp=b[i];
        }
    cout<<cnt<<endl;
    return 0;
}
点

```

### 1.1.2.6.7 An Easy Problem（信息学奥赛一本通-T1223）

#### An Easy Problem（信息学奥赛一本通-T1223）

原创 Alex\_McAvoy 发布于 2018-05-12 19:04:20 阅读数 1436 收藏  
展开

##### 【题目描述】

给定一个正整数  $N$ ，求最小的、比  $N$  大的正整数  $M$ ，使得  $M$  与  $N$  的二进制表示中有相同数目的 1。

举个例子，假如给定的  $N$  为 78，其二进制表示为 1001110，包含 4 个 1，那么最小的比  $N$  大的并且二进制表示中只包含 4 个 1 的数是 83，其二进制是 1010011，因此 83 就是答案。

##### 【输入】

输入若干行，每行一个数  $n(1 \leq n \leq 1000000)$ ，输入 "0" 结束。

##### 【输出】

输出若干行对应的值。

##### 【输入样例】

```

1
2

```

3  
4  
78  
0

【输出样例】

2  
4  
5  
8  
83

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 1001
using namespace std;
int main()
{
    int n;
    while(scanf("%d",&n)!=EOF&&n)
    {
        int cnt=0;
        int temp=n;
        while(temp>0)
        {
            if(temp%2)
                cnt++;
            temp/=2;
        }
        int sum=0;
        while(1)
        {
            temp=++n;
            while(temp>0)
            {
                if(temp%2)
                    sum++;
                temp/=2;
            }
            if(cnt==sum)
```

```

    {
        cout<<n<<endl;
        break;
    }
    sum=0;
}
}
return 0;
}

```

### 1.1.2.6.8 最大子矩阵（信息学奥赛一本通-T1224）

#### 最大子矩阵（信息学奥赛一本通-T1224）

原创 Alex\_McAvoy 发布于 2018-05-13 13:45:26 阅读数 1680 收藏  
展开

##### 【题目描述】

已知矩阵的大小定义为矩阵中所有元素的和。给定一个矩阵，你的任务是找到最大的非空(大小至少是  $1 \times 1 \times 1$ )子矩阵。

比如，如下  $4 \times 4$  的矩阵

```

0  -2  -7  0
9   2  -6  2
-4   1  -4  1
-1   8   0 -2

```

的最大子矩阵是

```

9   2
-4   1
-1   8

```

这个子矩阵的大小是 15。

##### 【输入】

输入是一个  $N \times N$  的矩阵。输入的第一行给出  $N(0 < N \leq 100)$ 。再后面的若干行中，依次(首先从左到右给出第一行的  $N$  个整数，再从左到右给出第二行的  $N$  个整数……)给出

矩阵中的  $N^2$  个整数，整数之间由空白字符分隔(空格或者空行)。已知矩阵中整数的范围都在  $[-127, 127]$ 。

**【输出】**

输出最大子矩阵的大小。

**【输入样例】**

```
4
0 -2 -7 0
9 2 -6 2
-4 1 -4 1
-1 8 0 -2
```

**【输出样例】**

```
15
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 1001
using namespace std;
int a[N][N],b[N][N];
int main()
{
    int n;
    int x,y;
    int num,sum;

    cin>>n;
    for(int i=1;i<=n*n;i++)
    {
        cin>>num;
        x=(i-1)/n+1;
        y=i%n;
        if(y==0)
            y=n;
        a[x][y]=num;
    }

    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
```

```

        b[i][j]=b[i-1][j]+a[i][j];
int max=b[1][1];
for(int i=1;i<=n;i++)
    for(int j=i;j<=n;j++)
    {
        sum=0;
        for(int k=1;k<=n;k++)
        {
            if(sum>0)
                sum+=b[j][k]-b[i-1][k];
            else
                sum=b[j][k]-b[i-1][k];
            if(sum>max)
                max=sum;
        }
    }
cout<<max<<endl;
return 0;
}

```

### 1.1.2.6.9 金银岛（信息学奥赛一本通-T1225）

#### 金银岛（信息学奥赛一本通-T1225）

原创 Alex\_McAvoy 发布于 2018-05-13 13:40:58 阅读数 1028 收藏  
展开

##### 【题目描述】

某天 KID 利用飞行器飞到了一个金银岛上，上面有许多珍贵的金属，KID 虽然更喜欢各种宝石的艺术品，可是也不拒绝这样珍贵的金属。但是他只带着一个口袋，口袋至多只能装重量为  $w$  的物品。岛上金属有  $s$  个种类，每种金属重量不同，分别为  $n_1, n_2, \dots, n_s$ ，同时每个种类的金属总的价值也不同，分别为  $v_1, v_2, \dots, v_s$ 。KID 想一次带走价值尽可能多的金属，问他最多能带走价值多少的金属。注意到金属是可以被任意分割的，并且金属的价值和其重量成正比。

##### 【输入】

第 1 行是测试数据的组数  $k$ ，后面跟着  $k$  组输入。

每组测试数据占 3 行，第 1 行是一个正整数  $w(1 \leq w \leq 10000)$ ，表示口袋承重上限。第 2 行是一个正整数  $s(1 \leq s \leq 100)$ ，表示金属种类。第 3 行有  $2s$  个正整数，分别为  $n_1, v_1, n_2, v_2, \dots, n_s, v_s$  分别为第一种，第二种，...，第  $s$  种金属的总重量和总价值 ( $1 \leq n_i \leq 10000, 1 \leq v_i \leq 10000$ )。

**【输出】**

k 行，每行输出对应一个输入。输出应精确到小数点后 2 位。

**【输入样例】**

```
2
50
4
10 100 50 30 7 34 87 100
10000
5
1 43 43 323 35 45 43 54 87 43
```

**【输出样例】**

```
171.93
508.00
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 101
using namespace std;
struct node{
    int w;
    int c;
    double g;
}a[N],temp;
int main()
{
    int t;
    int W,m;

    cin>>t;
    while(t-->0)
    {
        double cnt=0;
        cin>>W>>m;
        for(int i=1;i<=m;i++)
        {
            cin>>a[i].w>>a[i].c;
            a[i].g=a[i].c*1.0/a[i].w;
        }
    }
}
```



```

for(int i=1;i<=m;i++)
    for(int j=i+1;j<=m;j++)
        if(a[i].g<a[j].g)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
for(int i=1;i<=m;i++)
{
    if(W>=a[i].w)
    {
        cnt+=a[i].c;
        W-=a[i].w;
    }
    else
    {
        cnt+=a[i].g*W;
        break;
    }
}
printf("%.2lf\n",cnt);
}
return 0;
}

```

### 1.1.2.6.10 装箱问题（信息学奥赛一本通-T1226）

#### 装箱问题（信息学奥赛一本通-T1226）

原创 Alex\_McAvoy 发布于 2018-05-13 13:34:14 阅读数 1743 收藏  
展开

##### 【题目描述】

一个工厂制造的产品形状都是长方体，它们的高度都是  $h$ ，长和宽都相等，一共有六个型号，他们的长宽分别为  $1*1, 2*2, 3*3, 4*4, 5*5, 6*6$ 。这些产品通常使用一个  $6*6*h$  的长方体包裹包装然后邮寄给客户。因为邮费很贵，所以工厂要想方设法的减小每个订单运送时的包裹数量。他们很需要有一个好的程序帮他们解决这个问题从而节省费用。现在这个程序由你来设计。

##### 【输入】

输入文件包括几行，每一行代表一个订单。每个订单里的一行包括六个整数，中间用

空格隔开，分别为 1\*1 至 6\*6 这六种产品的数量。输入文件将以 6 个 0 组成的一行结尾。

### 【输出】

除了输入的最后一行 6 个 0 以外，输入文件里每一行对应着输出文件的一行，每一行输出一个整数代表对应的订单所需的最小包裹数。

### 【输入样例】

```
0 0 4 0 0 1
7 5 1 0 0 0
0 0 0 0 0 0
```

### 【输出样例】

```
2
1
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 101
using namespace std;

int main()
{
    int a,b,c,d,e,f;
    int g[4]={0,5,3,1};
    int x,y;

    while(cin>>a>>b>>c>>d>>e>>f)
    {
        if(a+b+c+d+e+f==0)
            break;
        int n=f+e+d+(c+3)/4;
        x=5*d+g[c%4];
        if(b>x)
            n+=(b-x+8)/9;
        y=36*n-36*f-25*e-16*d-9*c-4*b;
        if(a>y)
            n+=(a-y+35)/36;
        cout<<n<<endl;
    }
}
```

```

    return 0;
}

```

点赞 1

## 1.1.2.6.11 Ride to Office（信息学奥赛一本通-T1227）

### Ride to Office（信息学奥赛一本通-T1227）

原创 Alex\_McAvoy 发布于 2018-05-13 13:33:06 阅读数 1177 收藏  
展开

#### 【题目描述】

起点与终点相隔 4500 米。现 Charley 需要从起点骑车到终点。但是，他有个习惯，沿途需要有人陪伴，即以相同的速度，与另外一个人一起骑。而当他遇到以更快的速度骑车的人时，他会以相应的速度跟上这个更快的人。先给定所有与 Charley 同路的人各自的速度与出发时间，问 Charley 以这种方式跟人，骑完 4500 米需要多少时间。得出的结果若是小数，则向上取整。

#### 【输入】

输入若干组数据，每组数据第一行  $n(1 \leq n \leq 10000)$ ,  $n$  为 0，表示输入结束，接着输入  $n$  行数据，每行 2 个数据，表示速度  $v$  和出发时间  $t$ ，如果  $t < 0$ ，表示陪伴人提早出发了。

#### 【输出】

输出对应若干行数据，每行输出 1 个数，表示最快到达的时间。

#### 【输入样例】

```

4
20 0
25 -155
27 190
30 240
2
21 0
22 34
0

```

#### 【输出样例】

```

780
771

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 10001
using namespace std;
struct node{
    int v;
    int start;
    int endd;
}a[N];
int main()
{
    int n;
    int t;

    while(cin>>n&& n)
    {
        for(int i=1;i<=n;i++)
        {
            cin>>a[i].v>>a[i].start;
            t=16200/a[i].v;
            if(16200.0/a[i].v>t)
                t++;
            a[i].endd=a[i].start+t;
        }
        int min=INF;
        for(int i=1;i<=n;i++)
            if(a[i].start==0&&a[i].endd<min)
                min=a[i].endd;
        for(int i=1;i<=n;i++)
            if(a[i].start>0&&a[i].endd<min)
                min=a[i].endd;
        cout<<min<<endl;
    }
    return 0;
}

```

## 1.1.2.6.12 书架（信息学奥赛一本通-T1228）

### 书架（信息学奥赛一本通-T1228）

原创 Alex\_McAvoy 发布于 2018-05-13 13:32:01 阅读数 1196 收藏  
展开

#### 【题目描述】

John 最近买了一个书架用来存放奶牛养殖书籍，但书架很快被存满了，只剩最顶层有空余。

John 共有  $N$  头奶牛( $1 \leq N \leq 20,000$ )，每头奶牛有自己的高度  $H_i$ ( $1 \leq H_i \leq 10,000$ )， $N$  头奶牛的总高度为  $S$ 。书架高度为  $B$ ( $1 \leq B \leq S < 2,000,000,007$ )。

为了到达书架顶层，奶牛可以踩着其他奶牛的背，像叠罗汉一样，直到他们的总高度不低于书架高度。当然若奶牛越多则危险性越大。为了帮助 John 到达书架顶层，找出使用奶牛数目最少的解决方案吧。

#### 【输入】

第 1 行：空格隔开的整数  $N$  和  $B$ 。

第 2~ $N+1$  行：第  $i+1$  行为整数  $H_i$ 。

#### 【输出】

能达到书架高度所使用奶牛的最少数目。

#### 【输入样例】

```
6 40
6
18
11
13
19
11
```

#### 【输出样例】

```
3
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
```

```

#define INF 999999999
#define N 20001
using namespace std;
int a[N];
void qsort(int x,int y)
{
    int i=x,j=y,mid=a[(x+y)/2];
    while(i<=j)
    {
        while(a[i]>mid)
            i++;
        while(a[j]<mid)
            j--;
        if(i<=j)
        {
            swap(a[i],a[j]);
            i++;
            j--;
        }
    }
    if(x<j)
        qsort(x,j);
    if(i<y)
        qsort(i,y);
}
int main()
{
    int n,b;
    int sum=0;
    int i;

    cin>>n>>b;
    for(i=1;i<=n;i++)
        cin>>a[i];
    qsort(1,n);
    for(i=1;i<=n;i++)
    {
        sum+=a[i];
        if(sum>=b)
            break;
    }
    cout<<i<<endl;;
    return 0;
}

```

## 1.1.2.6.13 电池的寿命（信息学奥赛一本通-T1229）

### 电池的寿命（信息学奥赛一本通-T1229）

原创 Alex\_McAvoy 发布于 2018-05-13 14:32:09 阅读数 1456 收藏  
展开

#### 【题目描述】

小 S 新买了一个掌上游戏机，这个游戏机由两节 5 号电池供电。为了保证能够长时间玩游戏，他买了很多 5 号电池，这些电池的生产商不同，质量也有差异，因而使用寿命也有所不同，有的能使用 5 个小时，有的可能就只能使用 3 个小时。显然如果他只有两个电池一个能用 5 小时一个能用 3 小时，那么他只能玩 3 个小时的游戏，有一个电池剩下的电量无法使用，但是如果他有更多的电池，就可以更加充分地利用它们，比如他有三个电池分别能用 3、3、5 小时，他可以先使用两节能用 3 个小时的电池，使用半个小时后再把其中一个换成能使用 5 个小时的电池，两个半小时后再把剩下的一节电池换成刚才换下的电池（那个电池还能用 2.5 个小时），这样总共就可以使用 5.5 个小时，没有一点浪费。

现在已知电池的数量和电池能够使用的时间，请你找一种方案使得使用时间尽可能的长。

#### 【输入】

输入包含多组数据。每组数据包括两行，第一行是一个整数  $N(2 \leq N \leq 1000)$ ，表示电池的数目，接下来一行是  $N$  个正整数表示电池能使用的时间。

#### 【输出】

对每组数据输出一行，表示电池能使用的时间，保留到小数点后 1 位。

#### 【输入样例】

```
2
3 5
3
3 3 5
```

#### 【输出样例】

```
3.0
5.5
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
```

```

#define INF 999999999
#define N 1001
using namespace std;
int a[N];
int main()
{
    int n;
    while(cin>>n)
    {
        int sum=0;
        int max=-INF;
        for(int i=1;i<=n;i++)
        {
            cin>>a[i];
            if(a[i]>max)
                max=a[i];
            sum+=a[i];
        }
        if(max>=sum-max)
            printf("%.1f\n",(sum-max)*1.0);
        else
            printf("%.1f\n",sum*1.0/2);
    }
    return 0;
}

```

### 1.1.2.6.14 寻找平面上的极大点（信息学奥赛一本通-T1230）

#### 寻找平面上的极大点（信息学奥赛一本通-T1230）

原创 Alex\_McAvoy 发布于 2018-05-13 14:41:15 阅读数 1032 收藏  
展开

##### 【题目描述】

在一个平面上，如果有两个点 $(x,y)$ 、 $(a,b)$ ，如果说 $(x,y)$ 支配了 $(a,b)$ ，这是指  $x \geq a, y \geq b$ ；

用图形来看就是 $(a,b)$ 坐落在以 $(x,y)$ 为右上角的一个无限的区域内。

给定  $n$  个点的集合，一定存在若干个点，它们不会被集合中的任何一点所支配，这些点叫做极大值点。



编程找出所有的极大点，按照  $x$  坐标由小到大，输出极大点的坐标。

本题规定： $n$  不超过 100，并且不考虑点的坐标为负数的情况。

#### 【输入】

输入包括两行，第一行是正整数  $n$ ，表示是点数，第二行包含  $n$  个点的坐标，坐标值都是整数，坐标范围从 0 到 100，输入数据中不存在坐标相同的点。

#### 【输出】

按  $x$  轴坐标最小到大的顺序输出所有极大点。

输出格式为： $(x_1,y_1),(x_2,y_2),\dots,(x_k,y_k)(x_1,y_1),(x_2,y_2),\dots,(x_k,y_k)$ 。

注意：输出的每个点之间有","分隔,最后一个点之后没有",",少输出和多输出都会被判错。

#### 【输入样例】

```
5
1 2 2 2 3 1 2 3 1 4
```

#### 【输出样例】

```
(1,4),(2,3),(3,1)
```

#### 【提示】

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 101
using namespace std;
int x[N],y[N],vis[N];
int main()
{
    int n,cnt=0;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>x[i]>>y[i];
    for(int i=1;i<=n;i++)
    {
        for(int j=i+1;j<=n;j++)
        {
            if(x[i]>x[j])
```

```

    {
        swap(x[i],x[j]);
        swap(y[i],y[j]);
    }
    else if(x[i]==x[j]&& y[i]>y[j])
    {
        swap(x[i],x[j]);
        swap(y[i],y[j]);
    }
}
}
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        if(i!=j&&x[j]>=x[i]&&y[j]>=y[i])
        {
            vis[i]=1;
            break;
        }
for(int i=1;i<=n;i++)
    if(vis[i]==0)
    {
        cnt++;
        if(cnt==1)
            printf("(%d,%d)",x[i],y[i]);
        else
            printf(",(%d,%d)",x[i],y[i]);
    }
return 0;
}

```

### 1.1.2.6.15 最小新整数（信息学奥赛一本通-T1231）

#### 最小新整数（信息学奥赛一本通-T1231）

原创 Alex\_McAvoy 发布于 2018-05-13 14:42:28 阅读数 887 收藏  
展开

##### 【题目描述】

给定一个十进制正整数  $n$  ( $0 < n < 1000000000$ )，每个数位上数字均不为 0。 $n$  的位数为  $m$ 。

现在从  $m$  位中删除  $k$  位 ( $0 < k < m$ )，求生成的新整数最小为多少？

例如: $n=9128456, k=2$ , 则生成的新整数最小为 12456。

#### 【输入】

第一行  $t$ , 表示有  $t$  组数据;

接下来  $t$  行, 每一行表示一组测试数据, 每组测试数据包含两个数字  $n, k$ 。

#### 【输出】

$t$  行, 每行一个数字, 表示从  $n$  中删除  $k$  位后得到的最小整数。

#### 【输入样例】

```
2
9128456 2
1444 3
```

#### 【输出样例】

```
123456
1
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 101
using namespace std;
char str[N];
int main()
{
    int t,n;
    cin>>t;
    while(t--)
    {
        cin>>str>>n;
        int len=strlen(str);
        while(n--)
        {
            for(int i=0;i<len-1;i++)
                if(str[i]>str[i+1])
                {
                    for(int j=i;j<len-1;j++)
                        str[j]=str[j+1];
                    break;
                }
        }
    }
}
```

```

        len--;
    }
    str[len]='\0';
    cout<<str<<endl;
}
return 0;
}

```

点赞

## 1.1.2.6.16 Crossing River（信息学奥赛一本通-T1232）

### Crossing River（信息学奥赛一本通-T1232）

原创 Alex\_McAvoy 发布于 2018-05-13 14:36:41 阅读数 1525 收藏  
展开

#### 【题目描述】

几个人过河，每次过两人一人回，速度由慢者决定，问过河所需最短时间。

#### 【输入】

输入  $t$  组数据，每组数据第 1 行输入  $n$ ，第 2 行输入  $n$  个数，表示每个人过河的时间。

#### 【输出】

输出  $t$  行数据，每行 1 个数，表示每组过河最少时间。

#### 【输入样例】

```

1
4
1 2 5 10

```

#### 【输出样例】

```

17

```

#### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 1001
using namespace std;
int a[N],dp[N];

```

```

int main()
{
    int t;
    int n;
    cin>>t;
    while(t-->0)
    {
        cin>>n;
        for(int i=0;i<n;i++)
            cin>>a[i];
        sort(a,a+n);
        dp[0]=a[0];
        dp[1]=a[1];
        for(int i=2;i<n;i++)
            dp[i]=min(dp[i-1]+a[i],dp[i-2]+a[i-1]+a[i]*2);
        cout<<dp[n-1]<<endl;
    }
    return 0;
}

```

### 1.1.2.6.17 接水问题（信息学奥赛一本通-T1233）

#### 接水问题（信息学奥赛一本通-T1233）

原创 Alex\_McAvoy 发布于 2018-05-13 14:56:54 阅读数 1420 收藏  
展开

##### 【题目描述】

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  $w_i$ 。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  $w_j$  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。

##### 【输入】

第 1 行 2 个整数  $n$  和  $m$ ，用一个空格隔开，分别表示接水人数和龙头个数。

第 2 行  $n$  个整数  $w_1, w_2, \dots, w_n$ , 每两个整数之间用一个空格隔开,  $w_i$  表示  $i$  号同学的接水量。

**【输出】**

输出只有一行, 1 个整数, 表示接水所需的总时间。

**【输入样例】**

```
5 3
4 4 1 2 1
```

**【输出样例】**

```
4
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#define INF 999999999
#define N 10001
using namespace std;
int a[N],s[101];
int main()
{
    int n,m;
    int k;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    for(int i=1;i<=n;i++)
    {
        int min=INF;
        for(int j=1;j<=m;j++)
            if(s[j]<min)
            {
                min=s[j];
                k=j;
            }
        s[k]+=a[i];
    }
    int max=-INF;
    for(int i=1;i<=m;i++)
        if(s[i]>max)
            max=s[i];
```

```
cout<<max<<endl;
return 0;
}
```

点赞

## 1.1.2.7 第七章 分治算法

### 1.1.2.7.1 循环比赛日程表（信息学奥赛一本通-T1325）

#### 循环比赛日程表（信息学奥赛一本通-T1325）

原创 Alex\_McAvoy 发布于 2018-05-15 22:51:12 阅读数 1418 收藏  
展开

##### 【题目描述】

设有  $N$  个选手进行循环比赛，其中  $N=2M$ ，要求每名选手要与其他  $N-1$  名选手都赛一次，每名选手每天比赛一次，循环赛共进行  $N-1$  天，要求每天没有选手轮空。

##### 【输入】

输入：M。

##### 【输出】

输出：表格形式的比赛安排表。一行各数据间用一个空格隔开。

##### 【输入样例】

3

##### 【输出样例】

```
1 2 3 4 5 6 7 8
2 1 4 3 6 5 8 7
3 4 1 2 7 8 5 6
4 3 2 1 8 7 6 5
5 6 7 8 1 2 3 4
6 5 8 7 2 1 4 3
7 8 5 6 3 4 1 2
8 7 6 5 4 3 2 1
```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1001
#define MOD 1000000007
using namespace std;
int a[N][N];
int main()
{
    int m;
    int k=1,half=1;
    cin>>m;

    int n=1<<m;
    a[0][0]=1;
    while(k<=m)
    {
        for(int i=0;i<half;i++)
            for(int j=0;j<half;j++)
                a[i][j+half]=a[i][j]+half;

        for(int i=0;i<half;i++)
            for(int j=0;j<half;j++)
            {
                a[i+half][j]=a[i][j+half];
                a[i+half][j+half]=a[i][j];
            }
        half*=2;
        k++;
    }
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            printf("%d ",a[i][j]);
        cout<<endl;
    }
    return 0;
}

```



## 1.1.2.7.2 取余运算（信息学奥赛一本通-T1326）

### 取余运算（信息学奥赛一本通-T1326）

原创 Alex\_McAvoy 发布于 2018-05-15 22:52:23 阅读数 725 收藏  
展开

#### 【题目描述】

输入  $b$ ,  $p$ ,  $k$  的值, 求  $b^p \bmod k$  的值。其中  $b$ ,  $p$ ,  $k \times k$  为长整型数。

#### 【输入】

输入  $b$ ,  $p$ ,  $k$  的值。

#### 【输出】

求  $b^p \bmod k$  的值。

#### 【输入样例】

2 10 9

#### 【输出样例】

$2^{10} \bmod 9 = 7$

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1001
#define MOD 1000000007
using namespace std;
int b,p,k;
int calculate(int p)
{
    int temp;

    if(p==0)
        return 1;
    temp=calculate(p/2)%k;
    temp=(temp*temp)%k;
    if(p%2==1)
```

```

        temp=(temp*b)%k;
    return temp;
}
int main()
{
    cin>>b>>p>>k;
    int temp=b;
    b%=k;
    cout<<temp<<"^"<<p<<" mod "<<k<<"="<<calculate(p)<<endl;
    return 0;
}

```

### 1.1.2.7.3 黑白棋子的移动（信息学奥赛一本通-T1327）

#### 黑白棋子的移动（信息学奥赛一本通-T1327）

原创 Alex\_McAvoy 发布于 2018-05-15 22:53:38 阅读数 1094 收藏  
展开

##### 【题目描述】

有  $2n$  个棋子 ( $n \geq 4$ ) 排成一行，开始位置为白子全部在左边，黑子全部在右边，如下图为  $n=5$  的情形：○○○○●●●●●

移动棋子的规则是：每次必须同时移动相邻的两个棋子，颜色不限，可以左移也可以右移到空位上去，但不能调换两个棋子的左右位置。每次移动必须跳过若干个棋子（不能平移），要求最后能移成黑白相间的一行棋子。如  $n=5$  时，成为：○●○●○●○●○●

任务：编程打印出移动过程。

##### 【输入】

输入  $n$ 。

##### 【输出】

移动过程。

##### 【输入样例】

7

##### 【输出样例】

```

step 0:oooooooo*****--
step 1:oooooo--*****o*

```

```

step 2:000000*****--o*
step 3:000000--*****o*o*
step 4:000000*****--o*o*
step 5:000000--****o*o*o*
step 6:0000****--o*o*o*
step 7:000--***o*o*o*o*
step 8:000*o**--*o*o*o*
step 9:o--*o**oo*o*o*o*
step10:o*o*o*--o*o*o*o*
step11:--o*o*o*o*o*o*o*

```

### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1001
#define MOD 1000000007
using namespace std;
int n,st,sp;
char a[N];
void print()
{
    printf("step%2d:",st++);
    for(int i=1;i<=2*n+2;i++)
        printf("%c",a[i]);
    printf("\n");
}
void move(int m)
{
    a[sp]=a[m];
    a[sp+1]=a[m+1];
    a[m]=a[m+1]='-';
    sp=m;
    print();
}
void mv(int m)
{
    if(m==4)
    {
        move(4);
        move(8);
    }
}

```

```

        move(2);
        move(7);
        move(1);
    }
    else
    {
        move(m);
        move(2*m-1);
        mv(m-1);
    }
}
int main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        a[i]='o';
    for(int i=n+1;i<=2*n;i++)
        a[i]='*';
    for(int i=2*n+1;i<=2*n+2;i++)
        a[i]='-';
    st=0;
    sp=2*n+1;
    print();
    mv(n);
    return 0;
}

```

#### 1.1.2.7.4 光荣的梦想（信息学奥赛一本通-T1328）

##### 光荣的梦想（信息学奥赛一本通-T1328）

原创 Alex\_McAvoy 发布于 2018-05-17 21:40:40 阅读数 1488 收藏  
展开

##### 【题目描述】

Prince 对他在大陆上维护的秩序感到满意，于是决定启程离开艾泽拉斯。在他动身之前，Prince 决定赋予 King\_Bette 最强大的能量以守护世界、保卫这里的平衡与和谐。在那个时代，平衡是个梦想。因为有很多奇异的物种拥有各种不稳定的能量，平衡瞬间即被打破。KB 决定求助于你，帮助他完成这个梦想。

一串数列即表示一个世界的状态。

平衡是指这串数列以升序排列。而从一串无序数列到有序数列需要通过交换数列中的元素来实现。KB 的能量只能交换相邻两个数字。他想知道他最少需要交换几次就能使数列有序。

#### 【输入】

第一行为数列中数的个数  $n$ , 第二行为  $n \leq 10000$  个数。表示当前数列的状态。

#### 【输出】

输出一个整数，表示最少需要交换几次能达到平衡状态。

#### 【输入样例】

```
4
2 1 4 3
```

#### 【输出样例】

```
2
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
using namespace std;
int n;
int a[N],b[N];
long long ans=0;
void msort(int left,int right)
{
    int mid=(left+right)/2;
    if(left>=right)
        return ;
    msort(left,mid);
    msort(mid+1,right);

    int i=left,j=mid+1,n=mid,m=right,k=0;
    while(i<=n&&j<=m)
        if(a[i]>a[j])
        {
            ans+=n-i+1;
            b[k++]=a[j++];
        }
}
```

```

        else
            b[k++]=a[i++];
    while(i<=n)
        b[k++]=a[i++];
    while(j<=m)
        b[k++]=a[j++];
    for(i=0;i<k;i++)
        a[left+i]=b[i];
}
int main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    msort(1,n);
    cout<<ans<<endl;
    return 0;
}

```

### 1.1.2.7.5 2011（信息学奥赛一本通-T1234）

#### 2011（信息学奥赛一本通-T1234）

原创 Alex\_McAvoy 发布于 2018-05-17 21:42:18 阅读数 1629 收藏  
展开

##### 【题目描述】

已知长度最大为 200 位的正整数  $n$ ，请求出  $2011n$  的后四位。

##### 【输入】

第一行为一个正整数  $k$ ，代表有  $k$  组数据 ( $k \leq 200$ )，接下来的  $k$  行，每行都有一个正整数  $n$ ， $n$  的位数  $\leq 200$ 。

##### 【输出】

每一个  $n$  的结果为一个整数占一行，若不足 4 位，去除高位多余的 0。

##### 【输入样例】

```

3
5
28
792

```

## 【输出样例】

1051

81

5521

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 500
#define MOD 1000000007
using namespace std;
char a[N];
int main()
{
    int k;
    cin>>k;
    while(k--)
    {
        cin>>a;
        int len=strlen(a);
        int B=0,C=2011;
        int i;

        for(i=len-4;i<len;i++)
            if(i>=0)
                B=B*10+a[i]-'0';
        i=1;
        do{
            if(i*2<=B)
            {
                i*=2;
                C=(C*C)%10000;
            }
        }while(i*2<=B);
        for(;i<B;i++)
            C=(C*2011)%10000;
        cout<<C<<endl;
    }
    return 0;
}

```

}

点赞

## 1.1.2.7.6 输出前 k 大的数(信息学奥赛一本通-T1235)

### 输出前 k 大的数（信息学奥赛一本通-T1235）

原创 Alex\_McAvoy 发布于 2018-05-17 21:43:47 阅读数 1326 收藏  
展开

#### 【题目描述】

给定一个数组，统计前 k 大的数并且把这 k 个数从大到小输出。

#### 【输入】

第一行包含一个整数 n，表示数组的大小。n < 100000。

第二行包含 n 个整数，表示数组的元素，整数之间以一个空格分开。每个整数的绝对值不超过 100000000。

第三行包含一个整数 k，k < n。

#### 【输出】

从大到小输出前 k 大的数，每个数一行。

#### 【输入样例】

```
10
4 5 6 9 8 7 1 2 3 0
5
```

#### 【输出样例】

```
9
8
7
6
5
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
```



```

#define N 1000001
#define MOD 1000000007
using namespace std;

int a[N];

int cmp(const void *a,const void *b){
    return (*(int *)a) - (*(int *)b);
}

void Find(int st,int ed,int k){
    if(st-ed+1==k)
        return;

    inti=st,j=ed,key=a[st];
    while(i<j){
        while(i<j&& a[j]>=key)
            j--;
        a[i]=a[j];
        while(i<j&& a[i]<=key)
            i++;
        a[j]=a[i];
    }
    a[i]=key;
    if(ed-i+1==k)
        return;
    else if(ed-i+1>k)
        Find(i+1,ed,k);
    else
        Find(st,i-1,k-(ed-i+1));
}

int main(){
    int n,k;
    scanf("%d",&n);
    for(int i=0; i<n; i++)
        scanf("%d",&a[i]);
    scanf("%d",&k);

    Find(0,n-1,k);

    qsort(a+n-k,k,sizeof(a[0]),cmp);
    for(int i=n-1; i>=n-k; i--)
        printf("%d\n",a[i]);
}

```

```

    return 0;
}

```

### 1.1.2.7.7 区间合并（信息学奥赛一本通-T1236）

#### 区间合并（信息学奥赛一本通-T1236）

原创 Alex\_McAvoy 发布于 2018-05-17 21:45:24 阅读数 1215 收藏  
展开

##### 【题目描述】

给定  $n$  个闭区间  $[a_i; b_i]$ ，其中  $i=1,2,\dots,n$ 。任意两个相邻或相交的闭区间可以合并为一个闭区间。例如， $[1;2]$  和  $[2;3]$  可以合并为  $[1;3]$ ， $[1;3]$  和  $[2;4]$  可以合并为  $[1;4]$ ，但是  $[1;2]$  和  $[3;4]$  不可以合并。

我们的任务是判断这些区间是否可以最终合并为一个闭区间，如果可以，将这个闭区间输出，否则输出 `no`。

##### 【输入】

第一行为一个整数  $n$ ， $3 \leq n \leq 50000$ 。表示输入区间的数量。

之后  $n$  行，在第  $i$  行上 ( $1 \leq i \leq n$ )，为两个整数  $a_i$  和  $b_i$ ，整数之间用一个空格分隔，表示区间  $[a_i; b_i]$ （其中  $1 \leq a_i \leq b_i \leq 10000$ ）。

##### 【输出】

输出一行，如果这些区间最终可以合并为一个闭区间，输出这个闭区间的左右边界，用单个空格隔开；否则输出 `no`。

##### 【输入样例】

```

5
5 6
1 5
10 10
6 9
8 10

```

##### 【输出样例】

```

1 10

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>

```

```

#include<algorithm>
#include<string>
#define INF 999999999
#define N 50001
#define MOD 1000000007
using namespace std;
struct node{
    int left,right;
}a[N];
int cmp(struct node a,struct node b)
{
    return a.left<b.left||a.left==b.left&& a.right<b.right;
}
int main()
{
    int n;
    int i;
    int max=-INF;

    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i].left>>a[i].right;
    sort(a+1,a+1+n,cmp);

    for(i=1;i<n;i++)
    {
        if(a[i].right>max)
            max=a[i].right;
        if(a[i+1].left>max)
        {
            cout<<"no"<<endl;
            return 0;
        }
    }
    if(a[i].right>max)
        max=a[i].right;
    cout<<a[1].left<<" "<<max;
    return 0;
}

```

## 1.1.2.7.8 求排列的逆序数（信息学奥赛一本通-T1237）

### 求排列的逆序数（信息学奥赛一本通-T1237）

原创 Alex\_McAvoy 发布于 2018-05-17 21:50:25 阅读数 1285 收藏  
展开

#### 【题目描述】

在 Internet 上的搜索引擎经常需要对信息进行比较，比如可以通过某个人对一些事物的排名来估计他（或她）对各种不同信息的兴趣，从而实现个性化的服务。

对于不同的排名结果可以用逆序来评价它们之间的差异。考虑  $1, 2, \dots, n$  的排列  $i_1, i_2, \dots, i_n$ ，如果其中存在  $j, k$ ，满足  $j < k$  且  $i_j > i_k$ ，那么就称  $(i_j, i_k)$  是这个排列的一个逆序。

一个排列含有逆序的个数称为这个排列的逆序数。例如排列 **263451** 含有 8 个逆序  $(2, 1), (6, 3), (6, 4), (6, 5), (6, 1), (3, 1), (4, 1), (5, 1)$ ，因此该排列的逆序数就是 8。显然，由  $1, 2, \dots, n$  构成的所有  $n!$  个排列中，最小的逆序数是 0，对应的排列就是  $1, 2, \dots, n$ ；最大的逆序数是  $n(n-1)/2$ ，对应的排列就是  $n, (n-1), \dots, 2, 1$ 。逆序数越大的排列与原始排列的差异度就越大。

现给定  $1, 2, \dots, n$  的一个排列，求它的逆序数。

#### 【输入】

第一行是一个整数  $n$ ，表示该排列有  $n$  个数 ( $n \leq 100000$ )。

第二行是  $n$  个不同的正整数，之间以空格隔开，表示该排列。

#### 【输出】

输出该排列的逆序数。

#### 【输入样例】

```
6
2 6 3 4 5 1
```

#### 【输出样例】

```
8
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
```

```

#include<cstring>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
using namespace std;
int n;
int a[N],b[N];
long long ans=0;
void msort(int left,int right)
{
    int mid=(left+right)/2;
    if(left>=right)
        return ;
    msort(left,mid);
    msort(mid+1,right);

    int i=left,j=mid+1,n=mid,m=right,k=0;
    while(i<=n&&j<=m)
        if(a[i]>a[j])
        {
            ans+=n-i+1;
            b[k++]=a[j++];
        }
        else
            b[k++]=a[i++];
    while(i<=n)
        b[k++]=a[i++];
    while(j<=m)
        b[k++]=a[j++];
    for(i=0;i<k;i++)
        a[left+i]=b[i];
}
int main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    msort(1,n);
    cout<<ans<<endl;
    return 0;
}

```

## 1.1.2.7.9 一元三次方程求解（信息学奥赛一本通-T1238）

### 一元三次方程求解（信息学奥赛一本通-T1238）

原创 Alex\_McAvoy 发布于 2018-05-17 21:52:30 阅读数 1312 收藏  
展开

#### 【题目描述】

形如： $ax^3+bx^2+cx+d=0$  这样的 一个一元三次方程。

给出该方程中各项的系数(a, b, c, d 均为实数)，并约定该方程存在三个不同实根(根的范围在 $-100$  至  $100$  之间)，且根与根之差的绝对值 $\geq 1$ 。要求由小到大依次在同一行输出这三个实根(根与根之间留有空格)，并精确到小数点后 2 位。

#### 【输入】

一行，包含四个实数 a, b, c, d，相邻两个数之间用单个空格隔开。

#### 【输出】

一行，包含三个实数，为该方程的三个实根，按从小到大顺序排列，相邻两个数之间用单个空格隔开，精确到小数点后 2 位。

#### 【输入样例】

1.0 -5.0 -4.0 20.0

#### 【输出样例】

-2.00 2.00 5.00

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 50001
#define MOD 1000000007
#define E 1e-5
using namespace std;
double a,b,c,d;
```

```

double calculate(double x)
{
    double ans=0;
    double y=1;

    for(int i=1;i<=3;i++)
        y*=x;
    y*=a;
    ans+=y;

    y=1;
    for(int i=1;i<=2;i++)
        y*=x;
    y*=b;
    ans+=y;

    y=1;
    y*=x;
    y*=c;
    ans+=y;

    ans+=d;
    return ans;
}
int main()
{
    double x=-100,x1,x2,x3;

    cin>>a>>b>>c>>d;
    while(x-100<=E)
    {
        if(fabs(calculate(x))<=E)
            break;
        x+=0.01;
    }
    x1=x;

    x+=0.01;
    while(x-100<=E)
    {
        if(fabs(calculate(x))<=E)
            break;
        x+=0.01;
    }
}

```

```

x2=x;

x+=0.01;
while(x-100<=E)
{
    if(fabs(calculate(x))<E)
        break;
    x+=0.01;
}
x3=x;
printf("%.2lf %.2lf %.2lf\n",x1,x2,x3);
return 0;
}

```

### 1.1.2.7.10 统计数字（信息学奥赛一本通-T1239）

#### 统计数字（信息学奥赛一本通-T1239）

原创 Alex\_McAvoy 发布于 2018-05-17 21:47:22 阅读数 1250 收藏  
展开

##### 【题目描述】

某次科研调查时得到了  $n$  个自然数，每个数均不超过  $1500000000$  ( $1.5 \times 10^9$ )。已知不相同的数不超过  $10000$  个，现在需要统计这些自然数各自出现的次数，并按照自然数从小到大的顺序输出统计结果。

##### 【输入】

第一行是整数  $n$ ，表示自然数的个数；

第  $2 \sim n+1$  每行一个自然数。

##### 【输出】

包含  $m$  行 ( $m$  为  $n$  个自然数中不相同数的个数)，按照自然数从小到大的顺序输出。每行输出两个整数，分别是自然数和该数出现的次数，其间用一个空格隔开。

##### 【输入样例】

```

8
2
4
2
4
5
100

```



2  
100

【输出样例】

2 3  
4 2  
5 1  
100 2

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-5
using namespace std;
int a[N];
void quicksort(int left,int right)
{
    inti=left,j=right,mid=a[(left+right)/2];
    while(i<=j)
    {
        while(a[i]<mid)
            i++;
        while(a[j]>mid)
            j--;
        if(i<=j)
        {
            swap(a[i],a[j]);
            i++;
            j--;
        }
    }
    if(i<right)
        quicksort(i,right);
    if(left<j)
        quicksort(left,j);
}
int main()
{
```

```

int n;
int cnt=1;
cin>>n;
for(int i=1;i<=n;i++)
    cin>>a[i];
quicksort(1,n);
while(cnt<=n)
{
    ints=a[cnt],k=cnt;
    while(s==a[cnt])
        cnt++;
    cout<<a[cnt-1]<<" "<<cnt-k<<endl;
}
return 0;
}

```

点赞

### 1.1.2.7.11 查找最接近的元素（信息学奥赛一本通-T1240）

#### 查找最接近的元素（信息学奥赛一本通-T1240）

原创 Alex\_McAvoy 发布于 2018-05-17 21:53:55 阅读数 1480 收藏  
展开

##### 【题目描述】

在一个非降序列中，查找与给定值最接近的元素。

##### 【输入】

第一行包含一个整数  $n$ ，为非降序列长度。 $1 \leq n \leq 100000$ 。

第二行包含  $n$  个整数，为非降序列各元素。所有元素的大小均在  $0-1,000,000,000$  之间。

第三行包含一个整数  $m$ ，为要询问的给定值个数。 $1 \leq m \leq 10000$ 。

接下来  $m$  行，每行一个整数，为要询问最接近元素的给定值。所有给定值的大小均在  $0-1,000,000,000$  之间。

##### 【输出】

$m$  行，每行一个整数，为最接近相应给定值的元素值，保持输入顺序。若有多个值满足条件，输出最小的一个。

**【输入样例】**

```
3
2 5 8
2
10
5
```

**【输出样例】**

```
8
5
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-5
using namespace std;
int a[N];
int main()
{
    int n,m;
    int x;
    int left,right,mid;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    cin>>m;
    while(m--)
    {
        cin>>x;
        left=1;
        right=n;

        while(left<right-1)
        {
            mid=(left+right)/2;
            if(a[mid]>x)
                right=mid;
```

```

        else
            left=mid;
        }
        if(fabs(a[left]-x)<=fabs(a[right]-x))
            cout<<a[left]<<endl;
        else
            cout<<a[right]<<endl;
    }
    return 0;
}

```

### 1.1.2.7.12 二分法求函数的零点（信息学奥赛一本通-T1241）

#### 二分法求函数的零点（信息学奥赛一本通-T1241）

原创 Alex\_McAvoy 发布于 2018-05-18 15:53:24 阅读数 1580 收藏  
展开

##### 【题目描述】

有函数： $f(x)=x^5+15x^4+85x^3+225x^2+274x+121$

已知  $f(1.5)>0$ ,  $f(2.4)<0$  且方程  $f(x)=0$  在区间  $[1.5, 2.4]$  有且只有一个根，请用二分法求出该根。

##### 【输入】

(无)

##### 【输出】

该方程在区间  $[1.5, 2.4]$  中的根。要求四舍五入到小数点后 6 位。

##### 【输入样例】

(无)

##### 【输出样例】

(无)

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>

```

```

#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-7
using namespace std;
double calculate(double x)
{
    return x*x*x*x*x-15*x*x*x*x+85*x*x*x-225*x*x+274*x-121;
}
double myabs(double x){
    if(x<0)return -x;
    return x;
}
int main()
{
    double left=1.5,right=2.4;
    while(left+E<right)
    {
        double mid=(left+right)/2.0;
        if(calculate(mid)>0)
            left=mid;
        else right=mid;
    }
    if(calculate(left)==0)
        printf("%.6lf\n",left);
    else
        printf("%.6lf\n",left);
    return 0;
}

```

### 1.1.2.7.13 网线主管（信息学奥赛一本通-T1242）

#### 网线主管（信息学奥赛一本通-T1242）

原创 Alex\_McAvoy 发布于 2018-05-18 16:06:32 阅读数 1025 收藏  
展开

#### 【题目描述】

仙境的居民们决定举办一场程序设计区域赛。裁判委员会完全由自愿组成，他们承诺

要组织一次史上最公正的比赛。他们决定将选手的电脑用星形拓扑结构连接在一起，即将它们全部连到一个单一的中心服务器。为了组织这个完全公正的比赛，裁判委员会主席提出要将所有选手的电脑等距离地围绕在服务器周围放置。

为购买网线，裁判委员会联系了当地的一个网络解决方案提供商，要求能够提供一定数量的等长网线。裁判委员会希望网线越长越好，这样选手们之间的距离可以尽可能远一些。

该公司的网线主管承接了这个任务。他知道库存中每条网线的长度（精确到厘米），并且只要告诉他所需的网线长度（精确到厘米），他都能够完成对网线的切割工作。但是，这次，所需的网线长度并不知道，这让网线主管不知所措。

你需要编写一个程序，帮助网线主管确定一个最长的网线长度，并且按此长度对库存中的网线进行切割，能够得到指定数量的网线。

#### 【输入】

第一行包含两个整数  $N$  和  $K$ ，以单个空格隔开。 $N$  ( $1 \leq N \leq 10000$ ) 是库存中的网线数， $K$  ( $1 \leq K \leq 10000$ ) 是需要的网线数量。

接下来  $N$  行，每行一个数，为库存中每条网线的长度（单位：米）。所有网线的长度至少 1m，至多 100km。输入中的所有长度都精确到厘米，即保留到小数点后两位。

#### 【输出】

网线主管能够从库存的网线中切出指定数量的网线的最长长度（单位：米）。必须精确到厘米，即保留到小数点后两位。

若无法得到长度至少为 1cm 的指定数量的网线，则必须输出“0.00”（不包含引号）。

#### 【输入样例】

```
4 11
8.02
7.43
4.57
5.39
```

#### 【输出样例】

```
2.00
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
```

```

#define N 100001
#define MOD 1000000007
#define E 1e-3
using namespace std;
int n,k,a[N];
int judge(int x)
{
    int i,ans=0;
    for(i=1;i<=n;i++)
        ans+=a[i]/x;
    return ans>=k;
}
int main()
{
    double b;
    int i,j;
    int left=0,right=0,mid;

    cin>>n>>k;
    for(i=1;i<=n;i++)
    {
        cin>>b;
        a[i]=(int)(b*100+0.5);
        if(right<a[i])
            right=a[i];
    }

    right+=1;
    while(left+1<right)
    {
        mid=(left+right)/2;
        if(judge(mid))
            left=mid;
        else
            right=mid;
    }
    printf("%.2lf",left/100.0);
    return 0;
}

```

### 1.1.2.7.14 月度开销（信息学奥赛一本通-T1243）

#### 月度开销（信息学奥赛一本通-T1243）

原创 Alex\_McAvoy 发布于 2018-05-18 17:03:35 阅读数 1079 收藏  
展开

### 【题目描述】

农夫约翰是一个精明的会计师。他意识到自己可能没有足够的钱来维持农场的运转了。他计算出并记录下了接下来  $N$  ( $1 \leq N \leq 100,000$ ) 天里每天需要的开销。

约翰打算为连续的  $M$  ( $1 \leq M \leq N$ ) 个财政周期创建预算案，他把一个财政周期命名为 **fajo** 月。每个 **fajo** 月包含一天或连续的多天，每天被恰好包含在一个 **fajo** 月里。

约翰的目标是合理安排每个 **fajo** 月包含的天数，使得开销最多的 **fajo** 月的开销尽可能少。

### 【输入】

第一行包含两个整数  $N, M$ ，用单个空格隔开。

接下来  $N$  行，每行包含一个 1 到 10000 之间的整数，按顺序给出接下来  $N$  天里每天的开销。

### 【输出】

每行输出两个整数，分别是自然数和该数出现的次数，其间用一个空格隔开。一个整数，即最大月度开销的最小值。

### 【输入样例】

```
7 5
100
400
300
100
500
101
400
```

### 【输出样例】

```
500
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
```



```

#define MOD 1000000007
#define E 1e-3
using namespace std;
int n,m,a[N];
int judge(int x)
{
    int money=0,month=0,d,i;
    for(i=1;i<=n;i++)
    {
        money+=a[i];
        if(money>=x)
        {
            month++;
            if(a[i]<x)
                money=a[i];
            else
                return 1;
        }
    }
    return month>=m;
}
int main()
{
    int left,right,mid;
    int tot=0;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
        tot+=a[i];
    }
    left=1;
    right=tot;
    while(left+1<right)
    {
        mid=(left+right)/2;
        if(judge(mid))
            left=mid;
        else
            right=mid;
    }
    if(judge(left))
        cout<<left<<endl;
    else

```

```

    cout<<right<<endl;
    return 0;
}

```

### 1.1.2.7.15 和为给定数（信息学奥赛一本通-T1244）

#### 和为给定数（信息学奥赛一本通-T1244）

原创 Alex\_McAvoy 发布于 2018-05-18 17:09:01 阅读数 1106 收藏  
展开

##### 【题目描述】

给出若干个整数，询问其中是否有一对数的和等于给定的数。

##### 【输入】

第一行是整数  $n$  ( $0 < n \leq 100,000$ )，表示有  $n$  个整数。

第二行是  $n$  个整数。整数的范围是在 0 到 108 之间。

第三行是一个整数  $m$  ( $0 \leq m \leq 230$ )，表示需要得到的和。

##### 【输出】

若存在和为  $m$  的数对，输出两个整数，小的在前，大的在后，中间用单个空格隔开。若有多个数对满足条件，选择数对中较小的数更小的。若找不到符合要求的数对，输出一行 No。

##### 【输入样例】

```

4
2 5 1 4
6

```

##### 【输出样例】

```

1 5

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>

```

```

#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-3
using namespace std;
int a[N];
int main()
{
    long long n,m,i,j;
    int tot=0;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    cin>>m;
    sort(a+1,a+1+n);
    i=1;j=n;
    while(i<j)
    {
        if(a[i]+a[j]==m)
        {
            cout<<a[i]<<" "<<a[j]<<endl;
            exit(0);
        }
        else if(a[i]+a[j]<m)
            i++;
        else
            j--;
    }
    cout<<"No"<<endl;
    return 0;
}

```

### 1.1.2.7.16 不重复地输出数（信息学奥赛一本通-T1245）

#### 不重复地输出数（信息学奥赛一本通-T1245）

原创 Alex\_McAvoy 发布于 2018-05-18 17:12:14 阅读数 878 收藏  
展开

**【题目描述】**

输入  $n$  个数，从小到大将它们输出，重复的数只输出一次。保证不同的数不超过 500 个。

**【输入】**

第一行是一个整数  $n$ 。  $1 \leq n \leq 100000$ 。

之后  $n$  行，每行一个整数。整数大小在 `int` 范围内。

**【输出】**

一行，从小到大不重复地输出这些数，相邻两个数之间用单个空格隔开。

**【输入样例】**

```
5
2 4 4 5 1
```

**【输出样例】**

```
1 2 4 5
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-3
using namespace std;
int a[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    sort(a+1,a+1+n);
    cout<<a[1];
    for(int i=2;i<=n;i++)
        if(a[i]!=a[i-1])
            cout<<" "<<a[i];
    cout<<endl;
    return 0;
}
```

}

点赞

## 1.1.2.7.17 膨胀的木棍（信息学奥赛一本通-T1246）

### 膨胀的木棍（信息学奥赛一本通-T1246）

原创 Alex\_McAvoy 发布于 2018-05-18 17:17:48 阅读数 855 收藏  
展开

#### 【题目描述】

当长度为  $L$  的一根细木棍的温度升高  $n$  度，它会膨胀到新的长度  $L'=(1+n*C)*L$ ，其中  $C$  是热膨胀系数。

当一根细木棍被嵌在两堵墙之间被加热，它将膨胀形成弓形的弧，而这个弓形的弦恰好是未加热前木棍的原始位置。

你的任务是计算木棍中心的偏移距离。

#### 【输入】

三个非负实数：木棍初始长度（单位：毫米），温度变化（单位：度），以及材料的热膨胀系数。

保证木棍不会膨胀到超过原始长度的 1.5 倍。

#### 【输出】

木棍中心的偏移距离（单位：毫米），保留到小数点后第三位。

#### 【输入样例】

1000 100 0.0001

#### 【输出样例】

61.329

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
```

```

#define MOD 1000000007
#define E 1e-12
using namespace std;
int main()
{
    double l,n,c;
    cin>>l>>n>>c;
    double left=0,right=acos(-1.0),mid;
    double l2=(1+n*c)*l;
    while(right-left>E)
    {
        mid=(left+right)/2.0;
        if(2*l2/l>mid/sin(mid/2.0))
            left=mid;
        else
            right=mid;
    }
    printf("%.3lf\n",l2/mid*(1-cos(mid/2)));
    return 0;
}

```

### 1.1.2.7.18 河中跳房子（信息学奥赛一本通-T1247）

#### 河中跳房子（信息学奥赛一本通-T1247）

原创 Alex\_McAvoy 发布于 2018-05-18 17:24:24 阅读数 934 收藏  
展开

##### 【题目描述】

每年奶牛们都要举办各种特殊版本的跳房子比赛，包括在河里从一个岩石跳到另一个岩石。这项激动人心的活动在一条长长的笔直河道中进行，在起点和离起点  $L$  远 ( $1 \leq L \leq 1,000,000,000$ ) 的终点处均有一个岩石。在起点和终点之间，有  $N$  ( $0 \leq N \leq 50,000$ ) 个岩石，每个岩石与起点的距离分别为  $D_i$  ( $0 < D_i < L$ )。

在比赛过程中，奶牛轮流从起点出发，尝试到达终点，每一步只能从一个岩石跳到另一个岩石。当然，实力不济的奶牛是没有办法完成目标的。

农夫约翰为他的奶牛们感到自豪并且年年都观看了这项比赛。但随着时间的推移，看着其他农夫的胆小奶牛们在相距很近的岩石之间缓慢前行，他感到非常厌烦。他计划移走一些岩石，使得从起点到终点的过程中，最短的跳跃距离最长。他可以移走除起点和终点外的至多  $M$  ( $0 \leq M \leq N$ ) 个岩石。

请帮助约翰确定移走这些岩石后，最长可能的最短跳跃距离是多少？

**【输入】**

第一行包含三个整数  $L, N, M$ ，相邻两个整数之间用单个空格隔开。

接下来  $N$  行，每行一个整数，表示每个岩石与起点的距离。岩石按与起点距离从近到远给出，且不会有两个岩石出现在同一个位置。

**【输出】**

一个整数，最长可能的最短跳跃距离。

**【输入样例】**

```
25 5 2
2
11
14
17
21
```

**【输出样例】**

```
4
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#define INF 999999999
#define N 1000001
#define MOD 1000000007
#define E 1e-12
using namespace std;
int l,n,m;
int a[N];
int judge(int x)
{
    int i,dis=0,num=0;
    for(int i=1;i<=n;i++)
        if(a[i]-dis<x)
            num++;
    else
        dis=a[i];
    if(l-dis<x)
        num++;
    return num<=m;
}
```

```

}
int main()
{
    int left,right,mid;
    cin>>l>>n>>m;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    left=0;
    right=l;
    while(left+1<right)
    {
        mid=(left+right)/2;
        if(judge(mid))
            left=mid;
        else
            right=mid;
    }
    cout<<left<<endl;
    return 0;
}

```

点赞

## 1.1.2.8 第八章 广度优先搜索（BFS）

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第二部分 基础算法](#) > 第八章 广度优先搜索（BFS）

### 1.1.2.8.1 细胞（信息学奥赛一本通-T1329）

#### 细胞（信息学奥赛一本通-T1329）

原创 Alex\_McAvoy 发布于 2018-05-26 12:36:29 阅读数 1780 收藏  
展开

##### 【题目描述】

一矩形阵列由数字 00 到 99 组成,数字 11 到 99 代表细胞,细胞的定义为沿细胞数字上下左右还是细胞数字则为同一细胞,求给定矩形阵列的细胞个数。如下阵列,有 44 个细胞:



```

4 10
0234500067
1034560500
2045600671
0000000089

```

**【输入】**

第一行为矩阵的行  $n$  和列  $m$ ;

下面为一个  $n \times m$  的矩阵。

**【输出】**

细胞个数。

**【输入样例】**

```

4 10
0234500067
1034560500
2045600671
0000000089

```

**【输出样例】**

```

4

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
int dir[4][2]={{1,0},{-1,0},{0,1},{0,-1}};
using namespace std;
int n,m;
char g[N][N];
struct node{
    int x;
    int y;

```

```

}q[N*1000];
void bfs(int x0,int y0)
{
    inthead=1,tail=1;

    q[tail].x=x0;
    q[tail].y=y0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        g[x][y]='0';
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];<
BR>        if(0<=nx&&nx<n&&0<=ny&&ny<m&&g[nx][ny]!='0')
        {
            q[tail].x=nx;
            q[tail].y=ny;
            tail++;
        }
        head++;
    }
}
int main()
{
    int cnt=0;
    scanf("%d%d",&n,&m);
    for(int i=0;i<n;i++)
        scanf("%s",g[i]);
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            if(g[i][j]!='0')
            {
                cnt++;
                bfs(i,j);
            }
    printf("%d\n",cnt);
    return 0;
}

```

## 1.1.2.8.2 最少步数（信息学奥赛一本通-T1330）

### 最少步数（信息学奥赛一本通-T1330）

原创 Alex\_McAvoy 发布于 2018-05-26 12:34:37 阅读数 1299 收藏  
展开

#### 【题目描述】

在各种棋中，棋子的走法总是一定的，如中国象棋中马走“日”。有一位小学生就想如果马能有两种走法将增加其趣味性，因此，他规定马既能按“日”走，也能如象一样走“田”字。他的同桌平时喜欢下围棋，知道这件事后觉得很有趣，就想试一试，在一个（100×100）的围棋盘上任选两点 A、B，A 点放上黑子，B 点放上白子，代表两匹马。棋子可以按“日”字走，也可以按“田”字走，俩人一个走黑马，一个走白马。谁用最少的步数走到左上角坐标为(1,1)的点时，谁获胜。现在他请你帮忙，给你 A、B 两点的坐标，想知道两个位置到（1,1）点可能的最少步数。

#### 【输入】

A、B 两点的坐标。

#### 【输出】

最少步数。

#### 【输入样例】

```
12 16
18 10
```

#### 【输出样例】

```
8
9
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
```

```

#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N];
bool vis[N][N];
int
dir[][2]={{-2,1},{-2,-1},{-2,2},{-2,-2},{-1,-2},{-1,2},{1,-2},{1,2},{2,-1},{2,1},{2,-2},{2,2}};
struct node
{
    int x;
    int y;
    int step;
}q[N*100];
void bfs(int x0,int y0)
{
    int head=1,tail=1;
    memset(vis,0,sizeof(vis));

    q[tail].x=x0;
    q[tail].y=y0;
    q[tail].step=0;
    tail++;
    vis[x0][y0]=1;
    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int step=q[head].step;
        if(x==1&&y==1)
        {
            printf("%d\n",step);
            break;
        }
        for(int i=0;i<12;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(nx>=1&&nx<=100&&ny>=1&&ny<=100&&vis[nx][ny]==0)
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].step=step+1;
                tail++;
            }
        }
    }
}

```

```

    }
  }
  head++;
}
}
int main()
{
  int xa,ya,xb,yb;
  scanf("%d%d%d%d",&xa,&ya,&xb,&yb);
  bfs(xa,ya);
  bfs(xb,yb);
  return 0;
}

```

### 1.1.2.8.3 Dungeon Master（信息学奥赛一本通-T1248）

#### Dungeon Master（信息学奥赛一本通-T1248）

原创 Alex\_McAvoy 发布于 2018-05-26 12:33:09 阅读数 874 收藏  
展开

##### 【题目描述】

这题是一个三维的迷宫题目，其中用‘.’表示空地，‘#’表示障碍物，‘S’表示起点，‘E’表示终点，求从起点到终点的最小移动次数，解法和二维的类似，只是在行动时除了东南西北移动外还多了上下。可以上下左右前后移动，每次都只能移到相邻的空位，每次需要花费一分钟，求从起点到终点最少要多久。

##### 【输入】

多组测试数据。

一组测试数据表示一个三维迷宫：

前三个数，分别表示层数、一个面的长和宽，后面是每层的平面图。前三个数据为三个零表示结束。

##### 【输出】

最小移动次数。

##### 【输入样例】

```

3 4 5
S....
.###.
.##..

```

```
###.#
#####
#####
##.##
##...
#####
#####
#.###
#####E
1 3 3
S##
#E#
###
0 0 0
```

【输出样例】

Escaped in 11 minute(s).  
Trapped!

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int X,Y,Z;
char a[N][N][N];
bool vis[N][N][N];
int dir[6][3]={{1,0,0},{-1,0,0},{0,1,0},{0,-1,0},{0,0,1},{0,0,-1}};
struct node
{
    int x;
    int y;
    int z;
    int step;
```

```

}q[N*10000];
void bfs(int sx,int sy,int sz,int ex,int ey,int ez)
{
    inthead=1,tail=1;
    bool flag=true;
    memset(vis,0,sizeof(vis));

    vis[sx][sy][sz]=1;
    q[tail].x=sx;
    q[tail].y=sy;
    q[tail].z=sz;
    q[tail].step=0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int z=q[head].z;
        int step=q[head].step;    if(x==ex&&y==ey&&z==ez)
        {
            flag=false;
            printf("Escaped in %d minute(s).\n",step);
            break;
        }
        for(int i=0;i<6;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            int nz=z+dir[i][2];
            if(0<=nx&&nx<X&&0<=ny&&ny<Y&&0<=nz&&nz<Z&&a[nx][ny][nz]!='.'&&vis[nx][ny][nz]==0)
            {
                vis[nx][ny][nz]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].z=nz;
                q[tail].step=step+1;
                tail++;
            }
        }
        head++;
    }
    if(flag)

```

```

printf("Trapped!\n");
}
int main()
{
    int sx,sy,sz,ex,ey,ez;
    while(scanf("%d%d%d",&X,&Y,&Z)!=EOF&&X&&Y&&Z)
    {
        for(int i=0;i<X;i++)
            for(int j=0;j<Y;j++)
                scanf("%s",a[i][j]);
        for(int i=0;i<X;i++)
            for(int j=0;j<Y;j++)
                for(int k=0;k<Z;k++)
                {
                    if(a[i][j][k]=='S')
                    {
                        sx=i;
                        sy=j;
                        sz=k;
                    }
                    if(a[i][j][k]=='E')
                    {
                        ex=i;
                        ey=j;
                        ez=k;
                        a[i][j][k]='.';
                    }
                }
        bfs(sx,sy,sz,ex,ey,ez);
    }
    return 0;
}

```

#### 1.1.2.8.4 Lake Counting（信息学奥赛一本通-T1249）

##### Lake Counting（信息学奥赛一本通-T1249）

原创 Alex\_McAvoy 发布于 2018-05-26 12:30:46 阅读数 1015 收藏  
展开



**【题目描述】**

有一块  $N \times M$  的土地，雨后积起了水，有水标记为‘W’，干燥为‘.’。八连通的积水被认为是连接在一起的。请求出院子里共有多少水洼？

**【输入】**

第一行为  $N, M (1 \leq N, M \leq 110)$ 。

下面为  $N \times M$  的土地示意图。

**【输出】**

一行，共有的水洼数。

**【输入样例】**

```
10 12
W.....WW.
.WWW....WWW
....WW...WW.
.....WW.
.....W..
..W.....W..
.W.W....WW.
W.W.W....W.
.W.W.....W.
..W.....W.
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<stdio>
int n,m;
char a[120][120];
int dir[8][2]={-1,0},{1,0},{0,-1},{0,1},{-1,1},{1,1},{1,-1},{-1,-1}};
struct node
{
    int x;
    int y;
}q[100000];
void bfs(int x0,int y0)
{
    int head=1,tail=1;

    a[x0][y0]='.';
    q[tail].x=x0;
    q[tail].y=y0;
    tail++;
```

```

while(head<tail)
{
    int x=q[head].x;
    int y=q[head].y;
    for(int i=0;i<8;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=0&&nx<n&&ny>=0&&ny<m&&a[nx][ny]!='W')
        {
            a[nx][ny]='.';
            q[tail].x=nx;
            q[tail].y=ny;
            tail++;
        }
    }
    head++;
}
}
int main()
{
    int sum=0;
    scanf("%d%d",&n,&m);
    for(int i=0;i<n;i++)
        scanf("%s",a[i]);
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            if(a[i][j]=='W')
            {
                sum++;
                bfs(i,j);
            }
    printf("%d\n",sum);
    return 0;
}

```

### 1.1.2.8.5 The Castle（信息学奥赛一本通-T1250）

#### The Castle（信息学奥赛一本通-T1250）

原创 Alex\_McAvoy 发布于 2018-05-26 12:27:54 阅读数 1026 收藏  
展开

### 【题目描述】

一座城堡被分成  $m \times n$  个方块 ( $m \leq 50, n \leq 50$ ), 每个方块可有  $0 \sim 4$  堵墙 (0 表示无墙)。下面示出了建筑平面图:

图中的加粗黑线代表墙。几个连通的方块组成房间, 房间与房间之间一定是用黑线(墙)隔开的。

现在要求你编一个程序, 解决以下 2 个问题:

1、该城堡中有多少个房间?

2、最大的房间有多大?

### 【输入】

平面图用一个数字表示一个方块(第 1 个房间用二进制 1011 表示, 0 表示无东墙, 用十进制 11 表示)。

第一行一个整数  $m$  ( $m \leq 50$ ), 表示房子南北方向的长度。

第二行一个整数  $n$  ( $n \leq 50$ ), 表示房子东西方向的长度。

后面的  $m$  行, 每行有  $n$  个整数, 每个整数都表示平面图对应位置的方块的特征。每个方块中墙的特征由数字  $P$  来描述 ( $0 \leq P \leq 15$ )。数字  $P$  是下面的可能取的数字之和:

1 (西墙 west)

2 (北墙 north)

4 (东墙 east)

8 (南墙 south)

室内的墙被定义两次: 例如方块 (1, 1) 中的南墙也被位于其南面的方块 (2, 1) 定义了一次。

建筑中至少有两个房间。

### 【输出】

第 1 行: 一个整数, 表示房间总数;

第 2 行: 一个整数, 表示最大房间的面积 (方块数)。

## 【输入样例】

```

4
7
11 6 11 6 3 10 6
7 9 6 13 5 15 5
1 10 12 7 13 7 5
13 11 10 8 10 12 13

```

## 【输出样例】

```

5
9

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 51
#define MOD 2520
#define E 1e-12
using namespace std;
int n,m;
int a[N][N];
int b[4]={1,2,4,8};
bool vis[N][N];
int sum,maxx;
int dir[4][2]={0,-1},{-1,0},{0,1},{1,0}};
struct node
{
    int x;
    int y;
}q[N*100];
void bfs(int x0,int y0)
{
    int head=1,tail=1;
    int cnt=1;

    vis[x0][y0]=1;

```

```

q[tail].x=x0;
q[tail].y=y0;
tail++;

while(head<tail)
{
    int x=q[head].x;
    int y=q[head].y;
    for(int i=0;i<4;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&vis[nx][ny]==0&&(a[x][y]&b[i])==0)
        {
            cnt++;
            vis[nx][ny]=1;
            q[tail].x=nx;
            q[tail].y=ny;
            tail++;
        }
    }
    head++;
}
if(cnt>maxx)
    maxx=cnt;
sum++;
}
int main()
{
    memset(vis,0,sizeof(vis));
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            scanf("%d",&a[i][j]);

    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            if(vis[i][j]==0)
                bfs(i,j);
    printf("%d\n%d\n",sum,maxx);
    return 0;
}

```

## 1.1.2.8.6 仙岛求药（信息学奥赛一本通-T1251）

### 仙岛求药（信息学奥赛一本通-T1251）

原创 Alex\_McAvoy 发布于 2018-05-26 12:24:55 阅读数 1483 收藏  
展开

#### 【题目描述】

少年李逍遥的婶婶病了，王小虎介绍他去一趟仙灵岛，向仙女姐姐要仙丹救婶婶。叛逆但孝顺的李逍遥闯进了仙灵岛，克服了千险万难来到岛的中心，发现仙药摆在了迷阵的深处。迷阵由  $M \times N$  个方格组成，有的方格内有可以瞬秒李逍遥的怪物，而有的方格内则是安全。现在李逍遥想尽快找到仙药，显然他应避开有怪物的方格，并经过最少的方格，而且那里会有神秘人物等待着他。现在要求你来帮助他实现这个目标。

下图 显示了一个迷阵的样例及李逍遥找到仙药的路线。

#### 【输入】

输入有多组测试数据。每组测试数据以两个非零整数  $M$  和  $N$  开始，两者均不大于 20。 $M$  表示迷阵行数， $N$  表示迷阵列数。接下来有  $M$  行，每行包含  $N$  个字符，不同字符分别代表不同含义：

- 1) '@'：少年李逍遥所在的位置；
- 2) '.'：可以安全通行的方格；
- 3) '#'：有怪物的方格；
- 4) '\*'：仙药所在位置。

当在一行中读入的是两个零时，表示输入结束。

#### 【输出】

对于每组测试数据，分别输出一行，该行包含李逍遥找到仙药需要穿过的最少的方格数目(计数包括初始位置的方块)。如果他不可能找到仙药，则输出 -1。

#### 【输入样例】

```
8 8
.@##...#
#...#.#
#.#.##..
..#.###.
#.#...#.
```

```

..###.#.
...#.*..
.#...###
6 5
.*.#.
.#...
..##.
.....
.#...
....@
9 6

```

```

.#..#.
.#.*.#
.####.
..#...
..#...
..#...
..#...
#.@.##
.#..#.
0 0

```

#### 【输出样例】

```

10
8
-1

```

#### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 26
#define MOD 2520
#define E 1e-12

```

```

using namespace std;
int r,c;
char a[N][N];
bool vis[N][N];
int dir[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
struct node
{
    int x;
    int y;
    int step;
}q[N*100];
void bfs(int sx,int sy,int ex,int ey)
{
    inthead=1,tail=1;
    bool flag=true;
    memset(vis,0,sizeof(vis));

    vis[sx][sy]=1;
    q[tail].x=sx;
    q[tail].y=sy;
    q[tail].step=0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int step=q[head].step;
        if(x==ex&&y==ey)
        {
            flag=false;
            printf("%d\n",step);
            break;
        }
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(nx>=0&&nx<r&&ny>=0&&ny<c&&vis[nx][ny]==0&&a[nx][ny]!='.')
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].step=step+1;
            }
        }
    }
}

```



```

        tail++;
    }
}
head++;
}
if(flag)
    printf("-1\n");
}
int main()
{
    int sx,sy,ex,ey;
    while(scanf("%d%d",&r,&c)!=EOF&&(r||c))
    {

        for(int i=0;i<r;i++)
            scanf("%s",a[i]);
        for(int i=0;i<r;i++)
            for(int j=0;j<c;j++)
            {
                if(a[i][j]=='@')
                {
                    sx=i;
                    sy=j;
                }
                if(a[i][j]=='*')
                {
                    ex=i;
                    ey=j;
                    a[i][j]='.';
                }
            }
        bfs(sx,sy,ex,ey);
    }
    return 0;
}

```

### 1.1.2.8.7 走迷宫（信息学奥赛一本通-T1252）

#### 走迷宫（信息学奥赛一本通-T1252）

原创 Alex\_McAvoy 发布于 2018-05-28 20:39:55 阅读数 1479 收藏  
展开

**【题目描述】**

一个迷宫由  $R$  行  $C$  列格子组成，有的格子里有障碍物，不能走；有的格子是空地，可以走。

给定一个迷宫，求从左上角走到右下角最少需要走多少步(数据保证一定能走到)。只能在水平方向或垂直方向走，不能斜着走。

**【输入】**

第一行是两个整数， $R$  和  $C$ ，代表迷宫的长和宽。(  $1 \leq R, C \leq 40$  )

接下来是  $R$  行，每行  $C$  个字符，代表整个迷宫。

空地格子用 '.' 表示，有障碍物的格子用 '#' 表示。

迷宫左上角和右下角都是 '.'。

**【输出】**

输出从左上角走到右下角至少要经过多少步（即至少要经过多少个空地格子）。计算步数要包括起点和终点。

**【输入样例】**

```
5 5
..###
#....
###.
###.
###.
#.#.
```

**【输出样例】**

```
9
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
```

```

int r,c;
char a[N][N];
bool vis[N][N];
int dir[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
struct node
{
    int x;
    int y;
    int step;
}q[N*100];
void bfs(int sx,int sy,int ex,int ey)
{
    int head=1,tail=1;
    memset(vis,0,sizeof(vis));

    vis[sx][sy]=1;
    q[tail].x=sx;
    q[tail].y=sy;
    q[tail].step=1;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int step=q[head].step;
        if(x==ex&&y==ey)
        {
            printf("%d\n",step);
            break;
        }
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(nx>=0&&nx<r&&ny>=0&&ny<c&&vis[nx][ny]==0&&a[nx][ny]!='.')
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].step=step+1;
                tail++;
            }
        }
    }
}

```

```

        head++;
    }
}
int main()
{
    scanf("%d%d",&r,&c);
    for(int i=0;i<r;i++)
        scanf("%s",a[i]);
    bfs(0,0,r-1,c-1);
    return 0;
}

```

### 1.1.2.8.8 迷宫问题（信息学奥赛一本通-T1255）

#### 迷宫问题（信息学奥赛一本通-T1255）

原创 Alex\_McAvoy 发布于 2018-05-28 20:36:55 阅读数 1607 收藏  
更新于 2019-07-23 10:35:30 展开

##### 【题目描述】

定义一个二维数组：

```

int maze[5][5] = {

0,1,0,0,0,

0,1,0,1,0,

0,0,0,0,0,

0,1,1,1,0,

0,0,0,1,0,

};

```

它表示一个迷宫，其中的 1 表示墙壁，0 表示可以走的路，只能横着走或竖着走，不能斜着走，要求编程找出从左上角到右下角的最短路线。

##### 【输入】

一个  $5 \times 5$  的二维数组，表示一个迷宫。数据保证有唯一解。

##### 【输出】

左上角到右下角的最短路径，格式如样例所示。

**【输入样例】**

```

0 1 0 0 0
0 1 0 1 0
0 0 0 0 0
0 1 1 1 0
0 0 0 1 0

```

**【输出样例】**

```

(0, 0)
(1, 0)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(2, 4)
(3, 4)
(4, 4)

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 26
#define MOD 2520
#define E 1e-12
using namespace std;
int n=5;
char a[N][N];
bool vis[N][N];
int dir[4][2]={-1,0},{1,0},{0,-1},{0,1}};
struct node
{
    int x;
    int y;
}q[N*100],res[N][N];
void print(int x,int y)
{

```

```

    if(x== -1 && y== -1)
        return;
    print(res[x][y].x, res[x][y].y);
    printf("(%d, %d)\n", x, y);
}
void bfs(int sx, int sy, int ex, int ey)
{
    int head = 1, tail = 1;
    memset(vis, 0, sizeof(vis));

    vis[sx][sy] = 1;
    res[sx][sy].x = -1;
    res[sx][sy].y = -1;
    q[tail].x = sx;
    q[tail].y = sy;
    tail++;

    while(head < tail)
    {
        int x = q[head].x;
        int y = q[head].y;
        if(x == ex && y == ey)
        {
            print(x, y);
            break;
        }
        for(int i = 0; i < 4; i++)
        {
            int nx = x + dir[i][0];
            int ny = y + dir[i][1];
            if(nx >= 0 && nx < n && ny >= 0 && ny < n && vis[nx][ny] == 0 && a[nx][ny] == 0)
            {
                vis[nx][ny] = 1;
                q[tail].x = nx;
                q[tail].y = ny;
                tail++;
                res[nx][ny].x = x;
                res[nx][ny].y = y;
            }
        }
        head++;
    }
}
int main()

```

```

{
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    bfs(0,0,n-1,n-1);

    return 0;
}

```

点赞

### 1.1.2.8.9 献给阿尔吉侬的花束（信息学奥赛一本通-T1256）

#### 献给阿尔吉侬的花束（信息学奥赛一本通-T1256）

原创 Alex\_McAvoy 发布于 2018-05-28 20:35:08 阅读数 1133 收藏  
展开

##### 【题目描述】

阿尔吉侬是一只聪明又慵懒的小白鼠，它最擅长的就是走各种各样的迷宫。今天它要挑战一个非常大的迷宫，研究员们为了鼓励阿尔吉侬尽快到达终点，就在终点放了一块阿尔吉侬最喜欢的奶酪。现在研究员们想知道，如果阿尔吉侬足够聪明，它最少需要多少时间就能吃到奶酪。

迷宫用一个  $R \times C$  的字符矩阵来表示。字符 **S** 表示阿尔吉侬所在的位置，字符 **E** 表示奶酪所在的位置，字符 **#** 表示墙壁，字符 **.** 表示可以通行。阿尔吉侬在 1 个单位时间内可以从当前的位置走到它上下左右四个方向上的任意一个位置，但不能走出地图边界。

##### 【输入】

第一行是一个正整数  $T$  ( $1 \leq T \leq 10$ )，表示一共有  $T$  组数据。

每一组数据的第一行包含了两个用空格分开的正整数  $R$  和  $C$  ( $2 \leq R, C \leq 200$ )，表示地图是一个  $R \times C$  的矩阵。

接下来的  $R$  行描述了地图的具体内容，每一行包含了  $C$  个字符。字符含义如题目描述中所述。保证有且仅有一个 **S** 和 **E**。

##### 【输出】

对于每一组数据，输出阿尔吉侬吃到奶酪的最少单位时间。若阿尔吉侬无法吃到奶酪，则输出“oop!”（只输出引号里面的内容，不输出引号）。每组数据的输出结果占一行。

**【输入样例】**

```

3
3 4
.S..
###.
..E.
3 4
.S..
.E..
....
3 4
.S..
####
..E.

```

**【输出样例】**

```

5
1
oop!

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 300
#define MOD 2520
#define E 1e-12
using namespace std;
int r,c;
char a[N][N];
bool vis[N][N];
int dir[4][2]={-1,0},{1,0},{0,-1},{0,1};
struct node
{
    int x;
    int y;
    int step;

```



```

}q[N*100];
void bfs(int sx,int sy,int ex,int ey)
{
    int head=1,tail=1;
    bool flag=true;
    memset(vis,0,sizeof(vis));

    vis[sx][sy]=1;
    q[tail].x=sx;
    q[tail].y=sy;
    q[tail].step=0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int step=q[head].step;
        if(x==ex&&y==ey)
        {
            flag=false;
            printf("%d\n",step);
            break;
        }
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(nx>=0&&nx<r&&ny>=0&&ny<c&&vis[nx][ny]==0&&a[nx][ny]!='.')
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].step=step+1;
                tail++;
            }
        }
        head++;
    }
    if(flag)
        printf("oop!\n");
}
int main()
{

```

```

int t;
scanf("%d",&t);
while(t--)
{
    int start_x,start_y,end_x,end_y;
    scanf("%d%d",&r,&c);
    for(int i=0;i<r;i++)
        scanf("%s",a[i]);
    for(int i=0;i<r;i++)
        for(int j=0;j<c;j++)
        {
            if(a[i][j]=='S')
            {
                start_x=i;
                start_y=j;
            }
            if(a[i][j]=='E')
            {
                end_x=i;
                end_y=j;
                a[i][j]='.';
            }
        }
    bfs(start_x,start_y,end_x,end_y);
}
return 0;
}

```

点赞

## 1.1.2.8.10 Knight Moves（信息学奥赛一本通-T1257）

### Knight Moves（信息学奥赛一本通-T1257）

原创 Alex\_McAvoy 发布于 2018-05-28 20:33:39 阅读数 1197 收藏  
展开

#### 【题目描述】

输入  $n$  代表有个  $n \times n$  的棋盘，输入开始位置的坐标和结束位置的坐标，问一个骑士朝棋盘的八个方向走马字步，从开始坐标到结束坐标可以经过多少步。

**【输入】**

首先输入一个  $n$ , 表示测试样例的个数。

每个测试样例有三行。

第一行是棋盘的大小  $L(4 \leq L \leq 300)$ ;

第二行和第三行分别表示马的起始位置和目标位置( $0..L-1$ )。

**【输出】**

马移动的最小步数, 起始位置和目标位置相同时输出 0。

**【输入样例】**

```
3
8
0 0
7 0
100
0 0
30 50
10
1 1
1 1
```

**【输出样例】**

```
5
28
0
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 310
#define MOD 2520
#define E 1e-12
using namespace std;
int n;
```

```

char a[N][N];
bool vis[N][N];
int dir[8][2]={{-2,1},{-1,2},{1,2},{2,1},{2,-1},{1,-2},{-1,-2},{-2,-1}};
struct node
{
    int x;
    int y;
    int step;
}q[100000];
void bfs(int sx,int sy,int ex,int ey)
{
    int head=1,tail=1;
    memset(vis,0,sizeof(vis));

    vis[sx][sy]=1;
    q[tail].x=sx;
    q[tail].y=sy;
    q[tail].step=0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        int step=q[head].step;
        if(x==ex&&y==ey)
        {
            printf("%d\n",step);
            break;
        }
        for(int i=0;i<8;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(nx>=0&&nx<n&&ny>=0&&ny<n&&vis[nx][ny]==0)
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                q[tail].step=step+1;
                tail++;
            }
        }
        head++;
    }
}

```

```

    }
}
int main()
{
    int t;
    int sx,sy,ex,ey;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);
        scanf("%d%d%d%d",&sx,&sy,&ex,&ey);
        bfs(sx,sy,ex,ey);
    }

    return 0;
}

```

---

版权声明：本文为 CSDN 博主「Alex\_McAvoy」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011815404/article/details/80488212> to insert a line break. Press Enter to add a paragraph return.)

## 1.1.2.9 第九章 动态规划

### 1.1.2.9.1 第一节 动态规划的基本模型

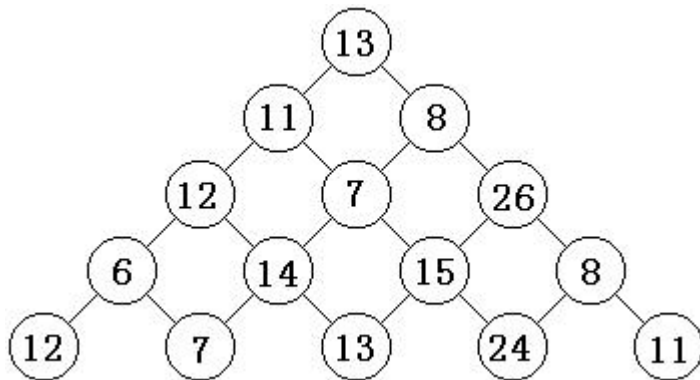
#### 1.1.2.9.1.1 T1258 数字金字塔

#### **T1258 数字金字塔 (#II-9-1)**

原创 Alex\_McAvoy 发布于 2018-05-31 20:36:39 阅读数 1260 收藏  
展开

**【题目描述】**

观察下面的数字金字塔。写一个程序查找从最高点到底部任意处结束的路径，使路径经过数字的和最大。每一步可以从当前点走到左下方的点也可以到达右下方的点。



在上面的样例中,从 13 到 8 到 26 到 15 到 24 的路径产生了最大的和 86。

**【输入】**

第一个行包含  $R(1 \leq R \leq 1000)$ ，表示行的数目。

后面每行为这个数字金字塔特定行包含的整数。

所有的被供应的整数是非负的且不大于 100。

**【输出】**

单独的一行，包含那个可能得到的最大的和。

**【输入样例】**

```

5
13
11 8
12 7 26
6 14 15 8
12 7 13 24 11
  
```

**【输出样例】**

```

86
  
```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
  
```

```

#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N];
int main()
{
    int n;

    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=i;j++)
            cin>>a[i][j];

    for(int j=1;j<=n;j++)
        f[n][j]=a[n][j];
    for(int i=n-1;i>=1;i--)
        for(int j=1;j<=i;j++)
            f[i][j]=max(f[i+1][j]+a[i][j],f[i+1][j+1]+a[i][j]);

    cout<<f[1][1]<<endl;

    return 0;
}

```

### 1.1.2.9.1.2 求最长不下降序列

#### 求最长不下降序列

原创 Alex\_McAvoy 发布于 2018-05-31 20:38:06 阅读数 2180 收藏  
展开

##### 【题目描述】

设有由  $n(1 \leq n \leq 200)$  个不相同的整数组成的数列，记为： $b(1)$ 、 $b(2)$ 、……、 $b(n)$  且  $b(i) \neq b(j)(i \neq j)$ ，若存在  $i_1 < i_2 < i_3 < \dots < i_e$  且有  $b(i_1) < b(i_2) < \dots < b(i_e)$  则称为长度为  $e$  的不下降序列。程序要求，当原数列出之后，求出最长的不下降序列。

例如 13, 7, 9, 16, 38, 24, 37, 18, 44, 19, 21, 22, 63, 15。

例中 13, 16, 18, 19, 21, 22, 63 就是一个长度为 7 的不下降序列, 同时也有 7, 9, 16, 18, 19, 21, 22, 63 组成的长度为 8 的不下降序列。

#### 【输入】

第一行为  $n$ , 第二行为用空格隔开的  $n$  个整数。

#### 【输出】

第一行为输出最大个数  $\max$  (形式见样例);

第二行为  $\max$  个整数形成的不下降序列, 答案可能不唯一, 输出一种就可以了, 本题进行特殊评测。

#### 【输入样例】

```
14
13 7 9 16 38 24 37 18 44 19 21 22 63 15
```

#### 【输出样例】

```
max=8
7 9 16 18 19 21 22 63
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N], f[N], c[N];
int main()
{
    int n;
    int maxx=-INF;

    cin>>n;
    for(int i=1; i<=n; i++)
        cin>>a[i];
```



```

int k;
for(int i=1;i<=n;i++)
{
    f[i]=1;
    for(int j=1;j<i;j++)
        if(a[j]<=a[i]&&f[j]+1>f[i])
            f[i]=f[j]+1;
    if(f[i]>maxx)
    {
        maxx=f[i];
        k=i;
    }
}
int q=0,m=maxx,i=k-1;
c[q++]=k;
while(m>1)
{
    if(f[i]==m-1&&a[i]<=a[k])
    {
        c[q++]=i;
        k=i;
        m--;
    }
    i--;
}

printf("max=%d\n",maxx);
for(i=q-1;i>=0;i--)
    printf("%d ",a[c[i]]);

return 0;
}

```

点赞 1

---

版权声明：本文为 CSDN 博主「Alex\_McAvoy」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011815404/article/details/80529637ss> Shift-Enter to insert a line break. Press Enter to add a paragraph return.)

### 1.1.2.9.1.3 拦截导弹

#### 拦截导弹

原创 Alex\_McAvoy 发布于 2018-05-31 20:54:21 阅读数 835 收藏  
展开

### 【题目描述】

某国为了防御敌国的导弹袭击，发展出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。

输入导弹依次飞来的高度（雷达给出的高度数据是不大于 30000 的正整数，导弹数不超过 1000），计算这套系统最多能拦截多少导弹，如果要拦截所有导弹最少要配备多少套这种导弹拦截系统。

### 【输入】

输入导弹依次飞来的高度。

### 【输出】

第一行：最多能拦截的导弹数；

第二行：要拦截所有导弹最少要配备的系统数。

### 【输入样例】

389 207 155 300 299 170 158 65

### 【输出样例】

6

2

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],b[N],c[N];
```

```

int main()
{
    int n=0,maxx=1;

    while(scanf("%d",&a[n++])!=EOF)

    for(int i=0;i<n;i++)
    {
        b[i]=1;
        for(int j=0;j<i;j++)
            if(a[j]>=a[i]&&b[j]+1>b[i])
                b[i]=b[j]+1;
        maxx=max(maxx,b[i]);
    }
    printf("%d\n",maxx);

    maxx=1;
    for(int i=0;i<n;i++)
    {
        c[i]=1;
        for(int j=0;j<i;j++)
            if(a[j]<a[i]&&c[j]+1>c[i])
                c[i]=c[j]+1;
        maxx=max(maxx,c[i]);
    }
    printf("%d\n",maxx);
    return 0;
}

```

#### 1.1.2.9.1.4 城市交通路网

##### 城市交通路网

原创 Alex\_McAvoy 发布于 2018-05-31 21:02:53 阅读数 1057 收藏  
展开

##### 【题目描述】

下图表示城市之间的交通路网，线段上的数字表示费用，单向通行由 A->E。试用动态规划的最优化原理求出 A->E 的最省费用。

如图：求 v1 到 v10 的最短路径长度及最短路径。

### 【输入】

第一行为城市的数量 N;

后面是 N\*N 的表示两个城市间费用组成的矩阵。

### 【输出】

A->E 的最省费用。

### 【输入样例】

```
10
0 2 5 1 0 0 0 0 0 0
0 0 0 0 12 14 0 0 0 0
0 0 0 0 6 10 4 0 0 0
0 0 0 0 13 12 11 0 0 0
0 0 0 0 0 0 0 3 9 0
0 0 0 0 0 0 0 6 5 0
0 0 0 0 0 0 0 0 10 0
0 0 0 0 0 0 0 0 0 5
0 0 0 0 0 0 0 0 0 2
0 0 0 0 0 0 0 0 0 0
```

### 【输出样例】

```
minlong=19
1 3 5 8 10
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
```

```

int a[N][N],f[N],c[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            cin>>a[i][j];

    memset(f,INF,sizeof(f));
    f[n]=0;

    for(int i=n-1;i>=1;i--)
        for(int j=i+1;j<=n;j++)
            if( a[i][j]>0 && f[j]!=INF && (f[i]>a[i][j]+f[j]) )
            {
                f[i]=a[i][j]+f[j];
                c[i]=j;
            }

    printf("minlong=%d\n",f[1]);
    int i=1;
    while(i!=0)
    {
        cout<<i<<" ";
        i=c[i];
    }
    cout<<endl;

    return 0;
}

```

### 1.1.2.9.1.5 挖地雷

#### 挖地雷

原创 Alex\_McAvoy 发布于 2018-05-31 21:05:21 阅读数 804 收藏  
展开

#### 【题目描述】

在一个地图上有  $n$  个地窖 ( $n \leq 200$ ), 每个地窖中埋有一定数量的地雷。同时, 给出地窖之间的连接路径, 并规定路径都是单向的, 且保证都是小序号地窖指向在序号地窖,

也不存在可以从一个地窖出发经过若干地窖后又回到原来地窖的路径。某人可以从任一处开始挖地雷，然后沿着指出的连接往下挖（仅能选择一条路径），当无连接时挖地雷工作结束。设计一个挖地雷的方案，使他能挖到最多的地雷。

### 【输入】

第一行：地窖的个数；

第二行为依次每个地窖地雷的个数；

下面若干行：

$x_i y_i$  //表示从  $x_i$  可到  $y_i$ ,  $x_i < y_i$ 。

最后一行为"0 0"表示结束。

### 【输出】

$k_1?k_2?...?k_v$  //挖地雷的顺序  
挖到最多的雷。

### 【输入样例】

```
6
5 10 20 5 4 5
1 2
1 4
2 4
3 4
4 5
4 6
5 6
0 0
```

### 【输出样例】

```
3-4-5-6
34
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<stdlib.h>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
```

```

#define PI acos(-1.0)
#define N 201
#define MOD 2520
#define E 1e-12
using namespace std;
int g[N][N];
int w[N],pre[N];
int f[N];
void print(int k)
{
    if(k==0)
        return;

    print(pre[k]);

    if(pre[k]==0)
        cout<<k;
    else
        cout<<"-"<<k;
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>w[i];
        f[i]=w[i];
    }

    int x,y;
    while(scanf("%d%d",&x,&y)!=EOF&&x&&y)
        g[x][y]=1;

    int maxx=-INF,k;
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            if( g[j][i]==1 && f[j]+w[i]>f[i] )
            {
                f[i]=f[j]+w[i];
                pre[i]=j;
            }
        }
    }
}

```

```

    }
    if(f[i]>maxx)
    {
        maxx=f[i];
        k=i;
    }
}
print(k);
cout<<endl<<maxx<<endl;
return 0;
}

```

### 1.1.2.9.1.6 友好城市

#### 友好城市

原创 Alex\_McAvoy 发布于 2018-05-31 21:10:24 阅读数 815 收藏  
展开

#### 【题目描述】

**Palmia** 国有一条横贯东西的大河，河有笔直的南北两岸，岸上各有位置各不相同的  $N$  个城市。北岸的每个城市有且仅有一个友好城市在南岸，而且不同城市的友好城市不相同。

每对友好城市都向政府申请在河上开辟一条直线航道连接两个城市，但是由于河上雾太大，政府决定避免任意两条航道交叉，以避免事故。编程帮助政府做出一些批准和拒绝申请的决定，使得在保证任意两条航线不相交的情况下，被批准的申请尽量多。

#### 【输入】

第 1 行，一个整数  $N(1 \leq N \leq 5000)$ ，表示城市数。

第 2 行到第  $n+1$  行，每行两个整数，中间用 1 个空格隔开，分别表示南岸和北岸的一对友好城市的坐标。 $(0 \leq x_i \leq 10000)$

#### 【输出】

仅一行，输出一个整数，表示政府所能批准的最多申请数。

#### 【输入样例】

```

7
22 4
2 6

```



10 3  
15 12  
9 8  
17 17  
4 2

【输出样例】

4

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 5001
#define MOD 2520
#define E 1e-12
using namespace std;
struct Node{
    int a;
    int b;
}q[N];
int f[N];
int main()
{
    int n;

    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>q[i].a>>q[i].b;

    for(int i=1;i<=n;i++)
        for(int j=i+1;j<=n;j++)
        {
            if(q[i].a>q[j].a)
                swap(q[i],q[j]);
            else if(q[i].a==q[j].a)
            {
                if(q[i].b>q[j].b)
```

```

        swap(q[i],q[j]);
    }
}

int maxx=-INF;
for(int i=1;i<=n;i++)
{
    f[i]=1;
    for(int j=1;j<i;j++)
        if(q[j].b<=q[i].b&&f[j]+1>f[i])
            f[i]=f[j]+1;
    maxx=max(maxx,f[i]);
}
cout<<maxx<<endl;
return 0;
}

```

### 1.1.2.9.1.7 合唱队形

#### 合唱队形

原创 Alex\_McAvoy 发布于 2018-05-31 21:14:33 阅读数 520 收藏  
展开

#### 【题目描述】

$N$  位同学站成一排，音乐老师要请其中的( $N \geq K$ )位同学出列，使得剩下的  $K$  位同学排成合唱队形。

合唱队形是指这样的一种队形：设  $K$  位同学从左到右依次编号为  $1, 2, \dots, K$ ，他们的身高分别为  $T_1, T_2, \dots, T_K$ ，则他们的身高满足  $T_1 < T_2 < \dots < T_i, T_i > T_{i+1} > \dots > T_K (1 \leq i \leq K)$ 。

你的任务是，已知所有  $N$  位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。

#### 【输入】

输入的第一行是一个整数  $N$  ( $2 \leq N \leq 100$ )，表示同学的总数。第二行有  $n$  个整数，用空格分隔，第  $i$  个整数  $T_i$  ( $130 \leq T_i \leq 230$ ) 是第  $i$  位同学的身高（厘米）。

**【输出】**

输出包括一行，这一行只包含一个整数，就是最少需要几位同学出列。

**【输入样例】**

```
8
186 186 150 200 160 130 197 220
```

**【输出样例】**

```
4
```

**【源程序】**

```
#include<iostream>
using namespace std;

int cmp(int x,int y)
{
    if(x>y) return x;
    else return y;
}
int dp_rise[10000],dp_fall[10000];
int main()
{
    int n,high[10000];
    int rise_temp,fall_temp,temp;
    int max=1;//留下人数
    int i,j;

    cin>>n;//人数
    for(i=1;i<=n;i++)
    {
        cin>>high[i];
        dp_rise[i]=1;
        dp_fall[i]=1;
    }

    for(i=1;i<=n;i++)
    {
        rise_temp=0;
        for(j=1;j<i;j++)
        {
            if(high[i]>high[j])
            {
                if(rise_temp<dp_rise[j])
                    rise_temp=dp_rise[j];
            }
        }
    }
}
```

```

        dp_rise[i]=rise_temp+1;
    }

    for(i=n;i>=1;i--)
    {
        fall_temp=0;
        for(j=n;j>i;j--)
        {
            if(high[i]>high[j])
            {
                if(fall_temp<dp_fall[j])
                    fall_temp=dp_fall[j];
            }
        }
        dp_fall[i]=fall_temp+1;
    }

    for(i=1;i<=n;i++)
    {
        temp=dp_rise[i]+dp_fall[i]-1;
        max=cmp(max,temp);
    }

    cout<<n-max<<endl;

    return 0;
}

```

### 1.1.2.9.1.8 最长公共子序列

#### 最长公共子序列

原创 Alex\_McAvoy 发布于 2018-05-31 21:18:51 阅读数 1050 收藏  
展开

#### 【题目描述】

一个给定序列的子序列是在该序列中删去若干元素后得到的序列。确切地说，若给定序列  $X=\langle x_1, x_2, \dots, x_m \rangle$ ，则另一序列  $Z=\langle z_1, z_2, \dots, z_k \rangle$  是  $X$  的子序列是指存在一个严格递增的下标序列  $\langle i_1, i_2, \dots, i_k \rangle$ ，使得对于所有  $j=1, 2, \dots, k$  有： $X_{i_j}=Z_j$

例如，序列  $Z=\langle B, C, D, B \rangle$  是序列  $X=\langle A, B, C, B, D, A, B \rangle$  的子序列，相应的递增下标序列为  $\langle 2, 3, 5, 7 \rangle$ 。给定两个序列  $X$  和  $Y$ ，当另一序列  $Z$  既是  $X$  的子序列又是  $Y$  的子序列时，称  $Z$  是序列  $X$  和  $Y$  的公共子序列。例如，若  $X=\langle A, B, C, B, D, A, B \rangle$  和  $Y=\langle B, D, C, A, B, A \rangle$ ，则序列  $\langle B, C, A \rangle$  是  $X$  和  $Y$  的一个公共子序列，序列  $\langle B, C, B, A \rangle$  也是  $X$  和  $Y$  的一个公共子序列。而且，后者是  $X$  和  $Y$  的一个最长公共子序列。因为  $X$  和  $Y$  没有长度大于 4 的公共子序列。

给定两个序列  $X=\langle x_1, x_2, \dots, x_m \rangle$  和  $Y=\langle y_1, y_2, \dots, y_n \rangle$ 。要求找出  $X$  和  $Y$  的一个最长公共子序列。

#### 【输入】

共有两行。每行为一个由大写字母构成的长度不超过 1000 的字符串，表示序列  $X$  和  $Y$ 。

#### 【输出】

第一行为一个非负整数。表示所求得的最长公共子序列的长度。若不存在公共子序列，则输出文件仅有一行输出一个整数 0。

#### 【输入样例】

```
ABCB DAB
BDCABA
```

#### 【输出样例】

```
4
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
```

```

using namespace std;
char a[N],b[N];
int f[N][N];
int main()
{
    scanf("%s%s",&a[1],&b[1]);
    int len_a=strlen(&a[1]);
    int len_b=strlen(&b[1]);

    for(int i=0;i<=len_a;i++)
        f[i][0]=0;
    for(int i=0;i<=len_b;i++)
        f[0][i]=0;
    for(int i=1;i<=len_a;i++)
        for(int j=1;j<=len_b;j++)
            if(a[i]==b[j])
                f[i][j]=f[i-1][j-1]+1;
            else
                f[i][j]=max(f[i-1][j],f[i][j-1]);

    cout<<f[len_a][len_b]<<endl;
    return 0;
}

```

点赞

### 1.1.2.9.1.9 机器分配

#### 机器分配

原创 Alex\_McAvoy 发布于 2018-05-31 21:20:08 阅读数 729 收藏  
展开

#### 【题目描述】

总公司拥有高效设备  $M$  台，准备分给下属的  $N$  个分公司。各分公司若获得这些设备，可以为国家提供一定的盈利。问：如何分配这  $M$  台设备才能使国家得到的盈利最大？

求出最大盈利值。其中  $M \leq 15$ ,  $N \leq 10$ 。分配原则：每个公司有权获得任意数目的设备，但总台数不超过设备数  $M$ 。

#### 【输入】

第一行有两个数，第一个数是分公司数  $N$ ，第二个数是设备台数  $M$ ；

接下来是一个  $N \times M$  的矩阵，表明了第  $I$  个公司分配  $J$  台机器的盈利。

#### 【输出】

第一行输出最大盈利值；

接下  $N$  行，每行有 2 个数，即分公司编号和该分公司获得设备台数。

#### 【输入样例】

```
3 3          //3 个分公司分 3 台机器
30 40 50
20 30 50
20 25 30
```

#### 【输出样例】

```
70           //最大盈利值为 70
1 1          //第一分公司分 1 台
2 1          //第二分公司分 1 台
3 1          //第三分公司分 1 台
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 30
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N],res[N][N];
void print(int x,int y)
{
    if(x==0)
        return;
    print(x-1,y-res[x][y]);
    printf("%d %d\n",x,res[x][y]);
}
```

```

}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            cin>>a[i][j];

    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            for(int k=0;k<=j;k++)
                if(f[i][j]<=f[i-1][j-k]+a[i][k])
                {
                    f[i][j]=f[i-1][j-k]+a[i][k];
                    res[i][j]=k;
                }

    cout<<f[n][m]<<endl;
    print(n,m);

    return 0;
}

```

### 1.1.2.9.1.10 最长上升子序列

#### 最长上升子序列

原创 Alex\_McAvoy 发布于 2018-06-01 20:20:34 阅读数 988 收藏  
展开

##### 【题目描述】

一个数的序列  $b_i$ ，当  $b_1 < b_2 < \dots < b_S$  的时候，我们称这个序列是上升的。对于给定的一个序列  $(a_1, a_2, \dots, a_N)$ ，我们可以得到一些上升的子序列  $(a_{i_1}, a_{i_2}, \dots, a_{i_K})$ ，这里  $1 \leq i_1 < i_2 < \dots < i_K \leq N$ 。比如，对于序列  $(1, 7, 3, 5, 9, 4, 8)$ ，有它的一些上升子序列，如  $(1, 7), (3, 4, 8)$  等等。这些子序列中最长的长度是 4，比如子序列  $(1, 3, 5, 8)$ 。

你的任务，就是对于给定的序列，求出最长上升子序列的长度。

##### 【输入】

输入的第一行是序列的长度  $N (1 \leq N \leq 1000)$ 。第二行给出序列中的  $N$  个整数，这些整数的取值范围都在 0 到 10000。



**【输出】**

最长上升子序列的长度。

**【输入样例】**

7

1 7 3 5 9 4 8

**【输出样例】**

4

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<stdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],f[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    int maxx=-INF;
    for(int i=1;i<=n;i++)
    {
        f[i]=1;
        for(int j=1;j<i;j++)
            if(a[j]<a[i]&&f[j]+1>f[i])
                f[i]=f[j]+1;
        maxx=max(maxx,f[i]);
    }
    cout<<maxx<<endl;
    return 0;
}
```

### 1.1.2.9.1.11 最大子矩阵

#### 最大子矩阵

原创 Alex\_McAvoy 发布于 2018-06-01 21:11:00 阅读数 1066 收藏  
展开

##### 【题目描述】

已知矩阵的大小定义为矩阵中所有元素的和。给定一个矩阵，你的任务是找到最大的非空(大小至少是  $1 \times 1$ )子矩阵。

比如，如下  $4 \times 4$  的矩阵

0 -2 -7 0

9 2 -6 2

-4 1 -4 1

-1 8 0 -2

的最大子矩阵是

9 2

-4 1

-1 8

这个子矩阵的大小是 15。

##### 【输入】

输入是一个  $N \times N$  的矩阵。输入的第一行给出  $N(0 < N \leq 100)$ 。再后面的若干行中，依次(首先从左到右给出第一行的  $N$  个整数，再从左到右给出第二行的  $N$  个整数.....)给出矩阵中的  $N^2$  个整数，整数之间由空白字符分隔(空格或者空行)。已知矩阵中整数的范围都在  $[-127, 127]$ 。

**【输出】**

输出最大子矩阵的大小。

**【输入样例】**

```
4
0 -2 -7 0
9 2 -6 2
-4 1 -4 1
-1 8 0 -2
```

**【输出样例】**

```
15
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N];
int maxArray(int t[],int n)
{
    int sum=0,maxx=-INF;
    for(int i=1;i<=n;i++)
    {
        if(sum>0)
            sum+=t[i];
        else
            sum=t[i];
        if(sum>maxx)
            maxx=sum;
    }
    return maxx;
}
int main()
{
    int n;
```

```

cin>>n;
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        cin>>a[i][j];

int maxx=-INF;
for(int i=1;i<=n;i++)
{
    memset(f,0,sizeof(f));
    for(int j=i;j<=n;j++)
    {
        for(int k=1;k<=n;k++)
            f[k]+=a[j][k];

        int temp=maxArray(f,n);
        if(temp>maxx)
            maxx=temp;
    }
}

cout<<maxx<<endl;

return 0;
}

```

### 1.1.2.9.1.12 登山

#### 登山

原创 Alex\_McAvoy 发布于 2018-06-01 20:44:02 阅读数 1283 收藏  
展开

##### 【题目描述】

五一到了，ACM 队组织大家去登山观光，队员们发现山上一个有  $N$  个景点，并且决定按照顺序来浏览这些景点，即每次所浏览景点的编号都要大于前一个浏览景点的编号。同时队员们还有另一个登山习惯，就是不连续浏览海拔相同的两个景点，并且一旦开始下山，就不再向上走了。队员们希望在满足上面条件的同时，尽可能多的浏览景点，你能帮他们找出最多可能浏览的景点数么？

##### 【输入】

第一行：  $N$  ( $2 \leq N \leq 1000$ ) 景点数;

第二行：  $N$  个整数，每个景点的海拔。

**【输出】**

最多能浏览的景点数。

**【输入样例】**

8  
186 186 150 200 160 130 197 220

**【输出样例】**

4

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],f[N],d[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    for(int i=1;i<=n;i++)
    {
        f[i]=1;
        for(int j=1;j<i;j++)
            if(a[j]<a[i]&&f[j]+1>f[i])
                f[i]=f[j]+1;
    }
    for(int i=n;i>=1;i--)
    {
        d[i]=1;
        for(int j=n;j>i;j--)
            if(a[j]<a[i]&&d[j]+1>d[i])
                d[i]=d[j]+1;
    }
}
```

```

}

int maxx=-INF;
for(int i=1;i<=n;i++)
    maxx=max(maxx,f[i]+d[i]-1);

cout<<maxx<<endl;

return 0;
}

```

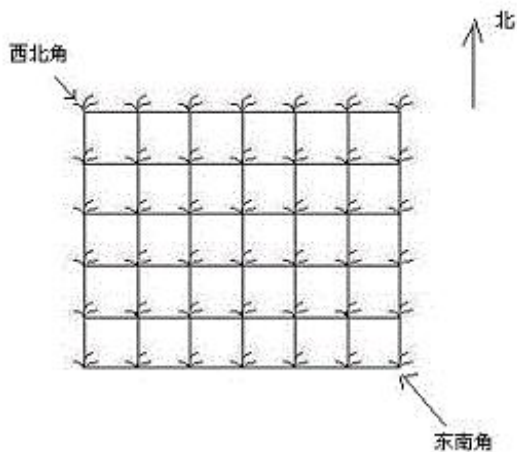
### 1.1.2.9.1.13 摘花生

#### 摘花生

原创 Alex\_McAvoy 发布于 2018-06-01 20:40:06 阅读数 1030 收藏  
展开

##### 【题目描述】

Hello Kitty 想摘点花生送给她喜欢的米老鼠。她来到一片有网格状道路的矩形花生地(如下图), 从西北角进去, 东南角出来。地里每个道路的交叉点上都有种着一株花生苗, 上面有若干颗花生, 经过一株花生苗就能摘走该它上面所有的花生。Hello Kitty 只能向东或向南走, 不能向西或向北走。问 Hello Kitty 最多能够摘到多少颗花生。



##### 【输入】

第一行是一个整数  $T$ , 代表一共有多少组数据。  $1 \leq T \leq 100$

接下来是  $T$  组数据。

每组数据的第一行是两个整数，分别代表花生苗的行数  $R$  和列数  $C$  ( $1 \leq R, C \leq 100$ )

每组数据的接下来  $R$  行数据，从北向南依次描述每行花生苗的情况。每行数据有  $C$  个整数，按从西向东的顺序描述了该行每株花生苗上的花生数目  $M$  ( $0 \leq M \leq 1000$ )。

#### 【输出】

对每组输入数据，输出一行，内容为 Hello Kitty 能摘到得最多的花生颗数。

#### 【输入样例】

```
2
2 2
1 1
3 4
2 3
2 3 4
1 6 5
```

#### 【输出样例】

```
8
16
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N];
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        memset(f,0,sizeof(f));
        int m,n;
        cin>>m>>n;
```

```

    for(int i=1;i<=m;i++)
        for(int j=1;j<=n;j++)
            cin>>a[i][j];
    for(int i=1;i<=m;i++)
        for(int j=1;j<=n;j++)
            f[i][j]=max(f[i-1][j],f[i][j-1])+a[i][j];
    cout<<f[m][n]<<endl;
}
return 0;
}

```

### 1.1.2.9.1.14 最大上升子序列和

#### 最大上升子序列和

原创 Alex\_McAvoy 发布于 2018-06-01 20:36:12 阅读数 846 收藏  
展开

##### 【题目描述】

一个数的序列  $b_i$ ，当  $b_1 < b_2 < \dots < b_S$  的时候，我们称这个序列是上升的。对于给定的一个序列  $(a_1, a_2, \dots, a_N)$ ，我们可以得到一些上升的子序列  $(a_{i_1}, a_{i_2}, \dots, a_{i_K})$ ，这里  $1 \leq i_1 < i_2 < \dots < i_K \leq N$ 。比如，对于序列  $(1, 7, 3, 5, 9, 4, 8)$ ，有它的一些上升子序列，如  $(1, 7), (3, 4, 8)$  等等。这些子序列中和最大为 18，为子序列  $(1, 3, 5, 9)$  的和。

你的任务，就是对于给定的序列，求出最大上升子序列和。注意，最长的上升子序列的和不一定是最大的，比如序列  $(100, 1, 2, 3)$  的最大上升子序列和为 100，而最长上升子序列为  $(1, 2, 3)$ 。

##### 【输入】

输入的第一行是序列的长度  $N (1 \leq N \leq 1000)$ 。第二行给出序列中的  $N$  个整数，这些整数的取值范围都在 0 到 10000(可能重复)。

##### 【输出】

最大上升子序列和。

##### 【输入样例】

```

7
1 7 3 5 9 4 8

```

##### 【输出样例】

```

18

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>

```



```

#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],f[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    int maxx=-INF;
    for(int i=1;i<=n;i++)
    {
        f[i]=a[i];
        for(int j=1;j<i;j++)
            if(a[j]<a[i]&&f[j]+a[i]>f[i])
                f[i]=f[j]+a[i];
        maxx=max(maxx,f[i]);
    }

    cout<<maxx<<endl;

    return 0;
}

```

### 1.1.2.9.1.15 怪盗基德的滑翔翼

#### 怪盗基德的滑翔翼

原创 Alex\_McAvoy 发布于 2018-06-01 20:34:14 阅读数 1112 收藏  
展开

**【题目描述】**

怪盗基德是一个充满传奇色彩的怪盗，专门以珠宝为目标的超级盗窃犯。而他最为突出的地方，就是他每次都能逃脱中村警部的重重围堵，而这也很大程度上是多亏了他随身携带的便于操作的滑翔翼。

有一天，怪盗基德像往常一样偷走了一颗珍贵的钻石，不料却被柯南小朋友识破了伪装，而他的滑翔翼的动力装置也被柯南踢出的足球破坏了。不得已，怪盗基德只能操作受损的滑翔翼逃脱。

假设城市中一共有  $N$  幢建筑排成一条线，每幢建筑的高度各不相同。初始时，怪盗基德可以在任何一幢建筑的顶端。他可以选择一个方向逃跑，但是不能中途改变方向（因为中森警部会在后面追击）。因为滑翔翼动力装置受损，他只能往下滑行（即：只能从较高的建筑滑翔到较低的建筑）。他希望尽可能多地经过不同建筑的顶部，这样可以减缓下降时的冲击力，减少受伤的可能性。请问，他最多可以经过多少幢不同建筑的顶部(包含初始时的建筑)?

**【输入】**

输入数据第一行是一个整数  $K(K < 100)$ ，代表有  $K$  组测试数据。

每组测试数据包含两行：第一行是一个整数  $N(N < 100)$ ，代表有  $N$  幢建筑。第二行包含  $N$  个不同的整数，每一个对应一幢建筑的高度  $h(0 < h < 10000)$ ，按照建筑的排列顺序给出。

**【输出】**

对于每一组测试数据，输出一行，包含一个整数，代表怪盗基德最多可以经过的建筑数量。

**【输入样例】**

```
3
8
300 207 155 299 298 170 158 65
8
65 158 170 298 299 155 207 300
10
2 1 3 4 5 6 7 8 9 10
```

**【输出样例】**

```
6
6
9
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
```

```

#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],f[N];
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n;
        cin>>n;
        for(int i=1;i<=n;i++)
            cin>>a[i];

        int maxx1=-INF,maxx2=-INF;
        for(int i=1;i<=n;i++)
        {
            f[i]=1;
            for(int j=1;j<i;j++)
                if(a[j]>a[i]&&f[j]+1>f[i])
                    f[i]=f[j]+1;
            maxx1=max(maxx1,f[i]);
        }
        for(int i=n;i>=1;i--)
        {
            f[i]=1;
            for(int j=n;j>i;j--)
                if(a[j]>a[i]&&f[j]+1>f[i])
                    f[i]=f[j]+1;
            maxx2=max(maxx2,f[i]);
        }

        int res=max(maxx1,maxx2);
        cout<<res<<endl;
    }
}

```

```
    return 0;
}
```

### 1.1.2.9.1.16 最低通行费

#### 最低通行费

原创 Alex\_McAvoy 发布于 2018-06-01 20:27:43 阅读数 898 收藏  
展开

##### 【题目描述】

一个商人穿过一个  $N \times N$  的正方形的网格，去参加一个非常重要的商务活动。他要从网格的左上角进，右下角出。每穿越中间 1 个小方格，都要花费 1 个单位时间。商人必须在  $(2N-1)$  个单位时间穿越出去。而在经过中间的每个小方格时，都需要缴纳一定的费用。

这个商人期望在规定时间内用最少费用穿越出去。请问至少需要多少费用？

注意：不能对角穿越各个小方格（即，只能向上下左右四个方向移动且不能离开网格）。

##### 【输入】

第一行是一个整数，表示正方形的宽度  $N$  ( $1 \leq N < 100$ )；

后面  $N$  行，每行  $N$  个不大于 100 的整数，为网格上每个小方格的费用。

##### 【输出】

至少需要的费用。

##### 【输入样例】

```
5
1 4 6 8 10
2 5 7 15 17
6 8 9 18 20
10 11 12 19 21
20 23 25 29 33
```

##### 【输出样例】

```
109
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
```

```

#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            cin>>a[i][j];

    f[1][1]=a[1][1];
    for(int i=2;i<=n;i++)
        f[i][1]=f[i-1][1]+a[i][1];
    for(int j=2;j<=n;j++)
        f[1][j]=f[1][j-1]+a[1][j];

    for(int i=2;i<=n;i++)
        for(int j=2;j<=n;j++)
            f[i][j]=min(f[i-1][j],f[i][j-1])+a[i][j];

    cout<<f[n][n]<<endl;

    return 0;
}

```

### 1.1.2.9.1.17 三角形最佳路径问题

#### 三角形最佳路径问题

原创 Alex\_McAvoy 发布于 2018-06-01 20:26:29 阅读数 520 收藏  
展开

**【题目描述】**

如下所示的由正整数数字构成的三角形：

```

7
3 8
8 1 0
2 7 4 4
4 5 2 6 5

```

从三角形的顶部到底部有很多条不同的路径。对于每条路径，把路径上面的数加起来可以得到一个和，和最大的路径称为最佳路径。你的任务就是求出最佳路径上的数字之和。

注意：路径上的每一步只能从一个数走到下一层上和它最近的下边(正下方)的数或者右边（右下方）的数。

**【输入】**

第一行为三角形高度  $100 \geq h \geq 1$ ，同时也是最底层边的数字的数目。

从第二行开始，每行为三角形相应行的数字，中间用空格分隔。

**【输出】**

最佳路径的长度数值。

**【输入样例】**

```

5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5

```

**【输出样例】**

```

30

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>

```

```

#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=i;j++)
            cin>>a[i][j];

    for(int i=n;i>=1;i--)
        for(int j=1;j<=i;j++)
            f[i][j]=max(f[i+1][j]+a[i][j],f[i+1][j+1]+a[i][j]);

    cout<<f[1][1]<<endl;

    return 0;
}

```

### 1.1.2.9.1.18 拦截导弹

#### 拦截导弹

原创 Alex\_McAvoy 发布于 2018-06-01 20:23:25 阅读数 1013 收藏  
展开

#### 【题目描述】

某国为了防御敌国的导弹袭击，发展出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。

输入导弹依次飞来的高度(雷达给出的高度数据是不大于 30000 的正整数)，计算这套系统最多能拦截多少导弹。

**【输入】**

第一行是一个整数  $N$ (不超过 15)，表示导弹数。

第二行包含  $N$  个整数，为导弹依次飞来的高度(雷达给出的高度数据是不大于 30000 的正整数)。

**【输出】**

一个整数，表示最多能拦截的导弹数。

**【输入样例】**

```
8
389 207 155 300 299 170 158 65
```

**【输出样例】**

```
6
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],f[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    int maxx=-INF;
    for(int i=1;i<=n;i++)
    {
        f[i]=1;
        for(int j=1;j<i;j++)
            if(a[j]>=a[i]&&f[j]+1>f[i])
                f[i]=f[j]+1;
        maxx=max(maxx,f[i]);
    }
}
```



```

    }
    cout<<maxx<<endl;
    return 0;
}

```

## 1.1.2.9.2 第二节 背包问题

Help > [信息学奥赛一本通](#) > [语言及算法基础篇](#) > [第二部分 基础算法](#) > [第九章 动态规划](#) > 第二节 背包问题

### 1.1.2.9.2.1 背包问题

#### 背包问题

原创 Alex\_McAvoy 发布于 2018-06-02 22:24:35 阅读数 851 收藏  
展开

##### 【题目描述】

一个旅行者有一个最多能装  $M$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ ，求旅行者能获得最大总价值。

##### 【输入】

第一行：两个整数， $M$ (背包容量， $M \leq 200$ )和  $N$ (物品数量， $N \leq 30$ )；

第  $2..N+1$  行：每行二个整数  $W_i, C_i$ ，表示每个物品的重量和价值。

##### 【输出】

仅一行，一个数，表示最大总价值。

##### 【输入样例】

```

10 4
2 1
3 3
4 5
7 9

```

##### 【输出样例】

```

12

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>

```

```

#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int m,n;
int w[N],c[N],f[N];
void ZeroOnePack(int cost,int weight)
{
    for(int v=m;v>=weight;v--)
        f[v]=max(f[v],f[v-weight]+cost);
}

int main()
{
    cin>>m>>n;
    for(int i=1;i<=n;i++)
        cin>>w[i]>>c[i];
    for(int i=1;i<=n;i++)
        ZeroOnePack(c[i],w[i]);
    cout<<f[m]<<endl;
    return 0;
}

```

### 1.1.2.9.2.2 完全背包问题

#### 完全背包问题

原创 Alex\_McAvoy 发布于 2018-06-02 22:23:11 阅读数 841 收藏  
展开

##### 【题目描述】

设有  $n$  种物品，每种物品有一个重量及一个价值。但每种物品的数量是无限的，同时有一个背包，最大载重量为  $M$ ，今从  $n$  种物品中选取若干件(同一种物品可以多次选取)，使其重量的和小于等于  $M$ ，而价值的和为最大。

##### 【输入】

第一行：两个整数，M(背包容量， $M \leq 200$ )和 N(物品数量， $N \leq 30$ )；

第 2..N+1 行：每行二个整数  $W_i, C_i$ ，表示每个物品的重量和价值。

**【输出】**

仅一行，一个数，表示最大总价值。

**【输入样例】**

10 4

2 1

3 3

4 5

7 9

**【输出样例】**

max=12

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int m,n;
int w[N],c[N],f[N];
void CompletePack(int cost,int weight)
{
    for(int v=weight;v<=m;v++)
        f[v]=max(f[v],f[v-weight]+cost);
}

int main()
{
    cin>>m>>n;
    for(int i=1;i<=n;i++)
        cin>>w[i]>>c[i];
    for(int i=1;i<=n;i++)
```

```

    CompletePack(c[i],w[i]);
    cout<<"max="<<f[m]<<endl;
    return 0;
}

```

### 1.1.2.9.2.3 庆功会

#### 庆功会

原创 Alex\_McAvoy 发布于 2018-06-02 22:22:04 阅读数 525 收藏  
展开

##### 【题目描述】

为了庆贺班级在校运动会上取得全校第一名成绩，班主任决定开一场庆功会，为此拨款购买奖品犒劳运动员。期望拨款金额能购买最大价值的奖品，可以补充他们的精力和体力。

##### 【输入】

第一行二个数  $n(n \leq 500)$ ， $m(m \leq 6000)$ ，其中  $n$  代表希望购买的奖品的种数， $m$  表示拨款金额。

接下来  $n$  行，每行 3 个数， $v$ 、 $w$ 、 $s$ ，分别表示第  $i$  种奖品的价格、价值（价格与价值是不同的概念）和能购买的最大数量（买 0 件到  $s$  件均可），其中  $v \leq 100$ ， $w \leq 1000$ ， $s \leq 10$ 。

##### 【输出】

一行：一个数，表示此次购买能获得的最大的价值（注意！不是价格）。

##### 【输入样例】

```

5 1000
80 20 4
40 50 9
30 50 7
40 30 6
20 20 1

```

##### 【输出样例】

```

1040

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>

```

```

#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 2520
#define E 1e-12
using namespace std;
int m,n;
int w[N],c[N],num[N],f[N];
void MultiplePack(int cost,int weight,int num)
{
    for(int j=m;j>=0;j--)
        for(int k=0;k<=num;k++)
            if(j-k*weight>=0)
                f[j]=max(f[j],f[j-k*weight]+k*cost);
}
int main()
{
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        cin>>w[i]>>c[i]>>num[i];
    for(int i=1;i<=n;i++)
        MultiplePack(c[i],w[i],num[i]);

    cout<<f[m]<<endl;
    return 0;
}

```

#### 1.1.2.9.2.4 混合背包

##### 混合背包

原创 Alex\_McAvoy 发布于 2018-06-02 22:20:49 阅读数 483 收藏  
展开

##### 【题目描述】

一个旅行者有一个最多能装  $V$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ 。有的物品只可以取一次（01 背包），有的物品可以取无限次（完全背包），有的物品可以取的次数有一个上限（多重背包）。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

**【输入】**

第一行：二个整数， $M$ (背包容量， $M \leq 200$ )， $N$ (物品数量， $N \leq 30$ )；

第  $2..N+1$  行：每行三个整数  $W_i, C_i, P_i$ ，前两个整数分别表示每个物品的重量，价值，第三个整数若为 0，则说明此物品可以购买无数件，若为其他数字，则为此物品可购买的最多件数( $P_i$ )。

**【输出】**

仅一行，一个数，表示最大总价值。

**【输入样例】**

```
10 3
2 1 0
3 3 1
4 5 4
```

**【输出样例】**

```
11
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 2520
#define E 1e-12
using namespace std;
int m,n;
int w[N],c[N],num[N],f[N];
void ZeroOnePack(int cost,int weight)
{
    for(int v=m;v>=weight;v--)
        f[v]=max(f[v],f[v-weight]+cost);
}
void CompletePack(int cost,int weight)
{
    for(int v=weight;v<=m;v++)
        f[v]=max(f[v],f[v-weight]+cost);
}
```

```

void MultiplePack(int cost,int weight,int num)
{
    for(int j=m;j>=0;j--)
        for(int k=0;k<=num;k++)
            if(j-k*weight>=0)
                f[j]=max(f[j],f[j-k*weight]+k*cost);
}
int main()
{
    cin>>m>>n;
    for(int i=1;i<=n;i++)
        cin>>w[i]>>c[i]>>num[i];

    for(int i=1;i<=n;i++)
    {
        if(num[i]==1)//0-1 背包
            ZeroOnePack(c[i],w[i]);
        else if(num[i]==0)//完全背包
            CompletePack(c[i],w[i]);
        else//多重背包
            MultiplePack(c[i],w[i],num[i]);
    }
    cout<<f[m]<<endl;
    return 0;
}

```

### 1.1.2.9.2.5 潜水员

#### 潜水员

##### 【题目描述】

潜水员为了潜水要使用特殊的装备。他有一个带 2 种气体的气缸：一个为氧气，一个为氮气。让潜水员下潜的深度需要各种的数量的氧和氮。潜水员有一定数量的气缸。每个气缸都有重量和气体容量。潜水员为了完成他的工作需要特定数量的氧和氮。他完成工作所需气缸的总重的最低限度的是多少？

例如：潜水员有 5 个气缸。每行三个数字为：氧，氮的（升）量和气缸的重量：

3 36 120

10 25 129

5 50 250

1 45 130

4 20 119

如果潜水员需要 5 升的氧和 60 升的氮则总重最小为 249（1，2 或者 4，5 号气缸）。

你的任务就是计算潜水员为了完成他的工作需要的气缸的权重的最低值。

#### 【输入】

第一行有 2 整数  $m, n$  ( $1 \leq m \leq 21, 1 \leq n \leq 79$ )。它们表示氧，氮各自需要的量。

第二行为整数  $k$  ( $1 \leq k \leq 1000$ ) 表示气缸的个数。

此后的  $k$  行，每行包括  $a_i, b_i, c_i$  ( $1 \leq a_i \leq 21, 1 \leq b_i \leq 79, 1 \leq c_i \leq 800$ ) 3 整数。这些各自是：第  $i$  个气缸里的氧和氮的容量及汽缸重量。

#### 【输出】

仅一行包含一个整数，为潜水员完成工作所需的气缸的重量总和的最低值。

#### 【输入样例】

```
5 60
5
3 36 120
10 25 129
5 50 250
1 45 130
4 20 119
```

#### 【输出样例】

```
249
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
```



```

int m,n,k;
int a[N],b[N],c[N],f[N][N];
void TwoDimensionPack(int weight_1,int weight_2,int cost)
{
    for(int j=m;j>=0;j--)
        for(int k=n;k>=0;k--)
            {
                int u=j+weight_1;
                int v=k+weight_2;
                if(u>=m)
                    u=m;
                if(v>=n)
                    v=n;
                f[u][v]=min(f[u][v],f[j][k]+cost);
            }
}
int main()
{
    cin>>m>>n>>k;
    for(int i=1;i<=k;i++)
        cin>>a[i]>>b[i]>>c[i];

    memset(f,INF,sizeof(f));
    f[0][0]=0;

    for(int i=1;i<=k;i++)
        TwoDimensionPack(a[i],b[i],c[i]);
    cout<<f[m][n]<<endl;
    return 0;
}

```

### 1.1.2.9.2.6 分组背包

#### 分组背包

原创 Alex\_McAvoy 发布于 2018-06-03 21:09:34 阅读数 734 收藏  
展开

##### 【题目描述】

一个旅行者有一个最多能装  $V$  公斤的背包，现在有  $n$  件物品，它们的重量分别是  $W_1, W_2, \dots, W_n$ ，它们的价值分别为  $C_1, C_2, \dots, C_n$ 。这些物品被划分为若干组，每组中的物品互相冲突，最多选一件。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

**【输入】**

第一行：三个整数， $V$ (背包容量， $V \leq 200$ )， $N$ (物品数量， $N \leq 30$ )和  $T$ (最大组号， $T \leq 10$ )；

第  $2..N+1$  行：每行三个整数  $W_i, C_i, P_i$ ，表示每个物品的重量，价值，所属组号。

**【输出】**

仅一行，一个数，表示最大总价值。

**【输入样例】**

```
10 6 3
2 1 1
3 3 1
4 8 2
6 9 2
2 8 3
3 9 3
```

**【输出样例】**

```
20
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int v,n,t;
int group[N][N],w[N],c[N],f[N];
void GroupPack(int groupp,int num)
{
    for(int j=v;j>=0;j--)
        for(int k=1;k<=groupp;k++)
        {
            int q=group[num][k]; //物品序号
            if(j<=w[q])
                f[j]=max(f[j],f[j-w[q]]+c[q]);
        }
}
```

```

}
int main()
{
    int p;
    cin>>v>>n>>t;
    for(int i=1;i<=n;i++)
    {
        cin>>w[i]>>c[i]>>p;
        group[p][++group[p][0]]=i;//group[p][0]存储 p 组元素个数
    }

    for(int i=1;i<=t;i++)
        GroupPack(group[i][0],i);
    cout<<f[v]<<endl;
    return 0;
}

```

### 1.1.2.9.2.7 货币系统

#### 货币系统

原创 Alex\_McAvoy 发布于 2018-06-03 21:10:41 阅读数 604 收藏  
展开

##### 【题目描述】

给你一个  $n$  种面值的货币系统，求组成面值为  $m$  的货币有多少种方案。

##### 【输入】

第一行为  $n$  和  $m$ 。

##### 【输出】

一行，方案数。

##### 【输入样例】

```

3 10      //3 种面值组成面值为 10 的方案
1         //面值 1
2         //面值 2
5         //面值 5

```

##### 【输出样例】

```

10        //有 10 种方案

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>

```

```

#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 2520
#define E 1e-12
using namespace std;
long long a[N],f[N];
int main()
{
    int n,m;
    cin>>n>>m;
    f[0]=1;
    for(int i=1;i<=n;i++)
        cin>>a[i];
    for(int i=1;i<=n;i++)
        for(int j=a[i];j<=m;j++)
            f[j]+=f[j-a[i]];
    cout<<f[m]<<endl;
    return 0;
}

```

### 1.1.2.9.2.8 采药

#### 采药

原创 Alex\_McAvoy 发布于 2018-06-04 00:18:29 阅读数 532 收藏  
展开

#### 【题目描述】

辰辰是个很有潜能、天资聪颖的孩子，他的梦想是称为世界上最伟大的医师。为此，他想拜附近最有威望的医师为师。医师为了判断他的资质，给他出了一个难题。医师把他带到个到处都是草药的山洞里对他说：“孩子，这个山洞里有一些不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。我会给你一段时间，在这段时间里，你可以采到一些草药。如果你是一个聪明的孩子，你应该可以让采到的草药的总价值最大。”

如果你是辰辰，你能完成这个任务吗？

### 【输入】

输入的第一行有两个整数  $T$  ( $1 \leq T \leq 1000$ ) 和  $M$  ( $1 \leq M \leq 100$ )， $T$  代表总共能够用来采药的时间， $M$  代表山洞里的草药的数目。

接下来的  $M$  行每行包括两个在 1 到 100 之间（包括 1 和 100）的整数，分别表示采摘某株草药的时间和这株草药的价值。

### 【输出】

输出只包括一行，这一行只包含一个整数，表示在规定的时间内，可以采到的草药的最大总价值。

### 【输入样例】

```
70 3
71 100
69 1
1 2
```

### 【输出样例】

```
3
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 2520
#define E 1e-12
using namespace std;
int t,m;
int w[N],c[N],f[N];
void ZeroOnePack(int cost,int weight)
{
    for(int v=t;v>=weight;v--)
        f[v]=max(f[v],f[v-weight]+cost);
}

int main()
```

```

{
    cin>>t>>m;
    for(int i=1;i<=m;i++)
        cin>>w[i]>>c[i];
    for(int i=1;i<=m;i++)
        ZeroOnePack(c[i],w[i]);
    cout<<f[t]<<endl;
    return 0;
}

```

### 1.1.2.9.2.9 数字组合

#### 数字组合

原创 Alex\_McAvoy 发布于 2018-06-04 00:18:50 阅读数 1024 收藏  
展开

##### 【题目描述】

有  $n$  个正整数，找出其中和为  $t$  ( $t$  也是正整数) 的可能的组合方式。如：

$n=5$ , 5 个数分别为 1,2,3,4,5,  $t=5$ ;

那么可能的组合有  $5=1+4$  和  $5=2+3$  和  $5=5$  三种组合方式。

##### 【输入】

输入的第一行是两个正整数  $n$  和  $t$ ，用空格隔开，其中  $1 \leq n \leq 20$ , 表示正整数的个数， $t$  为要求的和 ( $1 \leq t \leq 1000$ );

接下来的一行是  $n$  个正整数，用空格隔开。

##### 【输出】

和为  $t$  的不同的组合方式的数目。

##### 【输入样例】

```

5 5
1 2 3 4 5

```

##### 【输出样例】

```

3

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>

```

```

#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int n,t;
int w[N];
long long f[N][N];
int main()
{
    cin>>n>>t;
    for(int i=1;i<=n;i++)
    {
        cin>>w[i];
        f[i][w[i]]=1;
    }

    for(int i=1;i<=n;i++)
        for(int j=1;j<=t;j++)
            if(j<w[i])
                f[i][j]+=f[i-1][j];
            else
                f[i][j]+=f[i-1][j]+f[i-1][j-w[i]];
    cout<<f[n][t]<<endl;
    return 0;
}

```

### 1.1.2.9.2.10 宠物小精灵之收服

#### 宠物小精灵之收服

原创 Alex\_McAvoy 发布于 2018-06-04 00:19:09 阅读数 1054 收藏  
展开

#### 【题目描述】

宠物小精灵是一部讲述小智和他的搭档皮卡丘一起冒险的故事。

一天，小智和皮卡丘来到了小精灵狩猎场，里面有很多珍贵的野生宠物小精灵。小智也想收服其中的一些小精灵。然而，野生的小精灵并不那么容易被收服。对于每一个野生小精灵而言，小智可能需要使用很多个精灵球才能收服它，而在收服过程中，野生小精灵也会对皮卡丘造成一定的伤害（从而减少皮卡丘的体力）。当皮卡丘的体力小于等于 0 时，小智就必须结束狩猎（因为他需要给皮卡丘疗伤），而使得皮卡丘体力小于等于 0 的野生小精灵也不会被小智收服。当小智的精灵球用完时，狩猎也宣告结束。

我们假设小智遇到野生小精灵时有两个选择：收服它，或者离开它。如果小智选择了收服，那么一定会扔出能够收服该小精灵的精灵球，而皮卡丘也一定会受到相应的伤害；如果选择离开它，那么小智不会损失精灵球，皮卡丘也不会损失体力。

小智的目标有两个：主要目标是收服尽可能多的野生小精灵；如果可以收服的小精灵数量一样，小智希望皮卡丘受到的伤害越小（剩余体力越大），因为他们还要继续冒险。

现在已知小智的精灵球数量和皮卡丘的初始体力，已知每一个小精灵需要的用于收服的精灵球数目和它在被收服过程中会对皮卡丘造成的伤害数目。请问，小智该如何选择收服哪些小精灵以达到他的目标呢？

#### 【输入】

输入数据的第一行包含三个整数：N( $0 < N < 1000$ )，M( $0 < M < 500$ )，K( $0 < K < 100$ )，分别代表小智的精灵球数量、皮卡丘初始的体力值、野生小精灵的数量。

之后的 K 行，每一行代表一个野生小精灵，包括两个整数：收服该小精灵需要的精灵球的数量，以及收服过程中对皮卡丘造成的伤害。

#### 【输出】

输出为一行，包含两个整数：C，R，分别表示最多收服 C 个小精灵，以及收服 C 个小精灵时皮卡丘的剩余体力值最多为 R。

#### 【输入样例】

```
10 100 5
7 10
2 40
2 50
1 20
4 20
```

#### 【输出样例】

```
3 30
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
```



```

#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int V,U,n;
int v[N],u[N],f[N][N];
void TwoDimensionPack(int weight_1,int weight_2)
{
    for(int j=V;j>=weight_1;j--)
        for(int k=U;k>=weight_2;k--)
            f[j][k]=max(f[j][k],f[j-weight_1][k-weight_2]+1);
}
int main()
{
    cin>>V>>U>>n;
    for(int i=1;i<=n;i++)
        cin>>v[i]>>u[i];

    for(int i=0;i<=V;i++)//边界设定
        for(int j=0;j<=U;j++)
            f[i][j]=0;

    for(int i=1;i<=n;i++)
        TwoDimensionPack(v[i],u[i]);

    int cnt=0;
    for(int i=1;i<=U;i++)
    {
        if(f[V][i]==f[V][U])
        {
            cnt=i;
            break;
        }
    }
    cout<<f[V][U]<<" "<<U-cnt<<endl;
}

```

```
    return 0;
}
```

## 1.1.2.9.2.11 买书

### 买书

原创 Alex\_McAvoy 发布于 2018-06-04 00:19:28 阅读数 797 收藏  
展开

#### 【题目描述】

小明手里有  $n$  元钱全部用来买书，书的价格为 10 元，20 元，50 元，100 元。

问小明有多少种买书方案？

#### 【输入】

第一行为  $n$  和  $m$ 。一个整数  $n$ ，代表总共钱数。 $(0 \leq n \leq 1000)$

#### 【输出】

一行，方案数。

#### 【输入样例】

20

#### 【输出样例】

2

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int V;
```

```

int w[4]={10,20,50,100};
int f[N];
int main()
{
    memset(f,0,sizeof(f));
    cin>>V;
    f[0]=1;
    for(int i=0;i<4;i++)
        for(int j=w[i];j<=V;j++)
            f[j]+=f[j-w[i]];
    cout<<f[V]<<endl;
    return 0;
}

```

点赞

### 1.1.2.9.2.12 Charm Bracelet

#### Charm Bracelet

原创 Alex\_McAvoy 发布于 2018-06-04 00:19:49 阅读数 493 收藏  
展开

##### 【题目描述】

有  $n$  个物品，编号为  $i$  的物品的重量为  $w[i]$ ，价值为  $c[i]$ ，现在要从这些物品中选一些物品装到一个容量为  $m$  的背包中，使得背包内物体在总重量不超过  $m$  的前提下价值尽量大。

##### 【输入】

第 1 行:两个整数， $n$ (物品数量,  $n \leq 3500$ )和  $m$ (背包容量,  $m \leq 12880$ )。

第 2.. $n+1$  行: 每行二个整数  $w[i]$ ， $c[i]$ ，表示每个物品的重量和价值。

##### 【输出】

仅一行，一个数，表示最大总价值。

##### 【输入样例】

```

4 6
1 4
2 6
3 12
2 7

```

##### 【输出样例】

23

【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 3501
#define MOD 2520
#define E 1e-12
using namespace std;
int n,m;
int w[N],c[N],f[N*5];
void ZeroOnePack(int cost,int weight)
{
    for(int v=m;v>=weight;v--)
        f[v]=max(f[v],f[v-weight]+cost);
}

int main()
{
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        cin>>w[i]>>c[i];
    for(int i=1;i<=n;i++)
        ZeroOnePack(c[i],w[i]);
    cout<<f[m]<<endl;
    return 0;
}

```

### 1.1.2.9.2.13 装箱问题

#### 装箱问题

原创 Alex\_McAvoy 发布于 2018-06-04 00:20:08 阅读数 838 收藏  
展开

**【题目描述】**

有一个箱子容量为  $V$ （正整数， $0 \leq V \leq 20000$ ），同时有  $n$  个物品（ $0 < n \leq 30$ ），每个物品有一个体积（正整数）。

要求  $n$  个物品中，任取若干个装入箱内，使箱子的剩余空间为最小。

**【输入】**

第一行是一个整数  $V$ ，表示箱子容量。

第二行是一个整数  $n$ ，表示物品数。

接下来  $n$  行，每行一个正整数（不超过 10000），分别表示这  $n$  个物品的各自体积。

**【输出】**

一个整数，表示箱子剩余空间。

**【输入样例】**

```
24
6
8
3
12
7
9
7
```

**【输出样例】**

```
0
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 50
#define MOD 2520
#define E 1e-12
```

```

using namespace std;
int n,V;
int w[50],f[20000];
void ZeroOnePack(int weight)
{
    for(int v=V;v>=weight;v--)
        f[v]=max(f[v],f[v-weight]+weight);
}

int main()
{
    cin>>V>>n;
    for(int i=1;i<=n;i++)
        cin>>w[i];
    for(int i=1;i<=n;i++)
        ZeroOnePack(w[i]);
    cout<<V-f[V]<<endl;
    return 0;
}

```

## 1.1.2.9.2.14 开餐馆

### 开餐馆

原创 Alex\_McAvoy 发布于 2018-06-04 00:20:24 阅读数 978 收藏  
展开

#### 【题目描述】

信息学院的同学小明毕业之后打算创业开餐馆.现在共有  $n$  个地点可供选择.小明打算从中选择合适的位置开设一些餐馆.这  $n$  个地点排列在同一条直线上.我们用一个整数序列  $m_1, m_2, \dots, m_n$  来表示他们的相对位置.由于地段关系,开餐馆的利润会有所不同.我们用  $p_i$  表示在  $m_i$  处开餐馆的利润.为了避免自己的餐馆的内部竞争,餐馆之间的距离必须大于  $k$ .请你帮助小明选择一个总利润最大的方案。

#### 【输入】

输入第一行是整数  $T(1 \leq T \leq 1000)$ , 表明有  $T$  组测试数据.紧接着有  $T$  组连续的测试.每组测试数据有 3 行。

第 1 行:地点总数  $n(n < 100)$ , 距离限制  $k(k > 0$  且  $k < 1000)$ ;

第 2 行: $n$  个地点的位置  $m_1, m_2, \dots, m_n(1000000 > m_i > 0$  且为整数,升序排列);

第 3 行:  $n$  个地点的餐馆利润  $p_1, p_2, \dots, p_n$  ( $1000 > p_i > 0$  且为整数)。

**【输出】**

对于每组测试数据可能的最大利润。

**【输入样例】**

```
2
3 11
1 2 15
10 2 30
3 16
1 2 15
10 2 30
```

**【输出样例】**

```
40
30
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int w[N],c[N],f[N];
int main()
{
    int t;
    cin>>t;
    while(t-->0)
    {
        int n,k;
        cin>>n>>k;
        for(int i=1;i<=n;i++)
            cin>>w[i];
        for(int i=1;i<=n;i++)
        {
```

```

        cin>>c[i];
        f[i]=c[i];
    }

    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(w[i]-w[j]>k)
                f[i]=max(f[i],f[j]+c[i]);

    int maxx=-INF;
    for(int i=1;i<=n;i++)
        maxx=max(f[i],maxx);
    cout<<maxx<<endl;
}
return 0;
}

```

### 1.1.2.9.3 第三节 动态规划经典问题

#### 1.1.2.9.3.1 合并石子

##### 合并石子

原创 Alex\_McAvoy 发布于 2018-06-04 21:58:10 阅读数 1129 收藏  
展开

##### 【题目描述】

在一个操场上一排地摆放着  $N$  堆石子。现要将石子有次序地合并成一堆。规定每次只能选相邻的 2 堆石子合并成新的一堆，并将新的一堆石子数记为该次合并的得分。

计算出将  $N$  堆石子合并成一堆的最小得分。

##### 【输入】

第一行为一个正整数  $N$  ( $2 \leq N \leq 100$ );

以下  $N$  行,每行一个正整数, 小于 10000, 分别表示第  $i$  堆石子的个数( $1 \leq i \leq N$ )。

##### 【输出】

一个正整数, 即最小得分。



**【输入样例】**

```

7
13
7
8
16
21
4
18

```

**【输出样例】**

```

269

```

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int sum[N],f[N][N];
int main()
{
    int n,a;
    memset(f,INF,sizeof(f));

    cin>>n;
    sum[0]=0;
    for(int i=1;i<=n;i++)
    {
        cin>>a;
        sum[i]=sum[i-1]+a;
    }
    for(int i=1;i<=n;i++)
        f[i][i]=0;
    for(int i=n;i>=1;i--)
        for(int j=i+1;j<=n;j++)

```

```

    for(int k=i;k<=j-1;k++)
        f[i][j]=min(f[i][j],f[i][k]+f[k+1][j]+sum[j]-sum[i-1]);
    cout<<f[1][n]<<endl;
    return 0;
}

```

### 1.1.2.9.3.2 乘积最大

#### 乘积最大

原创 Alex\_McAvoy 发布于 2018-06-04 22:24:52 阅读数 660 收藏  
展开

##### 【题目描述】

今年是国际数学联盟确定的“2000——世界数学家年”，又恰逢我国著名数学家华罗庚先生诞辰 90 周年。在华罗庚先生的家乡江苏金坛，组织了一场别开生面的数学智力竞赛的活动，你的一个好朋友 XZ 也有幸得以参加。活动中，主持人给所有参加活动的选手出了这样一道题目：

设有一个长度为  $N$  的数字串，要求选手使用  $K$  个乘号将它分成  $K+1$  个部分，找出一种分法，使得这  $K+1$  个部分的乘积最大。

同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：

有一个数字串：312，当  $N=3$ ， $K=1$  时会有以下两种分法：

1)  $3*12=36$

2)  $31*2=62$

这时，符合题目要求的结果是： $31*2=62$ 。

现在，请你帮助你的好朋友 XZ 设计一个程序，求得正确的答案。

##### 【输入】

第一行共有 2 个自然数  $N$ ， $K$  ( $6 \leq N \leq 10$ ， $1 \leq K \leq 6$ )

第二行是一个长度为  $N$  的数字串。

**【输出】**

输出所求得的最大乘积（一个自然数）。

**【输入样例】**

4 2

1231

**【输出样例】**

62

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 30
#define MOD 2520
#define E 1e-12
using namespace std;
char a[N];
int f[N][N];
int cut(int left,int right)
{
    int ans=0;
    for(int i=left;i<=right;i++)
    {
        ans*=10;
        ans+=a[i]-'0';
    }
    return ans;
}
int main()
{
    int n,k;

    cin>>n>>k;
    scanf("%s",&a[1]);

    for(int i=1;i<=n;i++)
        f[i][0]=cut(1,i);
```

```

for(int l=2;l<=n;l++)
    for(int i=1;i<=min(l-1,k);i++)
        for(int j=i;j<l;j++)
            f[l][i]=max(f[l][i],f[j][i-1]*cut(j+1,l));

cout<<f[n][k]<<endl;
return 0;
}

```

点赞

### 1.1.2.9.3.3 编辑距离

#### 编辑距离

原创 Alex\_McAvoy 发布于 2018-06-04 22:27:32 阅读数 638 收藏  
展开

##### 【题目描述】

设 **A** 和 **B** 是两个字符串。我们要用最少的字符操作次数，将字符串 **A** 转换为字符串 **B**。这里所说的字符操作共有三种：

- 1、删除一个字符；
- 2、插入一个字符；
- 3、将一个字符改为另一个字符。

对任意的两个字符串 **A** 和 **B**，计算出将字符串 **A** 变换为字符串 **B** 所用的最少字符操作次数。

##### 【输入】

第一行为字符串 **A**；第二行为字符串 **B**；字符串 **A** 和 **B** 的长度均小于 2000。

##### 【输出】

只有一个正整数，为最少字符操作次数。

##### 【输入样例】

sfdqxbw

gfdgw

【输出样例】

4

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 3001
#define MOD 2520
#define E 1e-12
using namespace std;
char a[N],b[N];
int f[N][N];
int main()
{
    scanf("%s%s",a+1,b+1);
    int n=strlen(a+1);
    int m=strlen(b+1);

    for(int i=1;i<=n;i++)
        f[i][0]=i;
    for(int i=1;i<=m;i++)
        f[0][i]=i;

    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            if(a[i]==b[j])
                f[i][j]=f[i-1][j-1];
            else
                f[i][j]=min(f[i-1][j],min(f[i][j-1],f[i-1][j-1]))+1;

    cout<<f[n][m]<<endl;

    return 0;
}
```

### 1.1.2.9.3.4 方格取数

#### 方格取数

原创 Alex\_McAvoy 发布于 2018-06-05 23:16:47 阅读数 612 收藏  
展开

##### 【题目描述】

设有  $N \times N$  的方格图，我们在其中的某些方格中填入正整数，而其它的方格中则放入数字 0。如下图所示：

A	0	0	0	0	0	0	0	
	0	0	13	0	0	6	0	0
	0	0	0	0	7	0	0	0
	0	0	0	14	0	0	0	0
	0	21	0	0	0	4	0	0
	0	0	15	0	0	0	0	0
	0	14	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
								B

某人从图中的左上角 A 出发，可以向下行走，也可以向右行走，直到到达右下角的 B 点。在走过的路上，他可以取走方格中的数（取走后的方格中将变为数字 0）。

此人从 A 点到 B 点共走了两次，试找出两条这样的路径，使得取得的数字和为最大。

##### 【输入】

第一行为一个整数  $N$  ( $N \leq 10$ )，表示  $N \times N$  的方格图。

接下来的每行有三个整数，第一个为行号数，第二个为列号数，第三个为在该行、该列上所放的数。一行“0 0 0”表示结束。

##### 【输出】

第一个整数，表示两条路径上取得的最大的和。

##### 【输入样例】

```
8
2 3 13
2 6 6
3 5 7
4 4 14
```

5 2 21

5 6 4

6 3 15

7 2 14

0 0 0

【输出样例】

67

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int g[N][N],f[N][N][N][N];
int main()
{
    int n;
    int a,b,c;
    cin>>n;;
    while(scanf("%d%d%d",&a,&b,&c)!=EOF&&a&&b&&c)
        g[a][b]=c;

    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            for(int k=1;k<=n;k++)
                for(int l=1;l<=n;l++)
                {
                    int temp1=max(f[i-1][j][k-1][l],f[i-1][j][k][l-1]);
                    int temp2=max(f[i][j-1][k-1][l],f[i][j-1][k][l-1]);
                    f[i][j][k][l]=max(temp1,temp2)+g[i][j];
                    if(i!=k&&j!=l)
                        f[i][j][k][l]+=g[k][l];
                }
}
```

```

    cout<<f[n][n][n]<<endl;
    return 0;
}

```

### 1.1.2.9.3.5 复制书稿

#### 复制书稿

原创 Alex\_McAvoy 发布于 2018-06-05 23:24:18 阅读数 637 收藏

展开

##### 【题目描述】

现在要把  $m$  本有顺序的书分给  $k$  个人复制（抄写），每一个人的抄写速度都一样，一本书不允许给两个（或以上）的人抄写，分给每一个人的书，必须是连续的，比如不能把第一、第三和第四本书给同一个人抄写。

现在请你设计一种方案，使得复制时间最短。复制时间为抄写页数最多的人用去的时间。

##### 【输入】

第一行两个整数  $m, k$ ; ( $k \leq m \leq 500$ )

第二行  $m$  个整数，第  $i$  个整数表示第  $i$  本书的页数。

##### 【输出】

共  $k$  行，每行两个整数，第  $i$  行表示第  $i$  个人抄写的书的起始编号和终止编号。 $k$  行的起始编号应该从小到大排列，如果有多解，则尽可能让前面的人少抄写。

##### 【输入样例】

```

9 3
1 2 3 4 5 6 7 8 9

```

##### 【输出样例】

```

1 5
6 7
8 9

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>

```



```

#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N],b[N],c[N][N];
int sum[N],f[N][N];
int main()
{
    int m,n;
    cin>>m>>n;
    for(int i=1;i<=m;i++)
    {
        cin>>a[i];
        sum[i]=sum[i-1]+a[i];
    }

    sum[0]=0;
    memset(f,INF,sizeof(f));
    for(int i=1;i<=m;i++)
        f[1][i]=sum[i];

    for(int i=2;i<=n;i++)
        for(int j=1;j<=m;j++)
            for(int k=i-1;k<j;k++)
                f[i][j]=min(f[i][j],max(f[i-1][k],sum[j]-sum[k]));

    int maxx=f[n][n];
    int temp=n;
    for(int i=m;i>0;i--)
    {
        if(a[i]+b[temp]>maxx)
            temp--;
        b[temp]+=a[i];
        c[temp][++c[temp][0]]=i;
    }

    for(int i=1;i<=n;i++)
        cout<<c[i][c[i][0]]<<" "<<c[i][1]<<endl;
    return 0;
}

```

### 1.1.2.9.3.6 橱窗布置

#### 橱窗布置

原创 Alex\_McAvoy 发布于 2018-06-05 23:24:39 阅读数 448 收藏  
展开

##### 【题目描述】

假设以最美观的方式布置花店的橱窗，有  $F$  束花，每束花的品种都不一样，同时，至少有同样数量的花瓶，被按顺序摆成一行，花瓶的位置是固定的，并从左到右，从 1 到  $V$  顺序编号， $V$  是花瓶的数目，编号为 1 的花瓶在最左边，编号为  $V$  的花瓶在最右边，花束可以移动，并且每束花用 1 到  $F$  的整数惟一标识，标识花束的整数决定了花束在花瓶中列的顺序即如果  $i < j$ ，则花束  $i$  必须放在花束  $j$  左边的花瓶中。

例如，假设杜鹃花的标识数为 1，秋海棠的标识数为 2，康乃馨的标识数为 3，所有的花束在放入花瓶时必须保持其标识数的顺序，即：杜鹃花必须放在秋海棠左边的花瓶中，秋海棠必须放在康乃馨左边的花瓶中。如果花瓶的数目大于花束的数目，则多余的花瓶必须空，即每个花瓶中只能放一束花。

每一个花瓶的形状和颜色也不相同，因此，当各个花瓶中放入不同的花束时会产生不同的美学效果，并以美学值(一个整数)来表示，空置花瓶的美学值为 0。在上述例子中，花瓶与花束的不同搭配所具有的美学值，可以用如下表格表示。

	花瓶 1	花瓶 2	花瓶 3	花瓶 4	花瓶 5
杜鹃花	7	23	-5	-24	16
秋海棠	5	21	-4	10	23
康乃馨	-21	5	-4	-20	20

根据表格，杜鹃花放在花瓶 2 中，会显得非常好看，但若放在花瓶 4 中则显得很难看。

假设条件：

$1 \leq F \leq 100$ ，其中  $F$  为花束的数量，花束编号从 1 至  $F$ 。

$F \leq V \leq 100$ ，其中  $V$  是花瓶的数量。

$-50 \leq A_{ij} \leq 50$ ，其中  $A_{ij}$  是花束  $i$  在花瓶  $j$  中的美学值。

输入整数  $F$ ， $V$  和矩阵  $(A_{ij})$ ，输出最大美学值和每束花摆放在各个花瓶中的花瓶编号。

##### 【输入】

第一行包含两个数： $F$ ， $V$ 。

随后的  $F$  行中，每行包含  $V$  个整数， $A_{ij}$  即为输入文件中第  $(i+1)$  行中的第  $j$  个数。

### 【输出】

第一行是程序所产生摆放方式的美学值。

第二行必须用  $F$  个数表示摆放方式，即该行的第  $K$  个数表示花束  $K$  所在的花瓶的编号。

### 【输入样例】

```
3 5
7 23 -5 -24 16
5 21 -4 10 23
-21 5 -4 -20 20
```

### 【输出样例】

```
53
2 4 5
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],b[N][N],c[N][N],d[N];
int main()
{
    int f,v;
    cin>>f>>v;
    for(int i=1;i<=f;i++)
        for(int j=1;j<=v;j++)
            cin>>a[i][j];

    memset(b,128,sizeof(b));
    for(int i=1;i<=v-(f-1);i++)
        b[1][i]=a[1][i];
    for(int i=1;i<=f;i++)
```

```

for(int j=i;j<=v-(f-i);j++)
    for(int k=i-1;k<=j-1;k++)
        if(b[i][j]<b[i-1][k]+a[i][j])
        {
            b[i][j]=b[i-1][k]+a[i][j];
            c[i][j]=k;
        }

int maxx=-INF;
int temp;
for(int i=f;i<=v;i++)
    if(b[f][i]>maxx)
    {
        maxx=b[f][i];
        temp=i;
    }
cout<<maxx<<endl;

for(int i=f;i>=1;i--)
{
    d[f-(i-1)]=temp;
    temp=c[i][temp];
}

for(int i=f;i>=1;i--)
    cout<<d[i]<<" ";
return 0;
}

```

点赞

### 1.1.2.9.3.7 滑雪（信息学奥赛一本通-T1280）

#### 滑雪（信息学奥赛一本通-T1280）

原创 Alex\_McAvoy 发布于 2018-06-05 23:25:39 阅读数 559 收藏  
展开

##### 【题目描述】

小明喜欢滑雪，因为滑雪的确很刺激，可是为了获得速度，滑的区域必须向下倾斜，当小明滑到坡底，不得不再次走上坡或等着直升机来载他，小明想知道在一个区域中

最长的滑坡。滑坡的长度由滑过点的个数来计算，区域由一个二维数组给出，数组的每个数字代表点的高度。下面是一个例子：

11615141321724231231825221141920211056789

一个人可以从某个点滑向上下左右相邻四个点之一，当且仅当高度减小，在上面的例子中，一条可行的滑坡为 25-24-17-16-1（从 25 开始到 1 结束），当然 25-24.....2-1 更长，事实上这是最长的一条。

#### 【输入】

输入的第一行为表示区域的二维数组的行数  $R$  和列数  $C$  ( $1 \leq R, C \leq 100$ )，下面是  $R$  行，每行有  $C$  个数代表高度。

#### 【输出】

输出区域中最长的滑坡长度。

#### 【输入样例】

```
5 5
1 2 3 4 5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9
```

#### 【输出样例】

```
25
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N][N],f[N][N];
int r,c;
int dir[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
```

```

int dfs(int x,int y,int step)
{
    int temp=1;

    if(f[x][y]>0)
        return f[x][y];

    for(int i=0;i<4;i++)
    {
        int nx=x+dir[i][0];
        int ny=y+dir[i][1];
        if(nx>=1&&nx<=r&&ny>=1&&ny<=c&&a[x][y]>a[nx][ny])
            temp=max(temp,dfs(nx,ny,step+1)+1);
    }
    f[x][y]=temp;
    return temp;
}
int main()
{
    cin>>r>>c;
    for(int i=1;i<=r;i++)
        for(int j=1;j<=c;j++)
            cin>>a[i][j];

    int maxx=-INF;
    for(int i=1;i<=r;i++)
        for(int j=1;j<=c;j++)
            maxx=max(maxx,dfs(i,j,1));

    cout<<maxx<<endl;
    return 0;
}

```

### 1.1.2.9.3.8 公共子序列

#### 公共子序列

原创 Alex\_McAvoy 发布于 2018-06-05 23:25:57 阅读数 600 收藏  
展开

#### 【题目描述】

我们称序列  $Z=\langle z_1,z_2,\dots,z_k \rangle$  是序列  $X=\langle x_1,x_2,\dots,x_m \rangle$  的子序列当且仅当存在严格上

升的序列 $\langle i_1, i_2, \dots, i_k \rangle$ , 使得对  $j=1, 2, \dots, k$ , 有  $x_{i_j} = z_j$ 。比如  $Z = \langle a, b, f, c \rangle$  是  $X = \langle a, b, c, f, b, c \rangle$  的子序列。

现在给出两个序列  $X$  和  $Y$ , 你的任务是找到  $X$  和  $Y$  的最大公共子序列, 也就是说要找到一个最长的序列  $Z$ , 使得  $Z$  既是  $X$  的子序列也是  $Y$  的子序列。

#### 【输入】

输入包括多组测试数据。每组数据包括一行, 给出两个长度不超过 200 的字符串, 表示两个序列。两个字符串之间由若干个空格隔开。

#### 【输出】

对每组输入数据, 输出一行, 给出两个序列的最大公共子序列的长度。

#### 【输入样例】

```
abcfbc abfcab
programming contest
abcd mnp
```

#### 【输出样例】

```
4
2
0
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 2520
#define E 1e-12
using namespace std;
char a[N],b[N];
int f[N][N];
int main()
{
    while(scanf("%s%s",a+1,b+1)!=EOF)
    {
        int len_a=strlen(a+1);
        int len_b=strlen(b+1);
```

```

for(int i=1;i<=len_a;i++)
    f[i][0]=0;
for(int j=1;j<=len_b;j++)
    f[0][j]=0;

for(int i=1;i<=len_a;i++)
    for(int j=1;j<=len_b;j++)
    {
        if(a[i]==b[j])
            f[i][j]=f[i-1][j-1]+1;
        else
            f[i][j]=max(f[i-1][j],f[i][j-1]);
    }

cout<<f[len_a][len_b]<<endl;
}
return 0;
}

```

### 1.1.2.9.3.9 计算字符串距离

#### 计算字符串距离

原创 Alex\_McAvoy 发布于 2018-06-05 23:27:03 阅读数 365 收藏  
展开

##### 【题目描述】

对于两个不同的字符串，我们有一套操作方法来把他们变得相同，具体方法为：

修改一个字符（如把“a”替换为“b”）；

删除一个字符（如把“traveling”变为“travelng”）。

比如对于“abcdefg”和“abcdef”两个字符串来说，我们认为可以通过增加/减少一个“g”的方式来达到目的。无论增加还是减少“g”，我们都仅仅需要一次操作。我们把这个操作所需要的次数定义为两个字符串的距离。

给定任意两个字符串，写出一个算法来计算出他们的距离。



**【输入】**

第一行有一个整数  $n$ 。表示测试数据的组数。

接下来共  $n$  行，每行两个字符串，用空格隔开，表示要计算距离的两个字符串。

字符串长度不超过 1000。

**【输出】**

针对每一组测试数据输出一个整数，值为两个字符串的距离。

**【输入样例】**

```
3
abcdefg abcdef
ab ab
mnlkj jlknm
```

**【输出样例】**

```
1
0
4
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
char a[N],b[N];
int f[N][N];
int main()
{
    int n;
    cin>>n;
    while(n--)
    {
        scanf("%s%s",a+1,b+1);
        int len_a=strlen(a+1);
```

```

int len_b=strlen(b+1);

for(int i=0;i<=len_a;i++)
    f[i][0]=i;
for(int j=0;j<=len_b;j++)
    f[0][j]=j;
for(int i=1;i<=len_a;i++)
    for(int j=1;j<=len_b;j++)
    {
        if(a[i]==b[j])
            f[i][j]=f[i-1][j-1];
        else
            f[i][j]=min(min(f[i-1][j-1],f[i-1][j]),f[i][j-1])+1;
    }
cout<<f[len_a][len_b]<<endl;
}
return 0;
}

```

### 1.1.2.9.3.10 糖果

#### 糖果

原创 Alex\_McAvoy 发布于 2018-06-05 23:27:42 阅读数 867 收藏  
展开

##### 【题目描述】

由于在维护世界和平的事务中做出巨大贡献，Dzx 被赠予糖果公司 2010 年 5 月 23 日当天无限量糖果免费优惠券。在这一天，Dzx 可以从糖果公司的  $N$  件产品中任意选择若干件带回家享用。糖果公司的  $N$  件产品每件都包含数量不同的糖果。Dzx 希望他选择的产品包含的糖果总数是  $K$  的整数倍，这样他才能平均地将糖果分给帮助他维护世界和平的伙伴们。当然，在满足这一条件的基础上，糖果总数越多越好。Dzx 最多能带走多少糖果呢？

注意：Dzx 只能将糖果公司的产品整件带走。

##### 【输入】

第一行包含两个整数  $N(1 \leq N \leq 100)$  和  $K(1 \leq K \leq 100)$ 。

以下  $N$  行每行 1 个整数，表示糖果公司该件产品中包含的糖果数目，不超过 1000000。

##### 【输出】

符合要求的最多能达到的糖果总数，如果不能达到  $K$  的倍数这一要求，输出 0。

## 【输入样例】

5 7

1

2

3

4

5

## 【输出样例】

14

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int a[N];
int f[N][N];
int main()
{
    int n,k;
    cin>>n>>k;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    for(int i=1;i<k;i++)
        f[0][i]=-INF;
    for(int i=1;i<=n;i++)
        for(int j=0;j<k;j++)
            f[i][j]=max(f[i-1][j],f[i-1][(k+j-a[i]%k)%k]+a[i]);

    cout<<f[n][0]<<endl;
    return 0;
}

```

### 1.1.2.9.3.11 T1300 鸡蛋的硬度

#### T1300 鸡蛋的硬度

原创 Alex\_McAvoy 发布于 2018-06-05 23:28:03 阅读数 837 收藏  
展开

##### 【题目描述】

最近 XX 公司举办了一个奇怪的比赛：鸡蛋硬度之王争霸赛。参赛者是来自世界各地的母鸡，比赛的内容是看谁下的蛋最硬，更奇怪的是 XX 公司并不使用什么精密仪器来测量蛋的硬度，他们采用了一种最老土的办法--从高度扔鸡蛋--来测试鸡蛋的硬度，如果一次母鸡下的蛋从高楼的第  $a$  层摔下来没摔破，但是从  $a+1$  层摔下来时摔破了，那么就说这只母鸡的鸡蛋的硬度是  $a$ 。你当然可以找出各种理由说明这种方法不科学，比如同一只母鸡下的蛋硬度可能不一样等等，但是这不影响 XX 公司的争霸赛，因为他们只是为了吸引大家的眼球，一个个鸡蛋从 100 层的高楼上掉下来的时候，这情景还是能吸引很多人驻足观看的，当然，XX 公司也绝不会忘记在高楼上挂一条幅，写上“XX 公司”的字样--这比赛不过是 XX 公司的一个另类广告而已。

勤于思考的小 A 总是能从一件事情中发现一个数学问题，这件事也不例外。“假如有很多同样硬度的鸡蛋，那么我可以用二分的办法用最少的次数测出鸡蛋的硬度”，小 A 对自己的这个结论感到很满意，不过很快麻烦来了，“但是，假如我的鸡蛋不够用呢，比如我只有 1 个鸡蛋，那么我就不得不从第 1 层楼开始一层一层的扔，最坏情况下我要扔 100 次。如果有 2 个鸡蛋，那么就从 2 层楼开始的地方扔.....等等，不对，好像应该从  $1/3$  的地方开始扔才对，嗯，好像也不一定啊.....3 个鸡蛋怎么办，4 个，5 个，更多呢.....”，和往常一样，小 A 又陷入了一个思维僵局，与其说他是勤于思考，不如说他是喜欢自找麻烦。

好吧，既然麻烦来了，就得有人去解决，小 A 的麻烦就靠你来解决了吧！

##### 【输入】

输入包括多组数据，每组数据一行，包含两个正整数  $n$  和  $m(1 \leq n \leq 100, 1 \leq m \leq 10)$ ，其中  $n$  表示楼的高度， $m$  表示你现在拥有的鸡蛋个数，这些鸡蛋硬度相同（即它们从同样高的地方掉下来要么都摔碎要么都不碎），并且小于等于  $n$ 。你可以假定硬度为  $x$  的鸡蛋从高度小于等于  $x$  的地方摔无论如何都不会碎（没摔碎的鸡蛋可以继续使用），而只要从比  $x$  高的地方扔必然会碎。

对每组输入数据，你可以假定鸡蛋的硬度在 0 至  $n$  之间，即在  $n+1$  层扔鸡蛋一定会碎。

##### 【输出】

对于每一组输入，输出一个整数，表示使用最优策略在最坏情况下所需要的扔鸡蛋次数。

## 【输入样例】

100 1

100 2

## 【输出样例】

100

14

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 2520
#define E 1e-12
using namespace std;
int f[110][20];
int main()
{
    int n,m;
    while(scanf("%d%d",&n,&m)!=EOF)
    {
        for(int i=1;i<=n;i++)
            for(int j=1;j<=m;j++)
                f[i][j]=i;

        for(int i=1;i<=n;i++)
            for(int j=1;j<=i;j++)
                for(int k=2;k<=m;k++)
                    f[i][k]=min(f[i][k],max(f[j-1][k-1],f[i-j][k])+1);

        cout<<f[n][m]<<endl;
    }
    return 0;
}

```

### 1.1.2.9.3.12 大盗阿福

#### 大盗阿福

原创 Alex\_McAvoy 发布于 2018-06-07 15:13:25 阅读数 644 收藏  
展开

##### 【题目描述】

阿福是一名经验丰富的大盗。趁着月黑风高，阿福打算今晚洗劫一条街上的店铺。

这条街上一共有  $N$  家店铺，每家店中都有一些现金。阿福事先调查得知，只有当他同时洗劫了两家相邻的店铺时，街上的报警系统才会启动，然后警察就会蜂拥而至。

作为一向谨慎作案的大盗，阿福不愿意冒着被警察追捕的风险行窃。他想知道，在不惊动警察的情况下，他今晚最多可以得到多少现金？

##### 【输入】

输入的第一行是一个整数  $T(T \leq 50)$ ，表示一共有  $T$  组数据。

接下来的每组数据，第一行是一个整数  $N(1 \leq N \leq 100,000)$ ，表示一共有  $N$  家店铺。  
第二行是  $N$  个被空格分开的正整数，表示每一家店铺中的现金数量。每家店铺中的现金数量均不超过 1000。

##### 【输出】

对于每组数据，输出一行。该行包含一个整数，表示阿福在不惊动警察的情况下可以得到的现金数量。

##### 【输入样例】

```
2
3
1 8 2
4
10 7 6 14
```

##### 【输出样例】

```
8
24
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
```

```

#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 100001
#define E 1e-12
using namespace std;
int a[N],f[N];
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n;
        cin>>n;
        for(int i=1;i<=n;i++)
            cin>>a[i];
        f[0]=0;
        f[1]=a[1];
        for(int i=2;i<=n;i++)
            f[i]=max(f[i-1],f[i-2]+a[i]);
        cout<<f[n]<<endl;
    }
    return 0;
}

```

### 1.1.2.9.3.13 股票买卖

#### 股票买卖

原创 Alex\_McAvoy 发布于 2018-06-07 15:13:44 阅读数 890 收藏  
展开

##### 【题目描述】

最近越来越多的人都投身股市，阿福也有点心动了。谨记着“股市有风险，入市需谨慎”，阿福决定先来研究一下简化版的股票买卖问题。

假设阿福已经准确预测出了某只股票在未来  $N$  天的价格，他希望买卖两次，使得获得的利润最高。为了计算简单起见，利润的计算方式为卖出的价格减去买入的价格。

同一天可以进行多次买卖。但是在第一次买入之后，必须要先卖出，然后才可以第二次买入。

现在，阿福想知道他最多可以获得多少利润。

### 【输入】

输入的第一行是一个整数  $T(T \leq 50)$ ，表示一共有  $T$  组数据。

接下来的每组数据，第一行是一个整数  $N(1 \leq N \leq 100,000)$ ，表示一共有  $N$  天。第二行是  $N$  个被空格分开的整数，表示每天该股票的价格。该股票每天的价格的绝对值均不会超过 1,000,000。

### 【输出】

对于每组数据，输出一行。该行包含一个整数，表示阿福能够获得的最大的利润。

### 【输入样例】

```
3
7
5 14 -2 4 9 3 17
6
6 8 7 4 1 -2
4
18 9 5 2
```

### 【输出样例】

```
28
2
0
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
```



```

#define N 100001
#define MOD 100001
#define E 1e-12
using namespace std;
int a[N],f1[N],f2[N];
int main()
{
    int t,ans;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        for(int i=1;i<=n;i++)
            scanf("%d",&a[i]);

        int minn=INF,maxx=-INF;
        f1[0]=0;
        f2[n+1]=0;
        for(int i=1;i<=n;i++)
        {
            minn=min(minn,a[i]);
            f1[i]=max(f1[i-1],a[i]-minn);
        }
        for(int i=n;i>=1;i--)
        {
            maxx=max(maxx,a[i]);
            f2[i]=max(f2[n+1],maxx-a[i]);
        }

        int ans=-INF;
        for(int i=1;i<=n;i++)
            ans=max(ans,f1[i]+f2[i]);
        printf("%d\n",ans);
    }
    return 0;
}

```

点赞

### 1.1.2.9.3.14 鸣人的影分身

鸣人的影分身

原创 Alex\_McAvoy 发布于 2018-06-07 15:14:04 阅读数 923 收藏  
展开

### 【题目描述】

在火影忍者的世界里，令敌人捉摸不透是非常关键的。我们的主角漩涡鸣人所拥有的一个招数——多重影分身之术——就是一个很好的例子。

影分身是由鸣人身体的查克拉能量制造的，使用的查克拉越多，制造出的影分身越强。

针对不同的作战情况，鸣人可以选择制造出各种强度的影分身，有的用来佯攻，有的用来发起致命一击。

那么问题来了，假设鸣人的查克拉能量为  $M$ ，他影分身的个数最多为  $N$ ，那么制造影分身时有多少种（用  $K$  表示）不同的分配方法？（影分身可以被分配到 0 点查克拉能量）

### 【输入】

第一行是测试数据的数目  $t$  ( $0 \leq t \leq 20$ )。以下每行均包含二个整数  $M$  和  $N$  ( $1 \leq M, N \leq 10$ )，以空格分开。

### 【输出】

对输入的每组数据  $M$  和  $N$ ，用一行输出相应的  $K$ 。

### 【输入样例】

```
1
7 3
```

### 【输出样例】

```
8
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 201
#define MOD 100001
#define E 1e-12
using namespace std;
```

```

int f(int m,int n)
{
    if(m==0||n==1)
        return 1;
    if(m<n)
        return f(m,m);
    return
        f(m,n-1)+f(m-n,n);
}
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int m,n;
        cin>>m>>n;
        cout<<f(m,n)<<endl;
    }
    return 0;
}

```

### 1.1.2.9.3.15 数的划分

#### 数的划分

原创 Alex\_McAvoy 发布于 2018-06-07 15:14:25 阅读数 1172 收藏  
展开

##### 【题目描述】

将整数  $n$  分成  $k$  份，且每份不能为空，任意两份不能相同(不考虑顺序)。

例如： $n=7$ ， $k=3$ ，下面三种分法被认为是相同的。

1, 1, 5;    1, 5, 1;    5, 1, 1;

问有多少种不同的分法。 输出一个整数，即不同的分法。

**【输入】**

两个整数  $n, k$  ( $6 < n \leq 200, 2 \leq k \leq 6$ ), 中间用单个空格隔开。

**【输出】**

一个整数, 即不同的分法。

**【输入样例】**

7 3

**【输出样例】**

4

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 201
#define MOD 100001
#define E 1e-12
using namespace std;
int f[N][N];
int main()
{
    int n,k;
    cin>>n>>k;
    f[0][0]=1;
    for(int i=1;i<=k;i++)
        for(int j=i;j<=n;j++)
            f[i][j]=f[i][j-i]+f[i-1][j-1];
    cout<<f[k][n]<<endl;
    return 0;
}
```

### 1.1.2.9.3.16 Maximum sum

#### Maximum sum

原创 Alex\_McAvoy 发布于 2018-06-07 15:14:40 阅读数 770 收藏  
展开

### 【题目描述】

对于给定的整数序列  $A=\{a_1,a_2,\dots,a_n\}$ ，找出两个不重合连续子段，使得两子段中所有数字的和最大。我们如下定义函数  $d(A)$ ：

$$d(A) = \max_{1 \leq s_1 \leq t_1 \leq s_2 \leq t_2 \leq n} \left\{ \sum_{i=s_1}^{t_1} a_i + \sum_{j=s_2}^{t_2} a_j \right\}$$

我们的目标就是求出  $d(A)$ 。

### 【输入】

第一行是一个整数  $T(\leq 30)$ ，代表一共有多少组数据。

接下来是  $T$  组数据。

每组数据的第一行是一个整数，代表数据个数  $n(2 \leq n \leq 50000)$ ，第二行是  $n$  个整数  $a_1,a_2,\dots,a_n(|a_i| \leq 10000)$ 。

### 【输出】

输出一个整数，就是  $d(A)$  的值。

### 【输入样例】

```
1
10
1 -1 2 2 3 -3 4 -4 5 -5
```

### 【输出样例】

```
13
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
```

```

#define N 50001
#define MOD 50001
#define E 1e-12
using namespace std;
int a[N];
int leftt[N],rightt[N];
int leftmax[N],rightmax[N];
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n;
        cin>>n;
        for(int i=1;i<=n;i++)
            cin>>a[i];

        leftt[1]=a[1];
        leftmax[1]=a[1];
        rightt[n]=a[n];
        rightmax[n]=a[n];

        for(int i=2;i<=n;i++)
            leftt[i]=max(a[i],leftt[i-1]+a[i]);
        for(int i=n-1;i>=1;i--)
            rightt[i]=max(a[i],rightt[i+1]+a[i]);
        for(int i=2;i<=n;i++)
            leftmax[i]=max(leftmax[i-1],leftt[i]);
        for(int i=n-1;i>=1;i--)
            rightmax[i]=max(rightmax[i+1],rightt[i]);

        int ans=a[1];
        for(int i=2;i<=n;i++)
            ans=max(ans,leftmax[i-1]+rightmax[i]);

        cout<<ans<<endl;
    }
    return 0;
}

```

### 1.1.2.9.3.17 最长公共子上升序列

#### 最长公共子上升序列

原创 Alex\_McAvoy 发布于 2018-06-07 15:14:57 阅读数 1022 收藏  
展开

### 【题目描述】

给定两个整数序列，写一个程序求它们的最长上升公共子序列。

当以下条件满足的时候，我们将长度  $N$  的序列  $S_1, S_2, \dots, S_N$  称为长度为  $M$  的序列  $A_1, A_2, \dots, A_M$  的上升子序列：

存在  $1 \leq i_1 < i_2 < \dots < i_N \leq M$ ，使得对所有  $1 \leq j \leq N$ ，均有  $S_j = A_{i_j}$ ，且对于所有的  $1 \leq j < N$ ，均有  $S_j < S_{j+1}$ 。

### 【输入】

每个序列用两行表示，第一行是长度  $M$  ( $1 \leq M \leq 500$ )，第二行是该序列的  $M$  个整数  $A_i$  ( $-231 \leq A_i < 231$ )

### 【输出】

在第一行，输出两个序列的最长上升公共子序列的长度  $L$ 。在第二行，输出该子序列。如果有不止一个符合条件的子序列，则输出任何一个即可。

### 【输入样例】

```
5
1 4 2 5 -12
4
-12 1 2 4
```

### 【输出样例】

```
2
1 4
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 100001
#define E 1e-12
using namespace std;
struct Node{
```

```

    int len;
    int ans[N];
}order[N],now;
int a[N],b[N];
int main()
{
    int n,m;
    cin>>n;
    for(int i=1;i<=n;++i)
        cin>>a[i];
    cin>>m;
    for(int i=1;i<=m;++i)
        cin>>b[i];

    for(int i=1;i<=m;++i)
    {
        now.len=0;
        memset(now.ans,0,sizeof(now.ans));

        for(int j=1;j<=n;++j)
        {
            if(a[j]<b[i]&&order[j].len>now.len)
                now=order[j];

            if(a[j]==b[i])
            {
                order[j]=now;
                order[j].len++;
                order[j].ans[order[j].len]=a[j];
            }
        }
    }

    int flag=0;
    int maxx=-INF;
    for(int i=1;i<=n;++i)
        if(order[i].len>maxx)
        {
            maxx=order[i].len;
            flag=i;
        }

    cout<<order[flag].len<<endl;
    for(int i=1;i<=order[flag].len;++i)

```



```
    cout<<order[flag].ans[i]<<" ";
    cout<<endl;
    return 0;
}
```

### 1.1.3 第三部分 数据结构

### 1.1.3.1 第一章 栈

#### 1.1.3.1.1 后缀表达式的值

## 后缀表达式的值

原创 Alex\_McAvoy 发布于 2018-04-19 22:31:02 阅读数 2323 收藏  
展开

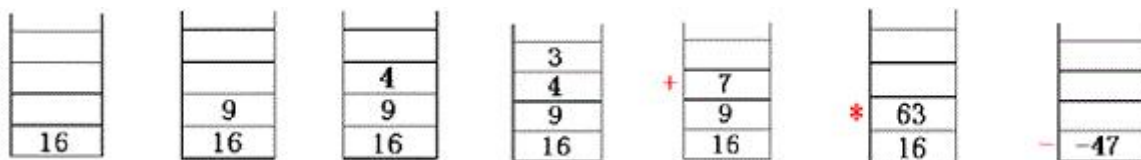
### 【题目描述】

从键盘读入一个后缀表达式（字符串），只含有 0-9 组成的运算数及加（+）、减（-）、乘（\*）、除（/）四种运算符。每个运算数之间用一个空格隔开，不需要判断给你的表达式是否合法。以@作为结束标志。

比如， $16-9*(4+3)$ 转换成后缀表达式为： $16\ \square\ 9\ \square\ 4\ \square\ 3\ \square\ +\ *\ -$ ，在字符数组 A 中的形式为：

a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a[11]	a[12]	a[13]
1	6	□	9	□	4	□	3	□	+	*	-	@

栈中的变化情况:



运行结果: -47

提示：输入字符串长度小于 250，参与运算的整数及结果之绝对值均在  $2^{64}$  范围内，如有除法保证能整除。

**【输入】**

一个后缀表达式。

**【输出】**

一个后缀表达式的值。

**【输入样例】**

16 9 4 3 +\* - @

**【输出样例】**

-47

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
char str[N];
stack<LL> S;
int main(){
    gets(str);
    int len=strlen(str)-1;
    for(int i=0;i<len;i++){
        switch(str[i]){
```

```

    case '+':{
        LL y=S.top();S.pop();
        LL x=S.top();S.pop();
        S.push(x+y);
        break;
    }
    case '-':{
        LL y=S.top();S.pop();
        LL x=S.top();S.pop();
        S.push(x-y);
        break;
    }
    case '*':{
        LL y=S.top();S.pop();
        LL x=S.top();S.pop();
        S.push(x*y);
        break;
    }
    case '/':{
        LL y=S.top();S.pop();
        LL x=S.top();S.pop();
        S.push(x/y);
        break;
    }
    case '@':i=len;break;
    default:{
        LL temp=0;
        while(str[i]!=' ')
            temp=temp*10+str[i]-'0',i++;
        S.push(temp);
        break;
    }
}
}
cout<<S.top()<<endl;
return 0;
}

```

### 1.1.3.1.2 表达式括号匹配

#### 表达式括号匹配

原创 Alex\_McAvoy 发布于 2018-04-19 22:33:44 阅读数 912 收藏  
展开

### 【题目描述】

假设一个表达式有英文字母（小写）、运算符（+，—，\*，/）和左右小（圆）括号构成，以“@”作为表达式的结束符。请编写一个程序检查表达式中的左右圆括号是否匹配，若匹配，则返回“YES”；否则返回“NO”。表达式长度小于 255，左圆括号少于 20 个。

### 【输入】

一行数据，即表达式。

### 【输出】

一行，即“YES” 或“NO”。

### 【输入样例】

2\*(x+y)/(1-x)@

### 【输出样例】

YES

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
```

```

char str[N];
stack<int> S;
int main(){
    gets(str);
    int len=strlen(str);

    for(int i=0;i<len;i++){
        if(str[i]=='(')//记录左括号
            S.push(1);
        else if(str[i]==')')//记录右括号
        {
            if(!S.empty())//栈不为空时
                S.pop();
            else
                S.push(1);
        }
    }
    if(S.empty())//栈为空说明匹配
        cout<<"YES"<<endl;
    else
        cout<<"NO"<<endl;
    return 0;
}

```

点赞

### 1.1.3.1.3 括弧匹配检验

#### 括弧匹配检验

原创 Alex\_McAvoy 发布于 2018-04-19 22:35:13 阅读数 2191 收藏  
展开

##### 【题目描述】

假设表达式中允许包含两种括号：圆括号和方括号，其嵌套的顺序随意，如  $((()))$  或  $[[[]]]$  等为正确的匹配， $[(])$  或  $([[])$  或  $(())$  均为错误的匹配。

现在的问题是，要求检验一个给定表达式中的括弧是否正确匹配？

输入一个只包含圆括号和方括号的字符串，判断字符串中的括弧是否匹配，匹配就输出“OK”，不匹配就输出“Wrong”。输入一个字符串： $[[[]]]$ ，输出：OK。

##### 【输入】

输入仅一行字符（字符个数小于 255）。

**【输出】**

匹配就输出“OK”，不匹配就输出“Wrong”。

**【输入样例】**

[()]

**【输出样例】**

Wrong

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
char str[10000];
stack<int> S;//用 1 代表(, 2 代表), 3 代表[, 4 代表]
int main(){
    cin>>str;
    int len=strlen(str);
    for(int i=0;i<len;i++)
    {
        if(str[i]=='(')//记录左圆括号
            S.push(1);
        else if(str[i]==')')//记录右圆括号
        {
            if(S.empty())//栈为空
```

```

        S.push(2);
    else if(S.top()==1)
        S.pop();
    else
        S.push(2);
}
else if(str[i]=='[')//记录左方括号
    S.push(3);
else if(str[i]==']')//记录右方括号
{
    if(S.empty())//栈为空
        S.push(4);
    else if(S.top()==3)
        S.pop();
    else
        S.push(4);
}
}
if(S.empty())
    printf("OK");
else
    printf("Wrong");
return 0;
}

```

### 1.1.3.1.4 字符串匹配问题

#### 字符串匹配问题

原创 Alex\_McAvoy 发布于 2018-04-19 22:36:44 阅读数 2161 收藏  
展开

##### 【题目描述】

字符串中只含有括号 (, [], <>, {}), 判断输入的字符串中括号是否匹配。如果括号有互相包含的形式, 从内到外必须是 <>, (), [], {}, 例如。输入: [()] 输出: YES, 而输入 (()), ([)] 都应该输出 NO。

##### 【输入】

第一行为一个整数  $n$ , 表示以下有多少个由括号组成的字符串。接下来的  $n$  行, 每行都是一个由括号组成的长度不超过 255 的字符串。

##### 【输出】

在输出文件中有  $n$  行, 每行都是 YES 或 NO。

## 【输入样例】

5

```

{}<><>()()[]
{}{}<><><><>()()[]
{}{}<><><><>()()[]
{<>}{[]}<<<><<<><<<><<<>((<>))()[(<>)][]
><>{}{}<<<><<<><<<><<<>((<>))()[(<>)][]

```

## 【输出样例】

```

YES
YES
YES
YES
NO

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
char a[]={'{','[','(','<','}',']',')','>'};//{→0 [→1 (→2 <→3 }→4 ]→5 )→6 >→7
char str[300];
int b[300];
stack<int> S;
int main(){

```



```

int n;
cin>>n;//输入测试数据组数
while(n--)
{
    while(!S.empty())
        S.pop();

    bool flag=true;//用于判断嵌套是否合法的标记

    cin>>str;
    int len=strlen(str);
    for(int i=0;i<len;i++)
        for(int j=0;j<8;j++)
            if(str[i]==a[j])
            {
                b[i]=j;////建立映射
                break;//跳出内循环
            }

    for(int i=0;i<len;i++)//判断是否匹配
    {
        if(b[i]<=3)//b[i]元素入栈
        {

            if(!S.empty()&&b[i]<S.top())//元素无法入栈，说明该组数据非法
            {
                flag=false;
                break;
            }
            else//b[i]元素可以入栈
                S.push(b[i]);
        }
        else if(b[i]>=4)//判断是否配对
        {
            if(S.empty()||(S.top()+4)!=b[i])//无法配对
            {
                flag=false;
                break;
            }
            else//可以配对
                S.pop();
        }
    }
}

```

```

    if(!S.empty())
        cout<<"NO"<<endl;//栈内有元素，匹配不成功
    else if(flag==1)
        cout<<"YES"<<endl;//栈内无元素，匹配成功
    else
        cout<<"NO"<<endl;//栈内无元素，匹配不成功
}
return 0;
}

```

### 1.1.3.1.5 计算

#### 计算

原创 Alex\_McAvoy 发布于 2018-04-19 22:38:55 阅读数 2093 收藏  
展开

##### 【题目描述】

小明在你的帮助下，破密了 Ferrari 设的密码门，正要往前走，突然又出现了一个密码门，门上有一个算式，其中只有“(", ")", "0-9", "+", "-", "\*", "/", "^", 求出的值就是密码。小明数学学得不好，还需你帮他的忙。(/ 用整数除法)

##### 【输入】

共 1 行，为一个算式。

##### 【输出】

共 1 行，就是密码。

##### 【输入样例】

1+(3+2)\*(7^2+6\*9)/(2)

##### 【输出样例】

258

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>

```

```

#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;

int n,m;
stack<int> s1;//操作数栈
stack<char> s2;//运算符栈

int lev(char x)//运算符优先级
{
    if(x=='+'||x=='-')
        return 1;
    if(x=='*'||x=='/')
        return 2;
    if(x=='^')
        return 3;
    return 0;
}

void calculate(stack<int> &s1,stack<char> &s2)//弹出栈顶元素并计算
{
    /*取出后弹出栈*/
    int y=s1.top();
    s1.pop();
    int x=s1.top();
    s1.pop();
    char z=s2.top();
    s2.pop();

    /*根据运算符计算,并压入栈*/
    if(z=='+')
        s1.push(x+y);

```

```

    if(z=='-')
        s1.push(x-y);
    if(z=='*')
        s1.push(x*y);
    if(z=='/')
        s1.push(x/y);
    if(z=='^')
        s1.push(pow(x,y));
}

int c(int x)
{
    return x!=0;
}
char str[1000000];
int sum[1000000];

int main(){

    scanf("%s",str+1);
    n=strlen(str+1);

    for(int i=1;i<=n;i++)//检查匹配
    {
        sum[i]+=sum[i-1];
        if(str[i]=='(')
            sum[i]++;
        if(str[i]==')')
            sum[i]--;
    }

    bool out=false;
    for(int i=2;i<=n;i++)
        if( c(lev(str[i])) && c(lev(str[i-1])) )
        {
            out=1;
            break;
        }

    if( ( n==1 && c(lev(str[1])) )||sum[n]||out )//表达式不合法
    {
        cout<<"NO"<<endl;
        return 0;
    }
}

```

```

stack<int> s1;
stack<char> s2;
int temp=0;
bool flag=false;
for(int i=1;i<=n;i++)
{
    if('0'<=str[i]&&str[i]<='9')//判断当前字符是否为数字
    {
        temp=(temp<<3)+(temp<<1)+str[i]-'0';
        flag=true;
    }
    else
    {
        if(flag)
        {
            s1.push(temp);
            temp=0;
            flag=false;
        }
        if(str[i]=='(')
        {
            s2.push(str[i]);
            continue;
        }
        if(str[i]==')')
        {
            while(s2.top()!='(')
                calculate(s1,s2);
            s2.pop();
            continue;
        }
        while(!s2.empty()&&lev(s2.top())>=lev(str[i]))//优先级判断
            calculate(s1,s2);
        s2.push(str[i]);//运算符入栈
    }
}
if(flag)
{
    s1.push(temp);
    temp=0;
    flag=false;
}
while(!s2.empty())

```

```

    calculate(s1,s2);
    cout<<s1.top()<<endl;
    return 0;
}

```

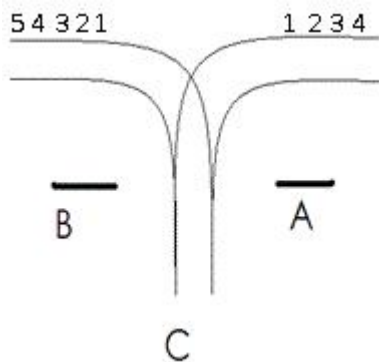
### 1.1.3.1.6 车厢调度

#### 车厢调度

原创 Alex\_McAvoy 发布于 2018-04-19 22:42:35 阅读数 1992 收藏  
展开

##### 【题目描述】

有一个火车站，铁路如图所示，每辆火车从 A 驶入，再从 B 方向驶出，同时它的车厢可以重新组合。假设从 A 方向驶来的火车有  $n$  节 ( $n \leq 1000$ )，分别按照顺序编号为 1, 2, 3, ...,  $n$ 。假定在进入车站前，每节车厢之间都不是连着的，并且它们可以自行移动到 B 处的铁轨上。另外假定车站 C 可以停放任意多节车厢。但是一旦进入车站 C，它就不能再回到 A 方向的铁轨上了，并且一旦当它进入 B 方向的铁轨，它就不能再回到车站 C。



负责车厢调度的工作人员需要知道能否使它以  $a_1, a_2, \dots, a_n$  的顺序从 B 方向驶出，请来判断能否得到指定的车厢顺序。

##### 【输入】

第一行为一个整数  $n$ ，其中  $n \leq 1000$ ，表示有  $n$  节车厢，第二行为  $n$  个数字，表示指定的车厢顺序。

##### 【输出】

如果可以得到指定的车厢顺序，则输出一个字符串“YES”，否则输出“NO”（注意要大写，不包含引号）。

**【输入样例】**

5  
5 4 3 2 1

**【输出样例】**

YES

**【提示】**

观察发现，整个调度过程其实是在模拟入栈出栈的过程，而在这个过程中，我们可以分成三种状态：栈前、栈中、栈后。我们可以发现，当某个数字出栈了，说明比它小的数字要么已经出栈了，要么还在栈里，不能是入栈前状态，并且在栈中的顺序是从大到小的(从栈顶往栈底看)，比如出 5，那么 1, 2, 3, 4 要么已经在 5 之前出了，要么还在栈中（假如 1, 3, 4 在栈中，从栈顶往栈底看依次为 4, 3, 1），不能是入栈前的状态。如果某个数字要出栈，那么当前在栈中的数字都必须小于它，否则就与栈的性质矛盾，不合法，于是我们可以这样解决：

从第一个数字开始扫描， $a[i]$ 表示当前出栈的数字，如果有比  $a[i]$  大的数字还在栈中，那么就产生矛盾，输出“NO”；否则，标记当前数字  $a[i]$  为栈后状态，那么  $[1, a[i]-1]$  这些数字如果还没出栈，标记为栈中状态。具体我们可以用 0 表示为确定状态，1 表示栈中状态，2 表示栈后状态。

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;
```

```

int a[N];
stack<int> S;
int main(){
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)//a[i]为到达 B 站的车厢
        cin>>a[i];

    int cur=1;//cur 为需要进栈的车厢
    for(int i=1;i<=n;i++)//进栈,到达 A 站; 出栈,到达 B 站
    {
        while(cur<=a[i])//比 a[i]小的车厢都要在栈中
            S.push(cur++);
        if(S.top()==a[i])//将 a[i]弹出栈
            S.pop();
        else
        {
            cout<<"NO"<<endl;
            return 0;
        }
    }
    cout<<"YES"<<endl;
    return 0;
}

```

### 1.1.3.1.7 中缀表达式值

#### 中缀表达式值

原创 Alex\_McAvoy 发布于 2018-04-19 22:51:29 阅读数 1716 收藏  
展开

##### 【题目描述】

输入一个中缀表达式（由 0-9 组成的运算数、加+减—乘\*除/四种运算符、左右小括号组成。注意“—”也可作为负数的标志，表达式以“@”作为结束符），判断表达式是否合法，如果不合法，请输出“NO”；否则请把表达式转换成后缀形式，再求出后缀表达式的值并输出。

注意：必须用栈操作，不能直接输出表达式的值。

##### 【输入】

行为一个以@结束的字符串。



**【输出】**

如果表达式不合法，请输出“NO”，要求大写。  
如果表达式合法，请输出计算结果。

**【输入样例】**

1+2\*8-9

**【输出样例】**

8

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#define PI acos(-1.0)
#define E 1e-9
#define INF 0x3f3f3f3f
#define LL long long
const int MOD=1000000007;
const int N=10000+5;
const int dx[]={-1,1,0,0};
const int dy[]={0,0,-1,1};
using namespace std;

int n,m;
stack<int> s1;//操作数栈
stack<char> s2;//运算符栈

int lev(char x)//运算符优先级
{
    if(x=='+'||x=='-')
        return 1;
    if(x=='*'||x=='/')
        return 2;
```

```

    if(x=='^')
        return 3;
    return 0;
}

void calculate(stack<int> &s1,stack<char> &s2)//弹出栈顶元素并计算
{
    /*取出后弹出栈*/
    int y=s1.top();
    s1.pop();
    int x=s1.top();
    s1.pop();
    char z=s2.top();
    s2.pop();

    /*根据运算符计算,并压入栈*/
    if(z=='+')
        s1.push(x+y);
    if(z=='-')
        s1.push(x-y);
    if(z=='*')
        s1.push(x*y);
    if(z=='/')
        s1.push(x/y);
    if(z=='^')
        s1.push(pow(x,y));
}

int c(int x)
{
    return x!=0;
}
char str[1000000];
int sum[1000000];

int main(){

    scanf("%s",str+1);
    n=strlen(str+1)-1;//忽略掉@的字符串长度

    for(int i=1;i<=n;i++)//检查匹配
    {
        sum[i]=sum[i-1];
        if(str[i]=='(')

```

```

        sum[i]++;
        if(str[i]=='')
            sum[i]--;
    }

    bool out=false;
    for(int i=2;i<=n;i++)
        if( c(lev(str[i])) && c(lev(str[i-1])) )
        {
            out=1;
            break;
        }

    if( ( n==1 && c(lev(str[1])) )||sum[n]||out )//表达式不合法
    {
        cout<<"NO"<<endl;
        return 0;
    }

    stack<int> s1;
    stack<char> s2;
    int temp=0;
    bool flag=false;
    for(int i=1;i<=n;i++)
    {
        if('0'<=str[i]&&str[i]<='9')//判断当前字符是否为数字
        {
            temp=(temp<<3)+(temp<<1)+str[i]-'0';
            flag=true;
        }
        else
        {
            if(flag)
            {
                s1.push(temp);
                temp=0;
                flag=false;
            }
            if(str[i]=='(')
            {
                s2.push(str[i]);
                continue;
            }
            if(str[i]==')')

```

```

    {
        while(s2.top()!='(')
            calculate(s1,s2);
        s2.pop();
        continue;
    }
    while(!s2.empty()&&lev(s2.top())>=lev(str[i]))//优先级判断
        calculate(s1,s2);
    s2.push(str[i]);//运算符入栈
}
}
if(flag)
{
    s1.push(temp);
    temp=0;
    flag=false;
}
while(!s2.empty())
    calculate(s1,s2);
cout<<s1.top()<<endl;
return 0;
}

```

## 1.1.3.2 第二章 队列

### 1.1.3.2.1 周末舞会

#### 周末舞会

原创 Alex\_McAvoy 发布于 2018-06-08 22:53:09 阅读数 712 收藏  
展开

##### 【题目描述】

假设在周末舞会上，男士们和女士们进入舞厅时，各自排成一队。跳舞开始时，依次从男队和女队的队头上各出一人配成舞伴。规定每个舞曲能有一对跳舞者。若两队初始人数不相同，则较长的那一队中未配对者等待下一轮舞曲。现要求写一个程序，模拟上述舞伴配对问题。

**【输入】**

第一行两队的人数;

第二行舞曲的数目。

**【输出】**

配对情况。

**【输入样例】**

```
4 6
7
```

**【输出样例】**

```
1 1
2 2
3 3
4 4
1 5
2 6
3 1
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int a[N],b[N];
int main()
{
    int m,n;
    cin>>m>>n;
    for(int i=1;i<=m;i++)
        a[i]=i;
    for(int i=1;i<=n;i++)
        b[i]=i;
```

```

int k,k1=1;
int r1=m,r2=n;
int f1=1,f2=1;

cin>>k;
while(k1<=k)
{
    cout<<a[f1]<<" "<<b[f2]<<endl;

    r1++;
    r2++;

    a[r1]=a[f1];
    b[r2]=b[f2];

    f1++;
    f2++;

    k1++;
}
return 0;
}

```

### 1.1.3.2.2 Blah 数集

#### Blah 数集

原创 Alex\_McAvoy 发布于 2018-06-08 22:51:31 阅读数 1056 收藏  
展开

##### 【题目描述】

大数学家高斯小时候偶然间发现一种有趣的自然数集合 **Blah**，对于以 **a** 为基的集合 **Ba** 定义如下：

- (1) **a** 是集合 **Ba** 的基，且 **a** 是 **Ba** 的第一个元素；
- (2) 如果 **x** 在集合 **Ba** 中，则  $2x+1$  和  $3x+1$  也都在集合 **Ba** 中；
- (3) 没有其他元素在集合 **Ba** 中了。

现在小高斯想知道如果将集合 **Ba** 中元素按照升序排列，第 **N** 个元素会是多少？

**【输入】**

输入包括很多行，每行输入包括两个数字，集合的基  $a(1 \leq a \leq 50)$  以及所求元素序号  $n(1 \leq n \leq 1000000)$ 。

**【输出】**

对于每个输入，输出集合  $B_a$  的第  $n$  个元素值。

**【输入样例】**

```
1 100
28 5437
```

**【输出样例】**

```
418
900585
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1000001
#define MOD 123
#define E 1e-6
using namespace std;
int q[N];
int main()
{
    int a,n;
    int x,y;
    while(cin>>a>>n)
    {
        int head1=1,head2=1,tail=1;
        q[tail]=a;
        tail++;

        while(tail<=n)
        {
            int x=q[head1]*2+1;
```

```

    int y=q[head2]*3+1;
    if(x<y)
    {
        q[tail]=x;
        tail++;
        head1++;
    }
    else if(x>y)
    {
        q[tail]=y;
        tail++;
        head2++;
    }
    else
    {
        q[tail]=x;
        tail++;
        head1++;
        head2++;
    }
}
cout<<q[tail-1]<<endl;
}
return 0;
}

```

### 1.1.3.2.3 围圈报数

#### 围圈报数

原创 Alex\_McAvoy 发布于 2018-06-08 22:50:15 阅读数 632 收藏  
展开

##### 【题目描述】

有  $n$  个人依次围成一圈，从第 1 个人开始报数，数到第  $m$  个人出列，然后从出列的下一个开始报数，数到第  $m$  个人又出列，...，如此反复到所有的人全部出列为止。设  $n$  个人的编号分别为 1, 2, ...,  $n$ ，打印出列的顺序。

##### 【输入】

$n$  和  $m$ 。

##### 【输出】

出列的顺序。



## 【输入样例】

4 17

## 【输出样例】

1 3 4 2

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int main()
{
    int n,m;
    cin>>n>>m;

    queue<int> Q;
    for(int i=1;i<=n;i++)
        Q.push(i);

    int cnt=0;
    while(!Q.empty()){
        cnt++;
        int temp=Q.front();
        if(cnt==m){
            cnt=0;
            cout<<temp<<" ";
        }
        else{
            Q.push(temp);
        }
        Q.pop();
    }
    cout<<endl;
}

```

```

    return 0;
}

```

### 1.1.3.2.4 连通块

#### 连通块

原创 Alex\_McAvoy 发布于 2018-06-08 22:49:02 阅读数 1289 收藏  
展开

##### 【题目描述】

一个  $n * m$  的方格图，一些格子被涂成了黑色，在方格图中被标为 1，白色格子标为 0。问有多少个四连通的黑色格子连通块。四连通的黑色格子连通块指的是一片由黑色格子组成的区域，其中的每个黑色格子能通过四连通的走法（上下左右），只走黑色格子，到达该联通块中的其它黑色格子。

##### 【输入】

第一行两个整数  $n, m (1 \leq n, m \leq 100)$ ，表示一个  $n * m$  的方格图。

接下来  $n$  行，每行  $m$  个整数，分别为 0 或 1，表示这个格子是黑色还是白色。

##### 【输出】

一行一个整数  $ans$ ，表示图中有  $ans$  个黑色格子连通块。

##### 【输入样例】

```

3 3
1 1 1
0 1 0
1 0 1

```

##### 【输出样例】

```

3

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<stdlib.h>
#include<queue>
#include<vector>

```

```

#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int x;
    int y;
}q[10100],p;
int a[N][N],vis[N][N];
int next[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            cin>>a[i][j];

    int cnt=0;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++)
            if(vis[i][j]==0&&a[i][j]==1)
            {
                cnt++;
                vis[i][j]=1;
                int head=1,tail=1;
                q[tail].x=i;
                q[tail].y=j;

                tail++;
                while(head<tail)
                {
                    p=q[head];
                    for(int k=0;k<4;k++)
                    {
                        int nx=p.x+next[k][0];
                        int ny=p.y+next[k][1];
                        if(vis[nx][ny]==0&&a[nx][ny]==1)
                        {
                            vis[nx][ny]=1;
                            q[tail].x=nx;
                            q[tail].y=ny;
                        }
                    }
                }
            }
}

```

```

        tail++;
    }
}
    head++;
}

cout<<cnt<<endl;
return 0;
}

```

### 1.1.3.2.5 围成面积

#### 围成面积

原创 Alex\_McAvoy 发布于 2018-06-08 23:07:00 阅读数 2096 收藏  
展开

##### 【题目描述】

编程计算由“\*”号围成的下列图形的面积。面积计算方法是统计\*号所围成的闭合曲线中水平线和垂直线交点的数目。如下图所示，在  $10 \times 10$  的二维数组中，有“\*”围住了 15 个点，因此面积为 15。

##### 【输入】

$10 \times 10$  的图形。

##### 【输出】

输出面积

##### 【输入样例】

```

0000000000
0000111000
0000100100
0000010010
0010001010
0101010010
0100110110
0010000100
0001111100
0000000000

```

## 【输出样例】

15

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 11
#define MOD 123
#define E 1e-6
using namespace std;
int a[N][N];
int vis[N][N];
int dir[4][2]={{-1,0},{1,0},{0,-1},{0,1}};
int cnt=0;
struct node{
    int x;
    int y;
}q[200];
void bfs(int x0,int y0)
{
    a[x0][y0]=1;
    vis[x0][y0]=1;

    int head=1,tail=1;
    q[tail].x=x0;
    q[tail].y=y0;
    tail++;
    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(1<=nx&&nx<=10&&1<=ny&&ny<=10&&a[nx][ny]==0&&vis[nx][ny]==0)

```

```

        {
            vis[nx][ny]=1;
            a[nx][ny]=1;
            q[tail].x=nx;
            q[tail].y=ny;
            tail++;
        }
    }
    head++;
}
}
void bfs_cnt(int x0,int y0)
{
    vis[x0][y0]=1;
    cnt++;

    int head=1,tail=1;
    q[tail].x=x0;
    q[tail].y=y0;
    tail++;

    while(head<tail)
    {
        int x=q[head].x;
        int y=q[head].y;
        for(int i=0;i<4;i++)
        {
            int nx=x+dir[i][0];
            int ny=y+dir[i][1];
            if(1<=nx&&nx<=10&&1<=ny&&ny<=10&&a[nx][ny]==0&&vis[nx][ny]==0)
            {
                vis[nx][ny]=1;
                q[tail].x=nx;
                q[tail].y=ny;
                tail++;
                cnt++;
            }
        }
        head++;
    }
}
}
int main()
{
    for(int i=1;i<=10;i++)

```

```

for(int j=1;j<=10;j++)
    cin>>a[i][j];

for(int j=1;j<=10;j++)
    if(a[1][j]==0)
        bfs(1,j);
for(int j=1;j<=10;j++)
    if(a[10][j]==0)
        bfs(10,j);
for(int i=1;i<=10;i++)
    if(a[1][i]==0)
        bfs(1,i);
for(int i=1;i<=10;i++)
    if(a[10][i]==0)
        bfs(10,i);

for(int i=1;i<=10;i++)
    for(int j=1;j<=10;j++)
        if(a[i][j]==0&&vis[i][j]==0)
            bfs_cnt(i,j);

cout<<cnt<<endl;
return 0;
}

```

### 1.1.3.2.6 奇怪的电梯

#### 奇怪的电梯

原创 Alex\_McAvoy 发布于 2018-06-08 23:05:47 阅读数 829 收藏  
展开

##### 【题目描述】

大楼的每一层楼都可以停电梯，而且第  $i$  层楼 ( $1 \leq i \leq N$ ) 上有一个数字  $K_i$  ( $0 \leq K_i \leq N$ )。电梯只有四个按钮：开，关，上，下。上下的层数等于当前楼层上的那个数字。当然，如果不能满足要求，相应的按钮就会失灵。例如：3 3 1 2 5 代表了  $K_i$  ( $K_1=3, K_2=3, \dots$ )，从一楼开始。在一楼，按“上”可以到 4 楼，按“下”是不起作用的，因为没有 -2 楼。那么，从 A 楼到 B 楼至少要按几次按钮呢？

##### 【输入】

共有二行，第一行为三个用空格隔开的正整数，表示  $N, A, B$  ( $1 \leq N \leq 200, 1 \leq A, B \leq N$ )，第二行为  $N$  个用空格隔开的正整数，表示  $K_i$ 。

**【输出】**

一行，即最少按键次数,若无法到达，则输出-1。

**【输入样例】**

```
5 1 5
3 3 1 2 5
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 301
#define MOD 123
#define E 1e-6
using namespace std;
int dp[N],k[N];
int main()
{
    int n,a,b;
    bool flag=true;

    cin>>n>>a>>b;
    for(int i=1;i<=n;i++)
        cin>>k[i];

    dp[a]=1;
    while(flag)
    {
        flag=false;
        for(int i=1;i<=n;i++)
        {
            if(dp[i]>0)
            {
                if(i-k[i]>0)
```



```

    {
        if(dp[i-k[i]]==0||dp[i-k[i]]>dp[i]+1)
        {
            dp[i-k[i]]=dp[i]+1;
            flag=true;
        }
    }
    if(i+k[i]<=n)
    {
        if(dp[i+k[i]]==0||dp[i+k[i]]>dp[i]+1)
        {
            dp[i+k[i]]=dp[i]+1;
            flag=true;
        }
    }
}
}
cout<<dp[b]-1<<endl;

return 0;
}

```

### 1.1.3.2.7 产生数

#### 产生数

原创 Alex\_McAvoy 发布于 2018-06-08 23:02:52 阅读数 890 收藏  
展开

##### 【题目描述】

给出一个整数  $n$  ( $n \leq 2000$ ) 和  $k$  个变换规则 ( $k \leq 15$ )。规则：

- ① 1 个数字可以变换成另 1 个数字；
- ② 规则中，右边的数字不能为零。

例如：  $n=234$ ，  $k=2$  规则为  $2 \rightarrow 5$ ，  $3 \rightarrow 6$

上面的整数 234 经过变换后可能产生出的整数为（包括原数）234， 534， 264， 564  
共 4 种不同的产生数。

求经过任意次的变换（0 次或多次），能产生出多少个不同的整数。仅要求输出不同整数个数。

**【输入】**

n

k

x1 x2 ... xn

y1 y2 ... yn

**【输出】**

格式为一个整数（满足条件的整数个数）。

**【输入样例】**

234

2

2 5

3 6

**【输出样例】**

4

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int vis[N];
char q[N][10];
struct node{
    int x;
    int y;
}b[20];
```

```

int turn(char s[],int len)
{
    int ans=0;
    for(int i=0;i<len;i++)
    {
        ans*=10;
        ans+=s[i]-'0';
    }
    return ans;
}
int main()
{
    char s_t[10];
    char s[10];
    int k;

    cin>>s;
    int len=strlen(s);
    cin>>k;
    for(int i=1;i<=k;i++)
        cin>>b[i].x>>b[i].y;

    int head=1,tail=1;
    int cnt=1;

    strcpy(q[tail],s);
    tail++;
    vis[turn(s,len)]=1;

    while(head<tail)
    {
        for(int i=1;i<=k;i++)
        {
            strcpy(s_t,q[head]);
            for(int j=0;j<len;j++)
                if(s_t[j]==b[i].x+'0')
                {
                    s_t[j]=b[i].y+'0';
                    int d=turn(s_t,len);
                    if(vis[d]==0)
                    {
                        vis[d]=1;
                        strcpy(q[tail],s_t);
                        tail++;
                    }
                }
        }
    }
}

```

```

        cnt++;
    }
    strcpy(s_t,q[head]);//初始化 s_t
    }
}
head++;
}
cout<<cnt<<endl;
return 0;
}

```

### 1.1.3.2.8 家庭问题

#### 家庭问题

原创 Alex\_McAvoy 发布于 2018-06-09 00:29:47 阅读数 1520 收藏  
展开

##### 【题目描述】

有  $n$  个人，编号为  $1, 2, \dots, n$ ，另外还知道存在  $K$  个关系。一个关系的表达为二元组  $(\alpha, \beta)$  形式，表示  $\alpha, \beta$  为同一家庭的成员。

当  $n, k$  和  $k$  个关系给出之后，求出其中共有多少个家庭、最大的家庭中有多少人？

例如：  $n=6, k=3$ ，三个关系为  $(1,2), (1,3), (4,5)$

此时，6 个人组成三个家庭，即：  $\{1,2,3\}$  为一个家庭，  $\{4,5\}$  为一个家庭，  $\{6\}$  单独为一个家庭，第一个家庭的人数为最多。

##### 【输入】

第一行为  $n, k$  二个整数 ( $1 \leq n \leq 100$ ) (用空格分隔)；

接下来的  $k$  行，每行二个整数 (用空格分隔) 表示关系。

##### 【输出】

二个整数 (分别表示家庭个数和最大家庭人数)。

##### 【输入样例】

```

6 3
1 2
1 3
4 5

```

## 【输出样例】

3 3

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 301
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int a[N];
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
{
    int f1=Find(x);
    int f2=Find(y);
    if(f1!=f2)
        father[f2]=f1;
}
int main()
{
    int n,k;

    cin>>n>>k;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=k;i++)
    {
        int x,y;
        cin>>x>>y;
    }
}

```

```

    Union(x,y);
}

for(int i=1;i<=n;i++)
    a[Find(i)]++;

int cnt_1=0,cnt_2=0;
for(int i=1;i<=n;i++)
    if(a[i]>cnt_2)
        cnt_2=a[i];
for(int i=1;i<=n;i++)
    if(father[i]==i)
        cnt_1++;

cout<<cnt_1<<" "<<cnt_2<<endl;
return 0;
}

```

### 1.1.3.3 第三章 树与堆

#### 1.1.3.3.1 第一节 树与二叉树

##### 1.1.3.3.1.1 找树根和孩子

##### 找树根和孩子

原创 Alex\_McAvoy 发布于 2018-06-09 23:34:33 阅读数 996 收藏  
展开

##### 【题目描述】

给定一棵树，输出树的根 **root**，孩子最多的结点 **max** 以及他的孩子。

##### 【输入】

第一行： **n**（结点个数 $\leq 100$ ）， **m**（边数 $\leq 200$ ）。

以下 **m** 行： 每行两个结点 **x** 和 **y**，表示 **y** 是 **x** 的孩子( $x,y \leq 1000$ )。

**【输出】**

第一行：树根：root;  
 第二行：孩子最多的结点 max;  
 第三行：max 的孩子（按编号由小到输出）。

**【输入样例】**

```
8 7
4 1
4 2
1 3
1 5
2 6
2 7
2 8
```

**【输出样例】**

```
4
2
6 7 8
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
int tree[101];
using namespace std;
int main()
{
    int n,m;
    int x,y;

    cin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        cin>>x>>y;
        tree[y]=x;
    }
}
```

```

    }

    int root;
    for(int i=1;i<=n;i++)
        if(tree[i]==0)
        {
            root=i;
            break;
        }
    int maxx=-INF;
    int maxroot;
    for(int i=1;i<=n;i++)
    {
        int sum=0;
        for(int j=1;j<=n;j++)
            if(tree[j]==i)
                sum++;
        if(maxx<sum)
        {
            maxx=sum;
            maxroot=i;
        }
    }

    cout<<root<<endl;
    cout<<maxroot<<endl;
    for(int i=1;i<=n;i++)
        if(tree[i]==maxroot)
            cout<<i<<" ";
    cout<<endl;

    return 0;
}

```

### 1.1.3.3.1.2 单词查找树

#### 单词查找树

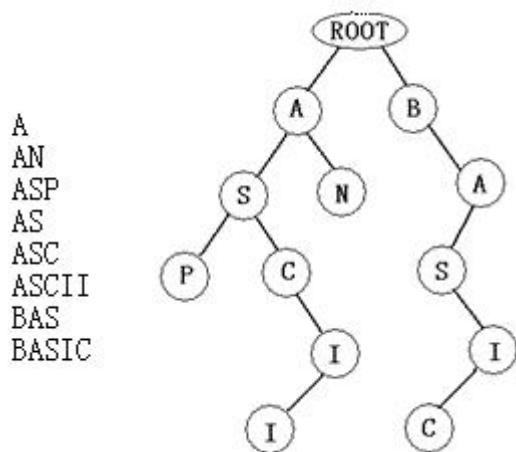
原创 Alex\_McAvoy 发布于 2018-06-09 23:34:34 阅读数 885 收藏  
展开



**【题目描述】**

在进行文法分析的时候，通常需要检测一个单词是否在我们的单词列表里。为了提高查找和定位的速度，通常都画出与单词列表所对应的单词查找树，其特点如下：

1. 根结点不包含字母，除根结点外每一个结点都仅包含一个大写英文字母；
2. 从根结点到某一结点，路径上经过的字母依次连起来所构成的字母序列，称为该结点对应的单词。单词列表中的每个单词，都是该单词查找树某个结点所对应的单词；
3. 在满足上述条件下，该单词查找树的结点数最少。
4. 例如图 3-2 左边的单词列表就对应于右边的单词查找树。注意，对一个确定的单词列表，请统计对应的单词查找树的结点数（包含根结点）。

**【输入】**

为一个单词列表，每一行仅包含一个单词和一个换行/回车符。每个单词仅由大写的英文字母组成，长度不超过 63 个字母。文件总长度不超过 32K，至少有一行数据。

**【输出】**

仅包含一个整数，该整数为单词列表对应的单词查找树的结点数。

**【输入样例】**

```

A
AN
ASP
AS
ASC
ASCII
  
```

BAS

BASIC

【输出样例】

13

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
int main()
{
    string a[N];
    int n=0;
    while(cin>>a[++n]);
    sort(a+1,a+1+n);

    int sum=a[1].length();
    for(int i=2;i<=n;i++)//依次计算每个单词对前一单词的差
    {
        int j=0;
        while(a[i][j]==a[i-1][j]&&j<a[i-1].length())//求两个单词相同部分的长度
            j++;
        sum+=a[i].length()-j;//累加两个单词的差
    }
    cout<<sum+1<<endl;

    return 0;
}
```

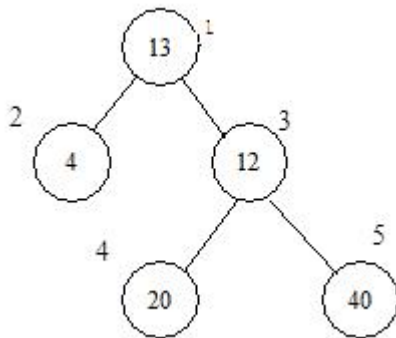
### 1.1.3.3.1.3 医院设置

医院设置

原创 Alex\_McAvoy 发布于 2018-06-09 23:32:15 阅读数 896 收藏  
展开

### 【题目描述】

设有一棵二叉树（如图 3-8，其中圈中的数字表示结点中居民的人口，圈边上数字表示结点编号。现在要求在某个结点上建立一个医院，使所有居民所走的路程之和为最小，同时约定，相邻结点之间的距离为 1。就本图而言，若医院建在 1 处，则距离和  $=4+12+2*20+2*40=136$ ；若医院建在 3 处，则距离和  $=4*2+13+20+40=81$ ...



### 【输入】

第一行一个整数  $n$ ，表示树的结点数 ( $n \leq 100$ )。接下来的  $n$  行每行描述了一个结点的状况，包含三个整数，整数之间用空格（一个或多个）分隔，其中：第一个数为居民人口数；第二个数为左链接，为 0 表示无链接；第三个数为右链接，为 0 表示无链接。

### 【输出】

一个整数，表示最小距离和。

### 【输入样例】

```

5
13 2 3
4 0 0
12 4 5
20 0 0
40 0 0
  
```

### 【输出样例】

```
81
```

### 【源程序】

```

#include<iostream>
#include<cstdio>
  
```

```

#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
int a[N][N],b[N],sum[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            if(i==j)
                a[i][j]=0;
            else
                a[i][j]=INF;
        }
    for(int i=1;i<=n;i++)
    {
        int left,right;
        cin>>b[i]>>left>>right;
        if(left!=0)
            a[i][left]=a[left][i]=1;
        if(right!=0)
            a[i][right]=a[right][i]=1;
    }

    for(int k=1;k<=n;k++)
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                if(a[i][j]>a[i][k]+a[k][j])
                    a[i][j]=a[i][k]+a[k][j];

    int minn=INF;
    for(int i=1;i<=n;i++)

```

```

{
    for(int j=1;j<=n;j++)
        sum[i]+=a[i][j]*b[j];
    if(sum[i]<minn)
        minn=sum[i];
}
cout<<minn<<endl;
return 0;
}

```

### 1.1.3.3.1.4 求后序遍历

#### 求后序遍历

原创 Alex\_McAvoy 发布于 2018-06-09 23:29:00 阅读数 1079 收藏  
展开

##### 【题目描述】

输入一棵二叉树的先序和中序遍历序列，输出其后序遍历序列。

##### 【输入】

共两行，第一行一个字符串，表示树的先序遍历，第二行一个字符串，表示树的中序遍历。树的结点一律用小写字母表示。

##### 【输出】

一行，表示树的后序遍历序列。

##### 【输入样例】

```
abdec
dbeac
```

##### 【输出样例】

```
debca
```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)

```

```

#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
string str1,str2;
void calculate(int left1,int right1,int left2,int right2)
{
    int m=str2.find(str1[left1]);
    if(m>left2)
        calculate(left1+1,left1+m-left2,left2,m-1);
    if(m<right2)
        calculate(left1+m-left2+1,right1,m+1,right2);
    cout<<str1[left1];
}
int main()
{
    cin>>str1>>str2;
    calculate(0,str1.length()-1,0,str2.length()-1);
    cout<<endl;
    return 0;
}

```

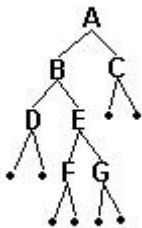
### 1.1.3.3.1.5 扩展二叉树

#### 扩展二叉树

原创 Alex\_McAvoy 发布于 2018-06-09 23:27:53 阅读数 726 收藏  
展开

##### 【题目描述】

由于先序、中序和后序序列中的任一个都不能唯一确定一棵二叉树，所以对二叉树做如下处理，将二叉树的空结点用·补齐，如图所示。我们把这样处理后的二叉树称为原二叉树的扩展二叉树，扩展二叉树的先序和后序序列能唯一确定其二叉树。



现给出扩展二叉树的先序序列，要求输出其中序和后序序列。

**【输入】**

扩展二叉树的先序序列。

**【输出】**

输出其中序和后序序列。

**【输入样例】**

ABD..EF..G..C..

**【输出样例】**

DBFEGAC

DFGEBCA

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
typedef struct Node;
typedef Node *tree;
struct Node{
    char data;
    tree lchild,rchild;
};
tree bt;
int i=-1;
string s;
void build(tree &bt)//建树
{
    if(s[++i]!='.')
    {
        bt=new Node;
        bt=new Node;
        bt->data=s[i];
        build(bt->lchild);
        build(bt->rchild);
    }
}
```

```

    }
    else
        bt=NULL;
    }
void inorder(tree bt)//输出中序序列
{
    if(bt)
    {
        inorder(bt->lchild);
        cout<<bt->data;
        inorder(bt->rchild);
    }
}
void postorder(tree bt)//输出后序序列
{
    if(bt)
    {
        postorder(bt->lchild);
        postorder(bt->rchild);
        cout<<bt->data;
    }
}
int main()
{
    cin>>s;
    build(bt);
    inorder(bt);
    cout<<endl;
    postorder(bt);
    cout<<endl;
    return 0;
}

```

### 1.1.3.3.1.6 小球

#### 小球

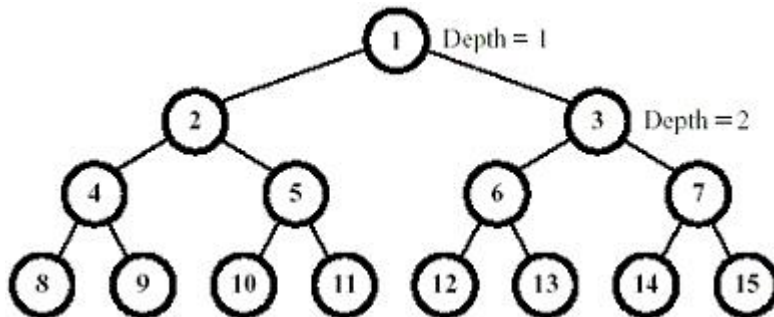
原创 Alex\_McAvoy 发布于 2018-06-09 23:24:50 阅读数 834 收藏  
展开

##### 【题目描述】

许多的小球一个一个的从一棵满二叉树上掉下来组成 FBT (Full Binary Tree, 满二叉树), 每一时间, 一个正在下降的球第一个访问的是非叶子节点。然后继续下降时,



或者走右子树，或者走左子树，直到访问到叶子节点。决定球运动方向的是每个节点的布尔值。最初，所有的节点都是 **false**，当访问到一个节点时，如果这个节点是 **false**，则这个球把它变成 **true**，然后从左子树走，继续它的旅程。如果节点是 **true**，则球也会改变它为 **false**，而接下来从右子树走。满二叉树的标记方法如下图：



因为所有的节点最初为 **false**，所以第一个球将会访问节点 1，节点 2 和节点 4，转变节点的布尔值后在节点 8 停止。第二个球将会访问节点 1、3、6，在节点 12 停止。明显地，第三个球在它停止之前，会访问节点 1、2、5，在节点 10 停止。

现在你的任务是，给定 FBT 的深度  $D$ ，和  $I$ ，表示第  $I$  个小球下落，你可以假定  $I$  不超过给定的 FBT 的叶子数，写一个程序求小球停止时的叶子序号。

#### 【输入】

一行包含两个用空格隔开的整数  $D$  和  $I$ 。其中  $2 \leq D \leq 20$ ， $1 \leq I \leq 524288$ 。

#### 【输出】

对应输出第  $I$  个小球下落停止时的叶子序号。

#### 【输入样例】

4 2

#### 【输出样例】

12

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
```

```

#define N 1000001
#define MOD 123
#define E 1e-6
using namespace std;
int D,l;
int b[N];
int ans;
void down(int k)
{
    if( pow(2,(D-1))<=k && k<pow(2,D) )
    {
        ans=k;
        return;
    }
    if(b[k])
    {
        b[k]=!b[k];
        down(2*k+1);
    }
    else
    {
        b[k]=!b[k];
        down(2*k);
    }
}
int main()
{
    int i,a;

    cin>>D>>l;
    for(i=1;i<=l;i++)
        down(1);
    cout<<ans<<endl;
    return 0;
}

```

点赞

收藏

分享

### 1.1.3.3.1.7 二叉树遍历

#### 二叉树遍历

原创 Alex\_McAvoy 发布于 2018-06-09 23:23:25 阅读数 845 收藏  
展开

### 【题目描述】

树和二叉树基本上都有先序、中序、后序、按层遍历等遍历顺序，给定中序和其它一种遍历的序列就可以确定一棵二叉树的结构。

假定一棵二叉树一个结点用一个字符描述，现在给出中序和按层遍历的字符串，求该树的先序遍历字符串。

### 【输入】

两行，每行是由字母组成的字符串（一行的每个字符都是唯一的），分别表示二叉树的中序遍历和按层遍历的序列。

### 【输出】

一行，表示二叉树的先序序列。

### 【输入样例】

DBEAC  
ABCDE

### 【输出样例】

ABDEC

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int len;
char a[N],b[N];
void preorder(int left,int right)
{
    int i,j;
    bool flag=false;
    for(i=0;i<len;i++)
```

```

{
    for(j=left;j<=right;j++)
        if(b[i]==a[j])
        {
            cout<<a[j];
            flag=1;
            break;
        }
    if(flag)
        break;
}
if(left<j)
    preorder(left,j-1);
if(j<right)
    preorder(j+1,right);
}
int main()
{
    cin>>a>>b;
    len=strlen(a);
    preorder(0,len-1);
    return 0;
}

```

### 1.1.3.3.1.8 FBI 树

#### FBI 树

原创 Alex\_McAvoy 发布于 2018-06-09 23:22:09 阅读数 790 收藏  
展开

##### 【题目描述】

我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为 B 串，全“1”串称为 I 串，既含“0”又含“1”的串则称为 F 串。

FBI 树是一种二叉树，它的结点类型也包括 F 结点，B 结点和 I 结点三种。由一个长度为  $2N$  的“01”串  $S$  可以构造出一棵 FBI 树  $T$ ，递归的构造方法如下：

$T$  的根结点为  $R$ ，其类型与串  $S$  的类型相同；

若串  $S$  的长度大于 1，将串  $S$  从中间分开，分为等长的左右子串  $S_1$  和  $S_2$ ；由左子串  $S_1$  构造  $R$  的左子树  $T_1$ ，由右子串  $S_2$  构造  $R$  的右子树  $T_2$ 。

现在给定一个长度为  $2N$  的“01”串，请用上述构造方法构造出一棵 FBI 树，并输出它的后序遍历序列。

**【输入】**

第一行是一个整数  $N$  ( $0 \leq N \leq 10$ )，第二行是一个长度为  $2N$  的“01”串。

**【输出】**

一行，这一行只包含一个字符串，即 FBI 树的后序遍历序列。

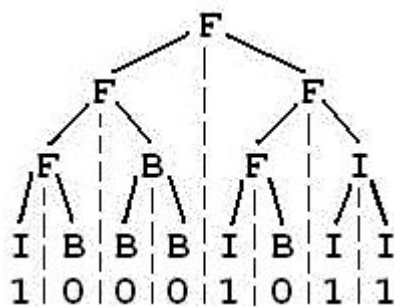
**【输入样例】**

3  
10001011

**【输出样例】**

IBFBBBFIBFIIFF

**【提示】**



**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 3001
#define MOD 123
```

```

#define E 1e-6
using namespace std;
int n;
char a[N],b[N];
void build(char *s,int n)
{
    int k=0;
    for(int i=(1<n);i<=(1<(n+1))-1;i++)
    {
        if(a[k++]=='0')
            b[i]='B';
        else
            b[i]='I';
    }

    for(int i=n-1;i>=0;i--)
        for(int j=(1<i);j<=(1<(i+1))-1;j++)
        {
            if(b[2*j]=='B'&&b[2*j+1]=='B')
                b[j]='B';
            else if(b[2*j]=='I'&&b[2*j+1]=='I')
                b[j]='I';
            else
                b[j]='F';
        }
}

void visit(int node)
{
    if(node>( 1<(n+1))-1 )
        return;
    visit(node*2);
    visit(node*2+1);
    cout<<b[node];
}

int main()
{
    cin>>n;
    cin>>a;
    build(a,n);
    visit(1);
    return 0;
}

```

点赞  
收藏  
分享

Alex\_McAvoy

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

### 1.1.3.3.1.9 二叉树输出

#### 二叉树输出

原创 Alex\_McAvoy 发布于 2018-06-11 20:49:56 阅读数 693 收藏  
展开

##### 【题目描述】

树的凹入表示法主要用于树的屏幕或打印输出，其表示的基本思想是兄弟间等长，一个结点的长度要不小于其子结点的长度。二叉树也可以这样表示，假设叶结点的长度为 1，一个非叶结点的长度等于它的左右子树的长度之和。

一棵二叉树的一个结点用一个字母表示（无重复），输出时从根结点开始：

每行输出若干个结点字符（相同字符的个数等于该结点长度），如果该结点有左子树就递归输出左子树；如果该结点有右子树就递归输出右子树。

假定一棵二叉树一个结点用一个字符描述，现在给出先序和中序遍历的字符串，用树的凹入表示法输出该二叉树。

##### 【输入】

两行，每行是由字母组成的字符串（一行的每个字符都是唯一的），分别表示二叉树的先序遍历和中序遍历的序列。

##### 【输出】

行数等于该树的结点数，每行的字母相同。

**【输入样例】**

ABCDEFG

CBDAFEG

**【输出样例】**

AAAA

BB

C

D

EE

F

G

**【源程序】**

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
char a[N],b[N],c[N];
int calculate(int left1,int right1,int left2,int right2)
{
    if(left1==right1)
    {
        c[left1]=1;
        return c[left1];
    }
    int i;
    for(i=left2;i<=right2;i++)
        if(a[left1]==b[i])
            break;
    if(left2<i)
        c[left1]+=calculate(left1+1,(i-1-left2)+(left1+1),left2,i-1);
    if(i<right2)
        c[left1]+=calculate(right1-(right2-(i+1)),right1,i+1,right2);
    return c[left1];
}

```



```

}
int main()
{
    cin>>a>>b;
    int lena=strlen(a);
    int lenb=strlen(b);
    calculate(0,lena-1,0,lenb-1);
    for(int i=0;i<lena;i++)
    {
        for(int j=1;j<=c[i];j++)
            cout<<a[i];
        cout<<endl;
    }
    return 0;
}

```

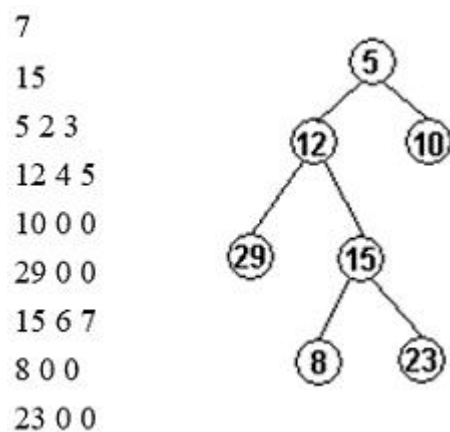
### 1.1.3.3.1.10 T1367 查找二叉树

#### T1367 查找二叉树

原创 Alex\_McAvoy 发布于 2018-06-11 20:48:32 阅读数 857 收藏  
展开

##### 【题目描述】

已知一棵二叉树用邻接表结构存储，中序查找二叉树中值为  $x$  的结点，并指出是第几个结点。例：如图二叉树的数据文件的数据格式如下：



**【输入】**

第一行  $n$  为二叉树的结点个数， $n \leq 100$ ；第二行  $x$  表示要查找的结点的值；以下第一列数据是各结点的值，第二列数据是左儿子结点编号，第三列数据是右儿子结点编号。

**【输出】**

一个数即查找的结点编号。

**【输入样例】**

```
7
15
5 2 3
12 4 5
10 0 0
29 0 0
15 6 7
8 0 0
23 0 0
```

**【输出样例】**

```
4
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int d;
    int left;
    int right;
}q[N];
int n,a;
int ans;
void inorder(int k)
```

```

{
    if(q[k].left)
        inorder(q[k].left);
    ans++;
    if(q[k].d==a)
    {
        cout<<ans<<endl;
        exit(0);
    }
    if(q[k].right)
        inorder(q[k].right);
}
int main()
{
    cin>>n>>a;
    for(int i=1;i<=n;i++)
        cin>>q[i].d>>q[i].left>>q[i].right;
    inorder(1);
    return 0;
}

```

### 1.1.3.3.1.11 T1368 对称二叉树

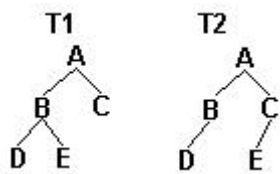
T1368 对称二叉树

原创 Alex\_McAvoy 发布于 2018-06-11 20:47:21 阅读数 992 收藏  
展开

#### 【题目描述】

如果二叉树的左右子树的结构是对称的，即两棵子树皆为空，或者皆不空，则称该二叉树是对称的。编程判断给定的二叉树是否对称。

例：如下图中的二叉树 T1 是对称的，T2 是不对称的。



二叉树用顺序结构给出，若读到#则为空，二叉树 T1=ABCDE，T2=ABCD#E，如果二叉树是对称的，输出“Yes”，反之输出“No”。

#### 【输入】

二叉树用顺序结构给出，若读到#则为空。

#### 【输出】

如果二叉树是对称的，输出“Yes”，反之输出“No”。

#### 【输入样例】

ABCDE

#### 【输出样例】

Yes

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
char s[N];
int main()
{
    cin>>s;
    int len=strlen(s);
    s[len]='#';

    int flag=true;
    for(int i=1;i<len;i+=2)
        if( (s[i]=='#'&& s[i+1]!='#') || (s[i+1]=='#'&& s[i]!='#') )
        {
            flag=false;
            break;
        }
}
```

```

if(flag)
    cout<<"Yes"<<endl;
else
    cout<<"No"<<endl;
return 0;
}

```

### 1.1.3.3.2 第二节 堆及其应用

#### 1.1.3.3.2.1 T1369 合并果子

##### T1369 合并果子

原创 Alex\_McAvoy 发布于 2018-06-11 20:46:05 阅读数 605 收藏  
展开

##### 【题目描述】

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过  $n-1$  次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力  $= 3 + 12 = 15$ 。可以证明 15 为最小的体力耗费值。

##### 【输入】

两行，第一行是一个整数  $n$  ( $1 \leq n \leq 30000$ )，表示果子的种类数。第二行包含  $n$  个整数，用空格分隔，第  $i$  个整数  $a_i$  ( $1 \leq a_i \leq 20000$ ) 是第  $i$  种果子的数目。

**【输出】**

一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于231231。

**【输入样例】**

3

1 2 9

**【输出样例】**

15

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
int a[N];
void quick_sort(int left,int right)
{
    int i=left,j=right;
    int mid=a[(i+j)/2];
    while(i<=j)
    {
        while(a[i]>mid)
            i++;
        while(a[j]<mid)
            j--;
        if(i<=j)
        {
            swap(a[i],a[j]);
            i++;
            j--;
        }
    }
    if(left<j)
        quick_sort(left,j);
```

```

        if(i<right)
            quick_sort(i,right);
    }

int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i];

    quick_sort(1,n);

    int ans=0;
    int i,j;
    while(1)
    {
        int t=a[n-1]+a[n];
        ans=ans+t;
        if(n==2)
            break;
        if(a[n-2]>=t)
            a[n-1]=t;
        else
        {
            j=n-2;
            while(j>=1&& a[j]<t)
                j--;
            for(i=n-1;i>j+1;i--)
                a[i]=a[i-1];
            a[i]=t;
        }
        n--;
    }
    cout<<ans<<endl;
    return 0;
}

```

### 1.1.3.3.2.2 T1370 最小函数值

#### T1370 最小函数值

原创 Alex\_McAvoy 发布于 2018-06-11 20:44:55 阅读数 797 收藏  
展开

### 【题目描述】

有  $n$  个函数，分别为  $F_1, F_2, \dots, F_n$ 。定义  $F_i(x) = A_i x^2 + B_i x + C_i (x \in \mathbb{N}^?)$ 。给定这些  $A_i$ 、 $B_i$  和  $C_i$ ，请求出所有函数的所有函数值中最小的  $m$  个（如有重复的要输出多个）。

### 【输入】

第一行输入两个正整数  $n$  和  $m$ 。

以下  $n$  行每行三个正整数，其中第  $i$  行的三个数分别位  $A_i$ 、 $B_i$  和  $C_i$ 。输入数据保证  $A_i \leq 10$ ， $B_i \leq 100$ ， $C_i \leq 10000$ 。

### 【输出】

将这  $n$  个函数所有可以生成的函数值排序后的前  $m$  个元素。这  $m$  个数应该输出到一行，用空格隔开。

### 【输入样例】

```
3 10
4 5 3
3 4 5
1 7 1
```

### 【输出样例】

```
9 12 12 19 25 29 31 44 45 54
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int a[N],b[N],c[N];
int f[N];
int main()
```



```

{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i]>>b[i]>>c[i];
        f[i]=1;
    }
    int minn;
    for(int i=1;i<=m;i++)
    {
        int temp=INF;
        for(int j=1;j<=n;j++)
        {
            if(a[j]*f[j]*f[j]+b[j]*f[j]+c[j]<temp)
            {
                temp=a[j]*f[j]*f[j]+b[j]*f[j]+c[j];
                minn=j;
            }
        }
        cout<<a[minn]*f[minn]*f[minn]+b[minn]*f[minn]+c[minn]<<" ";
        f[minn]++;
    }
    return 0;
}

```

### 1.1.3.3.2.3 T1371 看病

#### T1371 看病

原创 Alex\_McAvoy 发布于 2018-06-11 20:43:15 阅读数 1053 收藏  
展开

##### 【题目描述】

有个朋友在医院工作，想请 **BSNY** 帮忙做个登记系统。具体是这样的，最近来医院看病的人越来越多了，因此很多人要排队，只有当空闲时放一批病人看病。但医院的排队不同其他排队，因为多数情况下，需要病情严重的人优先看病，所以希望 **BSNY** 设计系统时，以病情的严重程度作为优先级，判断接下来谁可以去看病。

##### 【输入】

第一行输入  $n$ ，表示有  $n$  个操作。

对于每个操作，首先输入 `push` 或 `pop`。

`push` 的情况，之后会输入 `ai` 和 `bi`，分别表示患者姓名和患者病情优先级。

`pop` 后面没有输入，但需要你输出。

#### 【输出】

对于 `pop` 的操作，输出此时还在排队人中，优先级最大的患者姓名和优先级。

表示他可以进去看病了。

如果此时没人在排队，那么输出“none”，具体可见样例。

#### 【输入样例】

```
7
pop
push bob 3
push tom 5
push ella 1
pop
push zkw 4
pop
```

#### 【输出样例】

```
none
tom 5
zkw 4
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
struct Node{
    char name[20];
```

```

    int grade;
}a,heap[N],temp;
int len=0;
void build(Node hp)
{
    len++;
    int t=len;
    while( t>1 && heap[t/2].grade<hp.grade)
    {
        heap[t]=heap[t/2];
        t/=2;
    }
    heap[t]=hp;
}
void trun(int x)
{
    int left=2*x,right=2*x+1;
    int k=x;

    if(left<=len)
        if(heap[k].grade<=heap[left].grade)
            k=left;

    if(right<=len)
        if(heap[k].grade<=heap[right].grade)
            k=right;

    if(x!=k)
    {
        temp=heap[k];
        heap[k]=heap[x];
        heap[x]=temp;
        trun(k);
    }
}
void Delete()
{
    heap[1]=heap[len];
    len--;
    trun(1);
}
int main()
{
    int t;

```

```

cin>>t;
while(t--)
{
    char operate[10];
    cin>>operate;

    if(strcmp(operate,"push")==0)
    {
        cin>>a.name>>a.grade;
        build(a);
    }

    if(strcmp(operate,"pop")==0)
    {
        if(len==0)
            cout<<"none"<<endl;
        else
        {
            cout<<heap[1].name<<" "<<heap[1].grade<<endl;
            Delete();
        }
    }
}
return 0;
}

```

### 1.1.3.3.2.4 T1372 小明的账单

#### T1372 小明的账单

原创 Alex\_McAvoy 发布于 2018-06-11 20:41:38 阅读数 1348 收藏  
展开

##### 【题目描述】

小明在一次聚会中，不慎遗失了自己的钱包，在接下来的日子，面对小明的将是一系列的补卡手续和堆积的账单... 在小明的百般恳求下，老板最终同意延缓账单的支付时间。可老板又提出，必须从目前还没有支付的所有账单中选出面额最大和最小的两张，并把他们付清。还没有支付的账单会被保留到下一天。 请你帮他计算出支付的顺序。

##### 【输入】

第 1 行：一个正整数  $N(N \leq 15,000)$ ，表示小明补办银联卡总共的天数。

第 2 行到第  $N+1$  行：每一行描述一天中收到的帐单。先是一个非负整数  $M \leq 100$ ，表示当天收到的账单数，后跟  $M$  个正整数（都小于 1,000,000,000），表示每张帐单的面额。

输入数据保证每天都可以支付两张帐单。

#### 【输出】

输出共  $N$  行，每行两个用空格分隔的整数，分别表示当天支付的面额最小和最大的支票的面额。

#### 【输入样例】

```
4
3 3 6 5
2 8 2
3 7 1 7
0
```

#### 【输出样例】

```
3 6
2 8
1 7
5 7
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
multiset<int> st;//定义 int 类型的 set
int main()
{
    int n;
    cin>>n;
```

```

st.clear();
for(int i=1;i<=n;i++)
{
    int m;
    cin>>m;
    for(int j=1;j<=m;j++)
    {
        int a;
        cin>>a;
        st.insert(a);//插入元素
    }

    cout<<*st.begin()<<" ";//输出最小元素
    st.erase(st.begin());//删除 set 中最小元素
    cout<<*--st.end()<<endl;//输出最大元素
    st.erase(--st.end());//删除 set 中最大元素
}

return 0;
}

```

### 1.1.3.3.2.5 T1373 鱼塘钓鱼

#### T1373 鱼塘钓鱼

原创 Alex\_McAvoy 发布于 2018-06-11 20:40:23 阅读数 609 收藏  
展开

##### 【题目描述】

有  $N$  个鱼塘排成一行 ( $N < 100$ )，每个鱼塘中有一定数量的鱼，例如：  $N=5$  时，如下表：

鱼塘编号

每1分钟能钓到的鱼的数量（1..1000）

每1分钟能钓鱼数的减少量（1..100）

当前鱼塘到下一个相邻鱼塘需要的时间

即：在第1个鱼塘中钓鱼第1分钟内可钓到10条鱼，第2分钟内只能钓到8条鱼，……，第5分钟以后再也钓不到鱼了。从第1个鱼塘到第2个鱼塘需要3分钟，从第2个鱼塘到第3个鱼塘需要5分钟，……

给出一个截止时间  $T(T < 1000)$ ，设计一个钓鱼方案，从第1个鱼塘出发，希望能钓到最多的鱼。

假设能钓到鱼的数量仅和已钓鱼的次数有关，且每次钓鱼的时间都是整数分钟。

#### 【输入】

共5行，分别表示：

第1行为  $N$ ；

第2行为第1分钟各个鱼塘能钓到的鱼的数量，每个数据之间用一空格隔开；

第3行为每过1分钟各个鱼塘钓鱼数的减少量，每个数据之间用一空格隔开；

第4行为当前鱼塘到下一个相邻鱼塘需要的时间；

第5行为截止时间  $T$ 。

#### 【输出】

一个整数（不超过  $2^{31}-1$ ），表示你的方案能钓到的最多的鱼。

#### 【输入样例】

```
5
10 14 20 16 9
2 4 6 5 3
```

3 5 4 4

14

【输出样例】

76

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
#define fish first
#define lake second
priority_queue <pair<int, int> > heap;//pair 表示先按照第一个元素降序排序，第一个
元素相等时按照第二个元素降序排序
int t[N],f[N],d[N];
int main()
{
    int n,T;
    cin>>n;
    for(inti=1;i<=n;i++)//第 1 分钟各个鱼塘能钓到的鱼的数量
        cin>>f[i];
    for(inti=1;i<=n;i++)//每过 1 分钟各个鱼塘钓鱼数的减少量
        cin>>d[i];
    for(int i=1;i<n;i++)//当前鱼塘到下一个相邻鱼塘需要的时间
        cin>>t[i];
    cin>>T;//截止时间

    int maxx=-INF;
    int walk_time=0;
    for(intk=1;k<=n;k++)//枚举最远走到的池塘的编号
    {
        int rest_time=T-walk_time; //计算剩余时间
        int ans=0;
```



```

for(int i=1;i<=k;i++)//收集能够钓鱼的池塘的资料
    heap.push(make_pair(f[i],i));
while(rest_time>0&&heap.top().fish>0)
{
    pair<int,int> a=heap.top();
    heap.pop();
    ans+=a.fish;//贪心选取鱼最多的池塘
    a.fish-=d[a.lake];//修改鱼的数量
    heap.push(a);//堆维护
    rest_time--;//剩余时间变少
}
if(ans>maxx)//刷新最优解
    maxx=ans;
walk_time+=t[k]; //累计走路需要的时间
}
cout<<maxx<<endl;
return 0;
}

```

## 1.1.3.4 第四章 图论算法

### 1.1.3.4.1 第一节 图的遍历

#### 1.1.3.4.1.1 一笔画问题

##### 一笔画问题

原创 Alex\_McAvoy 发布于 2018-06-18 02:44:27 阅读数 1495 收藏  
展开

##### 【题目描述】

如果一个图存在一笔画，则一笔画的路径叫做欧拉路，如果最后又回到起点，那这个路径叫做欧拉回路。

根据一笔画的两个定理，如果寻找欧拉回路，对任意一个点执行深度优先遍历；找欧拉路，则对一个奇点执行 dfs，时间复杂度为  $O(m+n)$ ， $m$  为边数， $n$  是点数。

**【输入】**

第一行  $n, m$ ，有  $n$  个点， $m$  条边，以下  $m$  行描述每条边连接的两点。

**【输出】**

欧拉路或欧拉回路，输出一条路径即可。

**【输入样例】**

```
5 5
1 2
2 3
3 4
4 5
5 1
```

**【输出样例】**

```
1 5 4 3 2 1
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 123
#define E 1e-6
using namespace std;
int n,m;
int g[N][N];
int dis[N],path[N*2];
int cnt;
void dfs(int i)
{
    for(int j=1;j<=n;j++)
        if(g[i][j])
        {
            g[i][j]=0;
            g[j][i]=0;
```

```

        dfs(j);
    }
    path[cnt++]=i;
}
int main()
{
    cin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        int x,y;
        cin>>x>>y;
        g[x][y]=1;
        g[y][x]=1;
        dis[x]++;
        dis[y]++;
    }

    int start=1;
    int sum=0;
    for(int i=1;i<=n;i++)
        if(dis[i]%2)
        {
            sum++;
            if(sum==1)
                start=i;
        }

    dfs(start);

    for(int i=cnt-1;i>=0;i--)
        cout<<path[i]<<" ";
    return 0;
}

```

点赞

### 1.1.3.4.1.2 铲雪车

#### 铲雪车

原创 Alex\_McAvoy 发布于 2018-06-18 03:02:16 阅读数 1898 收藏  
展开

**【题目描述】**

随着白天越来越短夜晚越来越长，我们不得不考虑铲雪问题了。整个城市所有的道路都是双车道，因为城市预算的削减，整个城市只有 1 辆铲雪车。铲雪车只能把它开过的地方（车道）的雪铲干净，无论哪儿有雪，铲雪车都得从停放的地方出发，游历整个城市的街道。现在的问题是：最少要花多少时间去铲掉所有道路上的雪呢？

**【输入】**

输入数据的第 1 行表示铲雪车的停放坐标  $(x,y)$ ， $x, y$  为整数，单位为米。下面最多有 100 行，每行给出了一条街道的起点坐标和终点坐标，所有街道都是笔直的，且都是双向一个车道。铲雪车可以在任意交叉口、或任何街道的末尾任意转向，包括转 U 型弯。铲雪车铲雪时前进速度为 20 km/h，不铲雪时前进速度为 50 km/h。

保证：铲雪车从起点一定可以到达任何街道。

**【输出】**

铲掉所有街道上的雪并且返回出发点的最短时间，精确到分种。

**【输入样例】**

```
0 0
0 0 10000 10000
5000 -10000 5000 10000
5000 10000 10000 10000
```

**【输出样例】**

```
3:55
```

思路：对于每一条边都有两个方向可走，即每一个单独的点都至少有偶数条边相连，也就是有 0 个奇点，可从任意一点出发并不重复地遍历每一条路，这个图是欧拉图。统计每一条路的长度除速度 20 千米每小时即答案。

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<stdlib.h>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 123
#define E 1e-6
```

```

using namespace std;
int main()
{
    long long start,endd;
    long long x1,y1,x2,y2;
    double dis=0;

    cin>>start>>endd;
    while(scanf("%lld%lld%lld%lld",&x1,&y1,&x2,&y2)!=EOF)
        dis+=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));

    double ans=dis*2/1000/20;
    long long h=(long long)(ans);
    long long m=(long long)((ans-h)*60+0.5);

    printf("%lld:%02lld\n",h,m);
    return 0;
}

```

### 1.1.3.4.1.3 骑马修栅栏

#### 骑马修栅栏

原创 Alex\_McAvoy 发布于 2018-06-18 02:53:37 阅读数 601 收藏  
展开

##### 【题目描述】

农民 John 每年有很多栅栏要修理。他总是骑着马穿过每一个栅栏并修复它破损的地方。

John 是一个与其他农民一样懒的人。他讨厌骑马，因此从来不两次经过一个一个栅栏。你必须编一个程序，读入栅栏网络的描述，并计算出一条修栅栏的路径，使每个栅栏都恰好被经过一次。John 能从任何一个顶点(即两个栅栏的交点)开始骑马，在任意一个顶点结束。

每一个栅栏连接两个顶点，顶点用 1 到 500 标号(虽然有的农场并没有 500 个顶点)。一个顶点上可连接任意多( $\geq 1$ )个栅栏。所有栅栏都是连通的(也就是你可以从任意一个栅栏到达另外的所有栅栏)。

你的程序必须输出骑马的路径(用路上依次经过的顶点号码表示)。我们如果把输出的路径看成是一个 500 进制的数，那么当存在多组解的情况下，输出 500 进制表示法中最小的一个 (也就是输出第一个数较小的，如果还有多组解，输出第二个数较小的，等等)。 输入数据保证至少有一个解。

**【输入】**

第 1 行:一个整数  $F(1 \leq F \leq 1024)$ , 表示栅栏的数目;

第 2 到  $F+1$  行:每行两个整数  $i, j(1 \leq i, j \leq 500)$  表示这条栅栏连接  $i$  与  $j$  号顶点。

**【输出】**

输出应当有  $F+1$  行, 每行一个整数, 依次表示路径经过的顶点号。注意数据可能有多组解, 但是只有上面题目要求的那一组解是认为正确的。

**【输入样例】**

```
9
1 2
2 3
3 4
4 2
4 5
2 5
5 6
5 7
4 6
```

**【输出样例】**

```
1
2
3
4
2
5
4
6
5
7
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
```

```

#define N 501
#define MOD 123
#define E 1e-6
using namespace std;
int n,m;
int g[N][N];
int dis[N],path[2*N];
int beginn=INF;
int cnt;
void dfs(int i)
{
    for(int j=beginn;j<=n;j++)
        if(g[i][j])
        {
            g[i][j]--;
            g[j][i]--;
            dfs(j);
        }
    path[cnt++]=i;
}
int main()
{
    cin>>m;
    for(int i=1;i<=m;i++)
    {
        int x,y;
        cin>>x>>y;
        g[x][y]++;
        g[y][x]++;
        dis[x]++;
        dis[y]++;
        n=max(n,max(x,y));
        beginn=min(beginn,min(x,y));
    }
    int start=beginn;
    int sum=0;
    for(int i=beginn;i<=n;i++)
        if(dis[i]%2)
        {
            sum++;
            if(sum==1)
                start=i;
        }
    dfs(start);
}

```

```

for(int i=cnt-1;i>=0;i--)
    cout<<path[i]<<endl;
return 0;
}

```

## 1.1.3.4.2 第二节 最短路径算法

### 1.1.3.4.2.1 最短路径问题

#### 最短路径问题

原创 Alex\_McAvoy 发布于 2018-06-20 22:12:55 阅读数 886 收藏  
展开

##### 【题目描述】

平面上有  $n$  个点 ( $n \leq 100$ )，每个点的坐标均在  $-10000 \sim 10000$  之间。其中的一些点之间有连线。

若有连线，则表示可从一个点到达另一个点，即两点间有通路，通路的距离为两点间的直线距离。现在的任务是找出从一点到另一点之间的最短路径。

##### 【输入】

共  $n+m+3$  行，其中：

第一行为整数  $n$ 。

第 2 行到第  $n+1$  行（共  $n$  行），每行两个整数  $x$  和  $y$ ，描述了一个点的坐标。

第  $n+2$  行为一个整数  $m$ ，表示图中连线的个数。

此后的  $m$  行，每行描述一条连线，由两个整数  $i$  和  $j$  组成，表示第  $i$  个点和第  $j$  个点之间有连线。

最后一行：两个整数  $s$  和  $t$ ，分别表示源点和目标点。

##### 【输出】

一行，一个实数（保留两位小数），表示从  $s$  到  $t$  的最短路径长度。

##### 【输入样例】

```

5
0 0

```



2 0  
 2 2  
 0 2  
 3 1  
 5  
 1 2  
 1 3  
 1 4  
 2 5  
 3 5  
 1 5

**【输出样例】**

3.14

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int x[N],y[N];
double g[N][N];
double calculate(int x1,int y1,int x2,int y2)
{
    return sqrt((double)(x1-x2)*(x1-x2)+(double)(y1-y2)*(y1-y2));
}
int main()
{
    int n,m;
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>x[i]>>y[i];
    cin>>m;
```

```

for(int i=1;i<=n;i++)
{
    for(int j=1;j<=n;j++)
    {
        g[i][j]=INF;
        g[j][i]=INF;
    }
    g[i][i]=0;
}

for(int i=1;i<=m;i++)
{
    int u,v;
    cin>>u>>v;
    double temp=calculate(x[u],y[u],x[v],y[v]);
    g[u][v]=temp;
    g[v][u]=temp;
}

for(int k=1;k<=n;k++)
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(g[i][j]>g[i][k]+g[k][j])
                g[i][j]=g[i][k]+g[k][j];

int u,v;
cin>>u>>v;
printf("%.2lf",g[u][v]);

return 0;
}

```

点赞

收藏

分享

站内首发文章

**Alex\_McAvoy**

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

### 1.1.3.4.2.2 牛的旅行

#### 牛的旅行

原创 Alex\_McAvoy 发布于 2018-06-20 22:05:00 阅读数 874 收藏  
展开

##### 【题目描述】

农民 John 的农场里有很多牧区。有的路径连接一些特定的牧区。一片所有连通的牧区称为一个牧场。但是就目前而言，你能看到至少有两个牧区不连通。现在，John 想在农场里添加一条路径（注意，恰好一条）。对这条路径有这样的限制：一个牧场的直径就是牧场中最远的两个牧区的距离（本题中所提到的所有距离指的都是最短的距离）。考虑如下的两个牧场，图 1 是有 5 个牧区的牧场，牧区用“\*”表示，路径用直线表示。每一个牧区都有自己的坐标：

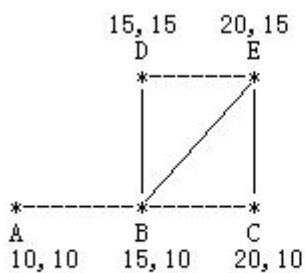


图1

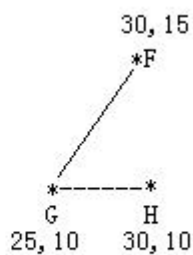


图2

图 1 所示的牧场的直径大约是 12.07106，最远的两个牧区是 A 和 E，它们之间的最短路径是 A-B-E。

这两个牧场都在 John 的农场上。John 将会在两个牧场中各选一个牧区，然后用一条路径连起来，使得连通后这个新的更大的牧场有最小的直径。注意，如果两条路径中途相交，我们不认为它们是连通的。只有两条路径在同一个牧区相交，我们才认为它们是连通的。

现在请你编程找出一条连接两个不同牧场的路径，使得连上这条路径后，这个更大的新牧场有最小的直径。

##### 【输入】

第 1 行：一个整数  $N$  ( $1 \leq N \leq 150$ )，表示牧区数；

第 2 到  $N+1$  行：每行两个整数  $X, Y$  ( $0 \leq X, Y \leq 100000$ )，表示  $N$  个牧区的坐标。每个牧区的坐标都是不一样的。

第  $N+2$  行到第  $2*N+1$  行：每行包括  $N$  个数字（0 或 1）表示一个对称邻接矩阵。

例如，题目描述中的两个牧场的矩阵描述如下：

A B C D E F G H

A 0 1 0 0 0 0 0 0

B 1 0 1 1 1 0 0 0

C 0 1 0 0 1 0 0 0

D 0 1 0 0 1 0 0 0

E 0 1 1 1 0 0 0 0

F 0 0 0 0 0 0 1 0

G 0 0 0 0 0 1 0 1

H 0 0 0 0 0 0 1 0

输入数据中至少包括两个不连通的牧区。

#### 【输出】

只有一行，包括一个实数，表示所求答案。数字保留六位小数。

#### 【输入样例】

```
8
10 10
15 10
20 10
15 15
20 15
30 15
25 10
30 10
01000000
10111000
01001000
01001000
01110000
00000010
00000101
00000010
```

#### 【输出样例】

```
22.071068
```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int x[N],y[N];
char s[N];
double g[150+10][150+10],f[150+10];
double calculate(int x1,int y1,int x2,int y2)
{
    return sqrt((double)(x1-x2)*(x1-x2)+(double)(y1-y2)*(y1-y2));
}
int main()
{
    int n;
    cin>>n;

    for(int i=1;i<=n;i++)
        cin>>x[i]>>y[i];
    for(int i=1;i<=n;i++)
    {
        scanf("%s",s+1);
        for(int j=1;j<=n;j++)
        {
            if(i!=j)
            {
                if(s[j]=='0')
                {
                    g[i][j]=INF;
                    g[j][i]=INF;
                }
            }
            else

```

```

        {
            g[i][j]=calculate(x[i],y[i],x[j],y[j]);
            g[j][i]=calculate(x[i],y[i],x[j],y[j]);
        }
    }
    else
        g[i][j]=0;
}
}

for(int k=1;k<=n;k++)
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(g[i][j]>g[i][k]+g[k][j])
                g[i][j]=g[i][k]+g[k][j];

for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        if(g[i][j]<INF&&g[i][j]>f[i])
            f[i]=g[i][j];

double minn=INF;
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        if( g[i][j]==INF && minn>f[i]+f[j]+calculate(x[i],y[i],x[j],y[j]) )
            minn=f[i]+f[j]+calculate(x[i],y[i],x[j],y[j]);

for(int i=1;i<=n;i++)
    if(minn<f[i])
        minn=f[i];

printf("%.6lf\n",minn);
return 0;
}

```

点赞

### 1.1.3.4.2.3 最小花费

#### 最小花费

原创 Alex\_McAvoy 发布于 2018-06-20 21:54:46 阅读数 795 收藏  
展开

### 【题目描述】

在  $n$  个人中，某些人的银行账号之间可以互相转账。这些人之间转账的手续费各不相同。给定这些人之间转账时需要从转账金额里扣除百分之几的手续费，请问  $A$  最少需要多少钱使得转账后  $B$  收到 100 元。

### 【输入】

第一行输入两个正整数  $n, m$ ，分别表示总人数和可以互相转账的人的对数

以下  $m$  行每行输入三个正整数  $x, y, z$ ，表示标号为  $x$  的人和标号为  $y$  的人之间互相转账需要扣除  $z\%$  的手续费 ( $z < 100$ )。

最后一行输入两个正整数  $A, B$ 。数据保证  $A$  与  $B$  之间可以直接或间接地转账。

### 【输出】

输出  $A$  使得  $B$  到账 100 元最少需要的总费用。精确到小数点后 8 位。

### 【输入样例】

```
3 3
1 2 1
2 3 2
1 3 3
1 3
```

### 【输出样例】

```
103.07153164
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
```

```

using namespace std;
int n,m;
double g[N][N],f[N];
int vis[N];
void Dijkstra()
{
    int sum=1,k;
    while(sum<n)
    {
        double minn=INF;
        for(int i=1;i<=n;i++)
            if(vis[i]==0&&f[i]<minn)
            {
                minn=f[i];
                k=i;
            }
        vis[k]=1;
        sum++;
        for(int i=1;i<=n;i++)
            if(vis[i]==0&&f[i]>f[k]*g[k][i])
                f[i]=f[k]*g[k][i];
    }
}
int main()
{
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(i==j)
                g[i][j]=0;
            else
                g[i][j]=INF;

    for(int i=1;i<=m;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        g[x][y]=100.0/(100-w);
        g[y][x]=100.0/(100-w);
    }

    int A,B;
    cin>>A>>B;
    vis[A]=1;

```



```

for(int i=1;i<=n;i++)
    if(g[A][i])
        f[i]=g[A][i];

Dijkstra();
printf("%.8lf",f[B]*100);
return 0;
}

```

点赞

### 1.1.3.4.2.4 香甜的黄油

#### 香甜的黄油

原创 Alex\_McAvoy 发布于 2018-06-20 21:35:01 阅读数 453 收藏  
展开

##### 【题目描述】

农夫 John 发现做出全威斯康辛州最甜的黄油的方法：糖。把糖放在一片牧场上，他知道  $N$  ( $1 \leq N \leq 500$ ) 只奶牛会过来舔它，这样就能做出能卖好价钱的超甜黄油。当然，他将付出额外的费用在奶牛上。

农夫 John 很狡猾。像以前的巴甫洛夫，他知道他可以训练这些奶牛，让它们在听到铃声时去一个特定的牧场。他打算将糖放在那里然后下午发出铃声，以至他可以在晚上挤奶。

农夫 John 知道每只奶牛都在各自喜欢的牧场（一个牧场不一定只有一头牛）。给出各头牛在的牧场和牧场间的路线，找出使所有牛到达的路程和最短的牧场（他将把糖放在那）。

##### 【输入】

第一行：三个数：奶牛数  $N$ ，牧场数  $P$  ( $2 \leq P \leq 800$ )，牧场间道路数  $C$  ( $1 \leq C \leq 1450$ )。

第二行到第  $N+1$  行：1 到  $N$  头奶牛所在的牧场号。

第  $N+2$  行到第  $N+C+1$  行：每行有三个数：相连的牧场  $A$ 、 $B$ ，两牧场间距 ( $1 \leq D \leq 255$ )，当然，连接是双向的。

##### 【输出】

一行 输出奶牛必须行走的最小的距离和。

##### 【输入样例】

?3 4 5

2  
3  
4  
1 2 1  
1 3 5  
2 3 7  
2 4 3  
3 4 5

【输出样例】

8

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int n,p,c;
int g[N],head[N];
int vis[N],f[N],q[N];
int cnt;
struct node{
    int pre;
    int next;
    int w;
}a[N];
void SPFA(int s)
{
    int headd=1,tail=1;

    q[tail]=s;
    tail++;
```

```

memset(vis,0,sizeof(vis));
for(int i=1;i<=p;i++)
    f[i]=INF;

f[s]=0;
vis[s]=1;

while(headd<tail)
{
    int u=q[headd];
    for(int b=head[u];b;b=a[b].next)
    {
        int v=a[b].pre;
        int w=a[b].w;
        if(f[v]>f[u]+w)
        {
            f[v]=f[u]+w;
            if(vis[v]==0)
            {
                q[tail]=v;
                tail++;
                vis[v]=1;
            }
        }
    }
    vis[u]=0;
    headd++;
}
}

void add_edge(int x,int y,int w)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].w=w;
    a[cnt].next=head[x];
    head[x]=cnt;
}

int main()
{
    cin>>n>>p>>c;
    for(int i=1;i<=n;i++)
        cin>>g[i];
    for(int i=1;i<=c;i++)

```

```

{
    int x,y,w;
    cin>>x>>y>>w;
    add_edge(x,y,w);
    add_edge(y,x,w);
}

int sum=0;
int minn=INF;
for(int i=1;i<=p;i++)
{
    SPFA(i);
    sum=0;
    for(int j=1;j<=n;j++)
        sum+=f[g[j]];
    minn=min(sum,minn);
}
cout<<minn<<endl;
return 0;
}

```

### 1.1.3.4.2.5 信使

#### 信使

原创 Alex\_McAvoy 发布于 2018-06-20 21:14:44 阅读数 1139 收藏  
展开

##### 【题目描述】

战争时期，前线有  $n$  个哨所，每个哨所可能会与其他若干个哨所之间有通信联系。信使负责在哨所之间传递信息，当然，这是要花费一定时间的（以天为单位）。指挥部设在第一个哨所。当指挥部下达一个命令后，指挥部就派出若干个信使向与指挥部相连的哨所送信。当一个哨所接到信后，这个哨所内的信使们也以同样的方式向其他哨所送信。直至所有  $n$  个哨所全部接到命令后，送信才算成功。因为准备充足，每个哨所内都安排了足够的信使（如果一个哨所与其他  $k$  个哨所有通信联系的话，这个哨所内至少会配备  $k$  个信使）。

现在总指挥请你编一个程序，计算出完成整个送信过程最短需要多少时间。

##### 【输入】

第 1 行有两个整数  $n$  和  $m$ ，中间用 1 个空格隔开，分别表示有  $n$  个哨所和  $m$  条通信线路，且  $1 \leq n \leq 100$ 。

第 2 至  $m+1$  行：每行三个整数  $i$ 、 $j$ 、 $k$ ，中间用 1 个空格隔开，表示第  $i$  个和第  $j$  个哨所之间存在通信线路，且这条线路要花费  $k$  天。

**【输出】**

一个整数，表示完成整个送信过程的最短时间。如果不是所有的哨所都能收到信，就输出-1。

**【输入样例】**

```
4 4
1 2 4
2 3 7
2 4 1
3 4 6
```

**【输出样例】**

```
11
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N];
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            if(i==j)
                g[i][j]=0;
```

```

        else
            g[i][j]=INF;
    }
}

for(int i=1;i<=m;i++)
{
    int x,y,w;
    cin>>x>>y>>w;
    g[x][y]=w;
    g[y][x]=w;
}

for(int k=1;k<=n;k++)
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(g[i][j]>g[i][k]+g[k][j])
                g[i][j]=g[i][k]+g[k][j];

int maxx=-INF;
for(int i=1;i<=n;i++)
    if(g[1][i]>maxx)
        maxx=g[1][i];

if(maxx==INF)
    cout<<"-1"<<endl;
else
    cout<<maxx<<endl;

return 0;
}

```

### 1.1.3.4.2.6 最优乘车

#### 最优乘车

原创 Alex\_McAvoy 发布于 2018-06-25 00:50:19 阅读数 1166 收藏  
展开

##### 【题目描述】

H 城是一个旅游胜地，每年都有成千上万的人前来观光。为方便游客，巴士公司在各个旅游景点及宾馆，饭店等地都设置了巴士站并开通了一些单程巴士线路。每条单程巴士线路从某个巴士站出发，依次途经若干个巴士站，最终到达终点巴士站。

一名旅客最近到 H 城旅游，他很想去看 S 公园游玩，但如果从他所在的饭店没有一路巴士可以直接到达 S 公园，则他可能要先乘某一路巴士坐几站，再下来换乘同一站台的另一路巴士，这样换乘几次后到达 S 公园。

现在用整数  $1, 2, \dots, N$  给 H 城的所有的巴士站编号，约定这名旅客所在饭店的巴士站编号为 1，S 公园巴士站的编号为 N。

写一个程序，帮助这名旅客寻找一个最优乘车方案，使他在从饭店乘车到 S 公园的过程中换车的次数最少。

#### 【输入】

第一行有两个数字 M 和 N ( $1 \leq M \leq 100$   $1 < N \leq 500$ )，表示开通了 M 条单程巴士线路，总共有 N 个车站。从第二行到第 M 行依次给出了第 1 条到第 M 条巴士线路的信息。其中第 i+1 行给出的是第 i 条巴士线路的信息，从左至右按运行顺序依次给出了该线路上的所有站号相邻两个站号之间用一个空格隔开。

#### 【输出】

只有一行。如果无法乘巴士从饭店到达 S 公园，则输出 "NO"，否则输出你的程序所找到的最少换车次数，换车次数为 0 表示不需换车即可到达。

#### 【输入样例】

```
3 7
6 7
4 7 3 6
2 1 3 5
```

#### 【输出样例】

```
2
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#include<sstream>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
```

```

#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N];
int a[N];
int main()
{
    int n,m;
    scanf("%d%d\n",&m,&n);

    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            if(i==j)
                g[i][j]=0;
            else
                g[i][j]=INF;
        }

    for(int i=1;i<=m;i++)
    {
        string str;
        getline(cin,str);
        stringstream input(str);//创建一个流
        int len=0;
        int x;
        while(input>>x)//把值传递流中
        {
            len++;
            a[len]=x;//获取转换后的字符转并将其写入 a[len]
        }
        for(int j=1;j<=len;j++)
            for(int k=j+1;k<=len;k++)
                g[a[j]][a[k]]=1;
    }
    for(int k=1;k<=n;k++)
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                g[i][j]=min(g[i][j],g[i][k]+g[k][j]);

    if(g[1][n]>=INF)
        cout<<"NO"<<endl;
    else
        cout<<g[1][n]-1<<endl;
}

```



```
    return 0;
}
```

### 1.1.3.4.2.7 最短路径

#### 最短路径

原创 Alex\_McAvoy 发布于 2018-06-25 00:23:10 阅读数 921 收藏

展开

##### 【题目描述】

给出一个有向图  $G=(V, E)$ ，和一个源点  $v_0 \in V$ ，请写一个程序输出  $v_0$  和图  $G$  中其它顶点的最短路径。只要所有的有向环权值和都是正的，我们就允许图的边有负值。顶点的标号从 1 到  $n$  ( $n$  为图  $G$  的顶点数)。

##### 【输入】

第 1 行：一个正数  $n$  ( $2 \leq n \leq 80$ )，表示图  $G$  的顶点总数。

第 2 行：一个整数，表示源点  $v_0$  ( $v_0 \in V$ ， $v_0$  可以是图  $G$  中任意一个顶点)。

第 3 至第  $n+2$  行，用一个邻接矩阵  $W$  给出了这个图。

##### 【输出】

共包含  $n-1$  行，按照顶点编号从小到大的顺序，每行输出源点  $v_0$  到一个顶点的最短距离。每行的具体格式参照样例。

##### 【输入样例】

```
5
1
0 2 - - 10
- 0 3 - 7
- - 0 4 -
- - - 0 5
- - 6 - 0
```

##### 【输出样例】

```
(1 -> 2) = 2
(1 -> 3) = 5
(1 -> 4) = 9
(1 -> 5) = 9
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
```

```

#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
int a[N][N];
int main()
{
    int n,s;
    cin>>n>>s;
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            int b;
            if(scanf("%d",&b)==1)//如果 b 是整数
                a[i][j]=b;
            else
                a[i][j]=INF;
        }
    }

    for(int k=1;k<=n;k++)
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                if(a[i][j]>a[i][k]+a[k][j])
                    a[i][j]=a[i][k]+a[k][j];

    for(int i=1;i<=n;i++)
        if(i!=s)
            printf("(%d -> %d) = %d\n",s,i,a[s][i]);

    return 0;
}

```

### 1.1.3.4.2.8 热浪

#### 热浪

原创 Alex\_McAvoy 发布于 2018-06-25 00:52:26 阅读数 518 收藏  
展开

##### 【题目描述】

德克萨斯纯朴的民眾们这个夏天正在遭受巨大的热浪!!! 他们的德克萨斯长角牛吃起来不错, 可是他们并不是很擅长生产富含奶油的乳製品。Farmer John 此时以先天下之忧而忧, 后天下之乐而乐的精神, 身先士卒地承担起向德克萨斯运送大量的营养冰凉的牛奶的重任, 以减轻德克萨斯人忍受酷暑的痛苦。

FJ 已经研究过可以把牛奶从威斯康星运送到德克萨斯州的路线。这些路线包括起始点和终点先一共经过  $T$  ( $1 \leq T \leq 2,500$ ) 个城镇, 方便地标号为 1 到  $T$ 。除了起点和终点外的每个城镇由两条双向道路连向至少两个其它的城镇。每条道路有一个通过费用 (包括油费, 过路费等等)。

给定一个地图, 包含  $C$  ( $1 \leq C \leq 6,200$ ) 条直接连接 2 个城镇的道路。每条道路由道路的起点  $R_s$ , 终点  $R_e$  ( $1 \leq R_s \leq T; 1 \leq R_e \leq T$ ), 和花费 ( $1 \leq C_i \leq 1,000$ ) 组成。求从起始的城镇  $T_s$  ( $1 \leq T_s \leq T$ ) 到终点的城镇  $T_e$  ( $1 \leq T_e \leq T$ ) 最小的总费用。

##### 【输入】

第一行: 4 个由空格隔开的整数:  $T, C, T_s, T_e$ ;

第 2 到第  $C+1$  行: 第  $i+1$  行描述第  $i$  条道路。有 3 个由空格隔开的整数:  $R_s, R_e$  和  $C_i$ 。

##### 【输出】

一个单独的整数表示从  $T_s$  到  $T_e$  的最小总费用。数据保证至少存在一条道路。

##### 【输入样例】

```
7? 11 5 4
2 4 2
1 4 3
7 2 2
3 4 3
5 7 5
7 3 3
6 1 1
6 3 4
2 4 3
5 6 3
7 2 1
```

## 【输出样例】

7

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 3001
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int pre;
    int next;
    int w;
}a[N*10];
int head[N],vis[N],f[N];
int cnt;
void add(int x,int y,int w)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].next=head[x];
    a[cnt].w=w;
    head[x]=cnt;

    cnt++;
    a[cnt].pre=x;
    a[cnt].next=head[y];
    a[cnt].w=w;
    head[y]=cnt;
}
int main()
{
    int t,c,start,endd;
    cin>>t>>c>>start>>endd;

```

```

for(int i=1;i<=c;i++)
{
    int x,y,w;
    cin>>x>>y>>w;
    add(x,y,w);
}
memset(f,INF,sizeof(f));
f[start]=0;

for(int i=1;i<=t;i++)
{
    int x=0;
    int minn=INF;
    for(int j=1;j<=t;j++)
        if(vis[j]==0&&f[j]<minn)
        {
            minn=f[j];
            x=j;
        }
    vis[x]=1;

    int k=head[x];
    while(k!=0)
    {
        int y=a[k].pre;
        if(vis[y]==0&&f[x]+a[k].w<f[y])
            f[y]=f[x]+a[k].w;
        k=a[k].next;
    }
}
cout<<f[endd]<<endl;
return 0;
}

```

### 1.1.3.4.2.9 分糖果

#### 分糖果

原创 Alex\_McAvoy 发布于 2018-06-25 00:51:19 阅读数 910 收藏  
展开

#### 【题目描述】

童年的我们，将和朋友分享美好的事物作为自己的快乐。这天，C 小朋友得到了 Plenty

of candies, 将要把这些糖果分给要好的朋友们。已知糖果从一个人传给另一个人需要 1 秒的时间, 同一个小朋友不会重复接受糖果。由于糖果足够多, 如果某时刻某小朋友接受了糖果, 他会将糖果分成若干份, 分给那些在他身旁且还没有得到糖果的小朋友们, 而且自己会吃一些糖果。由于嘴馋, 小朋友们等不及将糖果发完, 会在得到糖果后边吃边发。每个小朋友从接受糖果到吃完糖果需要  $m$  秒的时间。那么, 如果第一秒 C 小朋友开始发糖, 第多少秒所有小朋友都吃完了糖呢?

#### 【输入】

第一行为三个数  $n$ 、 $p$ 、 $c$ , 为小朋友数、关系数和 C 小朋友的编号。

第二行为一个数  $m$ , 表示小朋友吃糖的时间。

下面  $p$  行每行两个整数, 表示某两个小朋友在彼此身旁。

#### 【输出】

一个数, 为所有小朋友都吃完了糖的时间。

#### 【输入样例】

```
?4 3 1
2
1 2
2 3
1 4
```

#### 【输出样例】

```
5
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10010
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int pre;
```

```

    int next;
    int w;
}a[N*20];
int cnt;
int head[N],q[N*10];
int vis[N],f[N];
void add(int x,int y,int w)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].next=head[x];
    a[cnt].w=w;
    head[x]=cnt;

    cnt++;
    a[cnt].pre=x;
    a[cnt].next=head[y];
    a[cnt].w=w;
    head[y]=cnt;
}

int main()
{
    int n,m,c,p;
    cin>>n>>p>>c>>m;
    for(int i=1;i<=p;i++)
    {
        int x,y;
        cin>>x>>y;
        add(x,y,1);
    }

    memset(f,INF,sizeof(f));
    f[c]=0;
    vis[c]=1;

    int headd=1,tail=1;
    q[tail]=c;
    tail++;

    while(headd<tail)
    {
        int x=q[headd];
        int k=head[x];

```

```

while(k!=0)
{
    int y=a[k].pre;
    if(f[y]>f[x]+a[k].w)
    {
        f[y]=f[x]+a[k].w;
        if(vis[y]==0)
        {
            q[tail]=y;
            vis[y]=1;
            tail++;
        }
    }
    k=a[k].next;
}
headd++;
}

int maxx=-INF;
for(int i=1;i<=n;i++)
    maxx=max(maxx,f[i]);
cout<<maxx+m+1<<endl;
return 0;
}

```

### 1.1.3.4.2.10 城市路

#### 城市路

原创 Alex\_McAvoy 发布于 2018-06-25 00:50:02 阅读数 854 收藏  
展开

##### 【题目描述】

罗老师被邀请参加一个舞会，是在城市  $n$ ，而罗老师当前所处的城市为  $1$ ，附近还有很多城市  $2 \sim n-1$ ，有些城市之间没有直接相连的路，有些城市之间有直接相连的路，这些路都是双向的，当然也可能有多条。

现在给出直接相邻城市的路长度，罗老师想知道从城市  $1$  到城市  $n$ ，最短多少距离。

##### 【输入】

输入  $n, m$ ，表示  $n$  个城市和  $m$  条路；

接下来  $m$  行，每行  $a \ b \ c$ ，表示城市  $a$  与城市  $b$  有长度为  $c$  的路。



**【输出】**

输出 1 到 n 的最短路。如果 1 到达不了 n，就输出-1。

**【输入样例】**

```
?5 5
1 2 20
2 3 30
3 4 20
4 5 20
1 5 100
```

**【输出样例】**

```
90
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 3001
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int pre;
    int next;
    int w;
}a[N*10];
int n,m;
int cnt;
int head[N],vis[N],f[N];
void add(int x,int y,int w)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].next=head[x];
    a[cnt].w=w;
```

```

head[x]=cnt;

cnt++;
a[cnt].pre=x;
a[cnt].next=head[y];
a[cnt].w=w;
head[y]=cnt;
}

int main()
{
    cin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        add(x,y,w);
    }

    memset(f,INF,sizeof(f));
    f[1]=0;
    vis[1]=1;

    int x=head[1];
    while(x!=0)
    {
        int y=a[x].pre;
        if(f[y]>a[x].w)
            f[y]=a[x].w;
        x=a[x].next;
    }

    int cnt=0;
    while(cnt<n)
    {
        cnt++;
        int k;
        int minn=INF;
        for(int i=1;i<=n;i++)
            if(vis[i]==0&&f[i]<minn)
            {
                minn=f[i];
                k=i;
            }
    }

```

```

vis[k]=1;

int x=head[k];
while(x!=0)
{
    int y=a[x].pre;
    int w=a[x].w;
    if(vis[y]==0&&f[y]>f[k]+w)
        f[y]=f[k]+w;
    x=a[x].next;
}
}

if(f[n]==INF)
    cout<<"-1"<<endl;
else
    cout<<f[n]<<endl;
return 0;
}

```

### 1.1.3.4.2.11 最短路

#### 最短路

原创 Alex\_McAvoy 发布于 2018-06-25 00:48:34 阅读数 1280 收藏  
展开

##### 【题目描述】

给定  $M$  条边， $N$  个点的带权无向图。求 1 到  $N$  的最短路。

##### 【输入】

第一行：  $N, M (N \leq 100000, M \leq 500000)$

接下来  $M$  行 3 个正整数：  $a_i, b_i, c_i$  表示  $a_i, b_i$  之间有一条长度为  $c_i$  的路，  $c_i \leq 1000$ 。

##### 【输出】

一个整数，表示 1 到  $N$  的最短距离。

##### 【输入样例】

```

4 4
1 2 1
2 3 1
3 4 1
2 4 1

```

## 【输出样例】

2

## 【源程序】

```

#include<iostream>
#include<cstdio>
#define INF 0x3f3f3f3f
#define N 100100
using namespace std;
struct node{
    int pre;
    int next;
    int w;
}a[N*10];
int q[N*10],head[N];
int vis[N],f[N];
int cnt;
void add(int x,int y,int w)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].next=head[x];
    a[cnt].w=w;
    head[x]=cnt;

    cnt++;
    a[cnt].pre=x;
    a[cnt].next=head[y];
    a[cnt].w=w;
    head[y]=cnt;
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        add(x,y,w);
    }

    memset(f,INF,sizeof(f));
    f[1]=0;
    vis[1]=1;

```

```

int headd=1,tail=1;
q[tail]=1;
tail++;
while(headd<tail)
{
    int x=q[headd];
    int temp=head[x];
    while(temp!=0)
    {
        int y=a[temp].pre;
        if(f[y]>f[x]+a[temp].w)
        {
            f[y]=f[x]+a[temp].w;
            if(vis[y]==0)
            {
                vis[y]=1;
                q[tail]=y;
                tail++;
            }
        }
        temp=a[temp].next;
    }
    vis[x]=0;
    headd++;
}
cout<<f[n]<<endl;
return 0;
}

```

### 1.1.3.4.3 第三节 图的连通性问题

#### 第三节 图的连通性问题

#### 1.1.3.4.3.1 刻录光盘

##### 刻录光盘

原创 Alex\_McAvoy 发布于 2018-06-18 03:21:31 阅读数 450 收藏  
展开

**【题目描述】**

在 FJOI2010 夏令营快要结束的时候，很多营员提出来要把整个夏令营期间的资料刻录成一张光盘给大家，以便大家回去后继续学习。组委会觉得这个主意不错！可是组委会一时没有足够的空光盘，没法保证每个人都能拿到刻录上资料的光盘，怎么办呢？！

DYJ 分析了一下所有营员的地域关系，发现有些营员是一个城市的，其实他们只需要一张就可以了，因为一个人拿到光盘后，其他人可以带着 U 盘之类的东西去拷贝啊！

他们愿意让一些人到他那儿拷贝资料，当然也可能不愿意让另外一些人到他那儿拷贝资料，这与我们 FJOI 宣扬的团队合作精神格格不入!!!

现在假设总共有  $N$  个营员 ( $2 \leq N \leq 200$ )，每个营员的编号为  $1 \sim N$ 。DYJ 给每个人发了一张调查表，让每个营员填上自己愿意让哪些人到他那儿拷贝资料。当然，如果 A 愿意把资料拷贝给 B，而 B 又愿意把资料拷贝给 C，则一旦 A 获得了资料，则 B，C 都会获得资料。

现在，请你编写一个程序，根据回收上来的调查表，帮助 DYJ 计算出组委会至少要刻录多少张光盘，才能保证所有营员回去后都能得到夏令营资料？

**【输入】**

先是一个数  $N$ ，接下来的  $N$  行，分别表示各个营员愿意把自己获得的资料拷贝给其他哪些营员。即输入数据的第  $i+1$  行表示第  $i$  个营员愿意把资料拷贝给那些营员的编号，以一个 0 结束。如果一个营员不愿意拷贝资料给任何人，则相应的行只有 1 个 0，一行中的若干数之间用一个空格隔开。

**【输出】**

一个正整数，表示最少要刻录的光盘数。

**【输入样例】**

```
5
2 4 3 0
4 5 0
0
0
1 0
```

**【输出样例】**

```
1
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
```

```

#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N],pre[N];
int main()
{
    int n,j;
    cin>>n;
    for(int i=1;i<=n;i++)
        while(scanf("%d",&j)!=EOF && j)
            g[i][j]=1;

    for(int i=1;i<=n;i++)
        for(int k=1;k<=n;k++)
            for(int j=1;j<=n;j++)
                if(g[i][k]&&g[k][j])
                    g[i][j]=1;

    for(int i=1;i<=n;i++)
        pre[i]=i;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            if(g[i][j])
                pre[j]=pre[i];

    int cnt=0;
    for(int i=1;i<=n;i++)
        if(pre[i]==i)
            cnt++;
    cout<<cnt<<endl;
    return 0;
}

```

点赞

### 1.1.3.4.3.2 T1384 珍珠

#### T1384 珍珠

原创 Alex\_McAvoy 发布于 2018-06-18 03:25:52 阅读数 490 收藏  
展开

##### 【题目描述】

有  $n$  颗形状和大小都一致的珍珠，它们的重量都不相同。 $n$  为整数，所有的珍珠从 1 到  $n$  编号。你的任务是发现哪颗珍珠的重量刚好处于正中间，即在所有珍珠的重量中，该珍珠的重量列  $(n+1)/2$  位。下面给出将一对珍珠进行比较的办法：

给你一架天平用来比较珍珠的重量，我们可以比出两个珍珠哪个更重一些，在作出一系列的比较后，我们可以将某些肯定不具备中间重量的珍珠拿走。

例如，下列给出对 5 颗珍珠进行四次比较的情况：

- 1、珍珠 2 比珍珠 1 重
- 2、珍珠 4 比珍珠 3 重
- 3、珍珠 5 比珍珠 1 重
- 4、珍珠 4 比珍珠 2 重

根据以上结果，虽然我们不能精确地找出哪个珍珠具有中间重量，但我们可以肯定珍珠 1 和珍珠 4 不可能具有中间重量，因为珍珠 2、4、5 比珍珠 1 重，而珍珠 1、2、3 比珍珠 4 轻，所以我们可以移走这两颗珍珠。

写一个程序统计出共有多少颗珍珠肯定不会是中间重量。

##### 【输入】

第一行包含两个用空格隔开的整数  $N$  和  $M$ ，其中  $1 \leq N \leq 99$ ，且  $N$  为奇数， $M$  表示对珍珠进行的比较次数，接下来的  $M$  行每行包含两个用空格隔开的整数  $x$  和  $y$ ，表示珍珠  $x$  比珍珠  $y$  重。

##### 【输出】

一行包含一个整数，表示不可能是中间重量的珍珠的总数。

##### 【输入样例】

```
5 4
2 1
4 3
5 1
4 2
```



## 【输出样例】

2

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 501
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N];
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=m;i++)
    {
        int x,y;
        cin>>x>>y;
        g[x][y]=1;
    }

    for(int i=1;i<=n;i++)
        for(int k=1;k<=n;k++)
            for(int j=1;j<=n;j++)
                if(g[k][i]&&g[i][j])
                    g[k][j]=1;

    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            g[0][j]+=g[i][j];
            g[i][0]+=g[i][j];
        }
}

```

```

int cnt=0;
for(int i=1;i<=n;i++)
    if( g[0][i]>n/2 || g[i][0]>n/2 )
        cnt++;

cout<<cnt<<endl;
return 0;
}

```

点赞

### 1.1.3.4.4 第四节 并查集

#### 1.1.3.4.4.1 T1346 亲戚

##### T1346 亲戚

原创 Alex\_McAvoy 发布于 2018-06-26 23:50:45 阅读数 1024 收藏  
展开

##### 【题目描述】

或许你并不知道，你的某个朋友是你的亲戚。他可能是你的曾祖父的外公的女婿的外甥女的表姐的孙子。如果能得到完整的家谱，判断两个人是否是亲戚应该是可行的，但如果两个人的最近公共祖先与他们相隔好几代，使得家谱十分庞大，那么检验亲戚关系实非人力所能及。在这种情况下，最好的帮手就是计算机。为了将问题简化，你将得到一些亲戚关系的信息，如 **Marry** 和 **Tom** 是亲戚，**Tom** 和 **Ben** 是亲戚，等等。从这些信息中，你可以推出 **Marry** 和 **Ben** 是亲戚。请写一个程序，对于我们的关于亲戚关系的提问，以最快的速度给出答案。

##### 【输入】

接下来  $m$  行，每行  $a\ b\ c$ ，表示城市  $a$  与城市  $b$  有长度为  $c$  的路。输入由两部分组成。

第一部分以  $N, M$  开始。 $N$  为问题涉及的人的个数( $1 \leq N \leq 20000$ )。这些人的编号为  $1, 2, 3, \dots, N$ 。下面有  $M$  行( $1 \leq M \leq 1000000$ )，每行有两个数  $a_i, b_i$ ，表示已知  $a_i$  和  $b_i$  是亲戚。

第二部分以  $Q$  开始。以下  $Q$  行有  $Q$  个询问( $1 \leq Q \leq 1000000$ )，每行为  $c_i, d_i$ ，表示询问  $c_i$  和  $d_i$  是否为亲戚。

**【输出】**

对于每个询问  $c_i, d_i$ ，输出一行：若  $c_i$  和  $d_i$  为亲戚，则输出“Yes”，否则输出“No”。

**【输入样例】**

10 7

2 4

5 7

1 3

8 9

1 2

5 6

2 3

3

3 4

7 10

8 9

**【输出样例】**

Yes

No

Yes

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 20001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
```

```

{
    int xx=Find(x);
    int yy=Find(y);
    if(xx!=yy)
        father[xx]=yy;
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=m;i++)
    {
        int x,y;
        cin>>x>>y;
        Union(x,y);
    }

    for(int i=1;i<=n;i++)
        father[i]=Find(i);

    int q;
    cin>>q;
    for(int i=1;i<=q;i++)
    {
        int x,y;
        cin>>x>>y;
        if(father[x]==father[y])
            cout<<"Yes"<<endl;
        else
            cout<<"No"<<endl;
    }
    return 0;
}

```

### 1.1.3.4.4.2 格子游戏

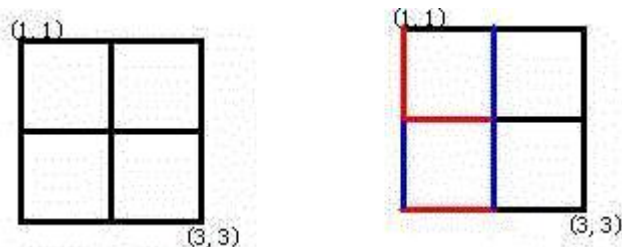
#### 格子游戏

原创 Alex\_McAvoy 发布于 2018-06-26 23:49:06 阅读数 702 收藏  
展开

## 【题目描述】

Alice 和 Bob 玩了一个古老的游戏：首先画一个  $n \times n$  的点阵（下图  $n = 3$ ）

接着，他们两个轮流在相邻的点之间画上红边和蓝边：



直到围成一个封闭的圈（面积不必为 1）为止，“封圈”的那个人就是赢家。因为棋盘实在是太大了( $n \leq 200$ )，他们的游戏实在是太长了！他们甚至在游戏中都不知道谁赢得了游戏。于是请你写一个程序，帮助他们计算他们是否结束了游戏？

## 【输入】

输入数据第一行为两个整数  $n$  和  $m$ 。 $m$  表示一共画了  $m$  条线。以后  $m$  行，每行首先有两个数字  $(x, y)$ ，代表了画线的起点坐标，接着用空格隔开一个字符，假如字符是 "D"，则是向下连一条边，如果是 "R" 就是向右连一条边。输入数据不会有重复的边且保证正确。

## 【输出】

输出一行：在第几步的时候结束。假如  $m$  步之后也没有结束，则输出一行 "draw"。

## 【输入样例】

```
3 5
1 1 D
1 1 R
1 2 D
2 1 R
2 2 D
```

## 【输出样例】

```
4
```

## 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
```

```

#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1000001
#define MOD 123
#define E 1e-6
using namespace std;
struct Node{
    int x;
    int y;
}father[210][210],a,b;
Node Find(Node temp)
{
    if(father[temp.x][temp.y].x==temp.x&&father[temp.x][temp.y].y==temp.y)
        return temp;
    return father[temp.x][temp.y]=Find(father[temp.x][temp.y]);
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            father[i][j].x=i;
            father[i][j].y=j;
        }

    for(int i=1;i<=m;i++)
    {
        int x,y;
        char ch[10];
        cin>>x>>y>>ch;
        if(ch[0]=='D')
        {
            a=Find(father[x][y]);
            b=Find(father[x+1][y]);
        }
        else if(ch[0]=='R')
        {
            a=Find(father[x][y]);

```

```

        b=Find(father[x][y+1]);
    }
    if(a.x==b.x&& a.y==b.y)
    {
        cout<<i<<endl;
        return 0;
    }
    else
        father[b.x][b.y]=a;
    }
    cout<<"draw"<<endl;
    return 0;
}

```

### 1.1.3.4.4.3 团伙

#### 团伙

原创 Alex\_McAvoy 发布于 2018-06-26 23:47:46 阅读数 898 收藏  
展开

##### 【题目描述】

在某城市里住着  $n$  个人，任何两个认识的人不是朋友就是敌人，而且满足：

- 1、我朋友的朋友是我的朋友；
- 2、我敌人的敌人是我的朋友；

所有是朋友的人组成一个团伙。告诉你关于这  $n$  个人的  $m$  条信息，即某两个人是朋友，或者某两个人是敌人，请你编写一个程序，计算出这个城市最多可能有多少个团伙？

##### 【输入】

第 1 行为  $n$  和  $m$ ， $1 \leq n \leq 1000, 1 \leq m \leq 100\ 000$ ；

以下  $m$  行，每行为  $p\ x\ y$ ， $p$  的值为 0 或 1， $p$  为 0 时，表示  $x$  和  $y$  是朋友， $p$  为 1 时，表示  $x$  和  $y$  是敌人。

##### 【输出】

一个整数，表示这  $n$  个人最多可能有多少个团伙。

##### 【输入样例】

```

6 4
1 1 4
0 3 5

```

0 4 6

1 1 2

【输出样例】

3

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 5001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int vis[N];
int cnt;
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
{
    int xx=Find(x);
    int yy=Find(y);
    if(xx!=yy)
        father[xx]=yy;
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=2*n;i++)
        father[i]=i;
```



```

while(m--)
{
    int p,x,y;
    cin>>p>>x>>y;
    if(p==0)
        Union(x,y);
    else
    {
        Union(x,y+n);
        Union(x+n,y);
    }
}

int cnt=0;
for(int i=1;i<=n;i++)
{
    int temp=Find(i);
    if(!vis[temp])
    {
        vis[temp]=1;
        cnt++;
    }
}
cout<<cnt<<endl;
return 0;
}

```

#### 1.1.3.4.4 打击犯罪

##### 打击犯罪

原创 Alex\_McAvoy 发布于 2018-06-26 23:40:33 阅读数 686 收藏  
展开

##### 【题目描述】

某个地区有  $n(n \leq 1000)$  个犯罪团伙，当地警方按照他们的危险程度由高到低给他们编号为  $1-n$ ，他们有些团伙之间有直接联系，但是任意两个团伙都可以通过直接或间接的方式联系，这样这里就形成了一个庞大的犯罪集团，犯罪集团的危险程度由集团内的犯罪团伙数量唯一确定，而与单个犯罪团伙的危险程度无关（该犯罪集团的危险程度为  $n$ ）。现在当地警方希望花尽量少的时间（即打击掉尽量少的团伙），使得庞大的犯罪集团分离成若干个较小的集团，并且他们中最大的一个的危险程度不超过  $n/2$ 。为达到最好的效果，他们将按顺序打击掉编号  $1$  到  $k$  的犯罪团伙，请编程求出  $k$  的最小值。

**【输入】**

第一行一个正整数  $n$ 。接下来的  $n$  行每行有若干个正整数，第一个整数表示该行除第一个外还有多少个整数，若第  $i$  行存在正整数  $k$ ，表示  $i, k$  两个团伙可以直接联系。

**【输出】**

一个正整数，为  $k$  的最小值。

**【输入样例】**

```
7
2 2 5
3 1 3 4
2 2 4
2 2 3
3 1 6 7
2 5 7
2 5 6
```

**【输出样例】**

```
1
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
struct node{
    int pre;
    int next;
}a[N*N];
int father[N];
int head[N],ans[N];
int cnt;
void add(int x,int y)
{
    cnt++;
```

```

a[cnt].pre=y;
a[cnt].next=head[x];
head[x]=cnt;

cnt++;
a[cnt].pre=x;
a[cnt].next=head[y];
head[y]=cnt;
}
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        father[i]=i;
        ans[i]=1;

        int m;
        cin>>m;
        for(int j=1;j<=m;j++)
        {
            int k;
            cin>>k;
            add(i,k);
        }
    }

    for(int i=n;i>=1;i--)
    {
        int x=Find(i);
        int k=head[i];
        while(k!=0)
        {
            int j=a[k].pre;
            if(j>i)
            {
                int y=Find(j);

```

```

        if(x!=y)
        {
            father[y]=x;
            ans[x]+=ans[y];
            if(ans[x]>n/2)
            {
                cout<<i<<endl;
                return 0;
            }
        }
    }
    k=a[k].next;
}
return 0;
}

```

### 1.1.3.4.4.5 搭配购买

#### 搭配购买

原创 Alex\_McAvoy 发布于 2018-06-29 17:56:40 阅读数 554 收藏  
展开

##### 【题目描述】

Joe 觉得云朵很美，决定去山上的商店买一些云朵。商店里有  $n$  朵云，云朵被编号为  $1, 2, \dots, n$ ，并且每朵云都有一个价值。但是商店老板跟他说，一些云朵要搭配来买才好，所以买一朵云则与这朵云有搭配的云都要买。

但是 Joe 的钱有限，所以他希望买的价值越多越好。

##### 【输入】

第 1 行  $n, m, w$ ，表示  $n$  朵云， $m$  个搭配，Joe 有  $w$  的钱。

第  $2 \sim n+1$  行，每行  $c_i, d_i$  表示  $i$  朵云的价钱和价值。

第  $n+2 \sim n+1+m$  行，每行  $u_i, v_i$ ，表示买  $u_i$  就必须买  $v_i$ ，同理，如果买  $v_i$  就必须买  $u_i$ 。

##### 【输出】

一行，表示可以获得的最大价值。

##### 【输入样例】

```
5 3 10
```

3 10  
 3 10  
 3 10  
 5 100  
 10 1  
 1 3  
 3 2  
 4 2

【输出样例】

1

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int c[N],d[N];
int dp[N];
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
    }
}
```

```

        c[x]+=c[y];
        d[x]+=d[y];
    }
}
int main()
{
    int n,m,w;
    cin>>n>>m>>w;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=n;i++)
        cin>>c[i]>>d[i];
    for(int i=1;i<=m;i++)
    {
        int u,v;
        cin>>u>>v;
        Union(u,v);
    }

    for(int i=1;i<=n;i++)
        Find(i);

    for(int i=1;i<=n;i++)
        if(father[i]==i)
            for(int j=w;j>=c[i];j--)
                dp[j]=max(dp[j],dp[j-c[i]]+d[i]);

    cout<<dp[w]<<endl;
    return 0;
}

```

点赞 1

### 1.1.3.4.4.6 家谱

#### 家谱

原创 Alex\_McAvoy 发布于 2018-06-29 17:55:32 阅读数 433 收藏  
展开

**【题目描述】**

现代的人对于本家族血统越来越感兴趣，现在给出充足的父子关系，请你编写程序找到某个人的最早的祖先。

**【输入】**

由多行组成，首先是一系列有关父子关系的描述，其中每一组父子关系由二行组成，用**#name**的形式描写一组父子关系中的父亲的名字，用**+name**的形式描写一组父子关系中的儿子的名字；接下来用**?name**的形式表示要求该人的最早的祖先；最后用单独的一个**\$**表示文件结束。规定每个人的名字都有且只有**6**个字符，而且首字母大写，且没有任意两个人的名字相同。最多可能有**1000**组父子关系，总人数最多可能达到**50000**人，家谱中的记载不超过**30**代。

**【输出】**

按照输入的要求顺序，求出每一个要找祖先的人的祖先，格式：本人的名字+一个空格+祖先的名字+回车。

**【输入样例】**

```
#George
+Rodney
#Arthur
+Gareth
+Walter
#Gareth
+Edward
?Edward
?Walter
?Rodney
?Arthur
$
```

**【输出样例】**

```
Edward Arthur
Walter Arthur
Rodney George
Arthur Arthur
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
```

```

#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
map <string,string> father;
string Find(string x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int main()
{
    string str;
    string temp1,temp2,temp3;
    while(cin>>str&&str!="$")
    {
        if(str[0]=='#')
        {
            temp1=str.substr(1,str.size()-1);
            if(father[temp1]=="")
                father[temp1]=temp1;
        }
        else if(str[0]=='+')
        {
            temp2=str.substr(1,str.size()-1);
            Find(temp1);
            father[temp2]=father[temp1];
        }
        else if(str[0]=='?')
        {
            temp3=str.substr(1,str.size()-1);
            Find(temp3);
            cout<<temp3<<" "<<father[temp3]<<endl;
        }
    }
    return 0;
}

```



### 1.1.3.4.4.7 亲戚

#### 亲戚

原创 Alex\_McAvoy 发布于 2018-06-29 17:54:15 阅读数 780 收藏  
展开

##### 【题目描述】

若某个家族人员过于庞大，要判断两个是否是亲戚，确实还很难，现在给出某个亲戚关系图，求任意给出的某个人所在家族的人数。

规定：x 和 y 是亲戚，y 和 z 是亲戚，那么 x 和 z 也是亲戚。如果 x,y 是亲戚，那么 x 的亲戚都是 y 的亲戚，y 的亲戚也都是 x 的亲戚。

##### 【输入】

第一行：三个整数 n，(n≤100,000,m≤200,000)，分别表示有 n 个人，m 个信息。

以下 m 行：信息包含两种形式：

**M a b:** 表示 a 和 b 具有亲戚关系。

**Q a:** 要求输出 a 所在家族的人数。

##### 【输出】

要求输出 a 所在家族的人数。

##### 【输入样例】

```
5 10
M 3 2
Q 4
M 1 2
Q 4
M 3 2
Q 1
M 3 1
Q 5
M 4 2
Q 4
```

##### 【输出样例】

```
1
1
3
1
4
```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int cnt[N];
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
        cnt[x]+=cnt[y];
    }
}
int main()
{
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
    {
        father[i]=i;
    }
}

```

```

        cnt[i]=1;
    }
    while(m--)
    {
        char ch[10];
        scanf("%s",ch);
        if(ch[0]=='M')
        {
            int a,b;
            scanf("%d%d",&a,&b);
            Union(a,b);
        }
        if(ch[0]=='Q')
        {
            int a;
            scanf("%d",&a);
            printf("%d\n",cnt[Find(a)]);
        }
    }
    return 0;
}

```

#### 1.1.3.4.4.8 食物链

##### 食物链

原创 Alex\_McAvoy 发布于 2018-06-29 17:52:04 阅读数 422 收藏  
展开

##### 【题目描述】

动物王国中有三类动物 A,B,C，这三类动物的食物链构成了有趣的环形。A 吃 B， B 吃 C， C 吃 A。

现有 N 个动物，以 1—N 编号。每个动物都是 A,B,C 中的一种，但是我们并不知道它到底是哪一种。

有人用两种说法对这 N 个动物所构成的食物链关系进行描述：

第一种说法是"1 X Y"，表示 X 和 Y 是同类。

第二种说法是"2 X Y"，表示 X 吃 Y。

此人对 N 个动物，用上述两种说法，一句接一句地说出 K 句话，这 K 句话有的是真的，有的是假的。当一句话满足下列三条之一时，这句话就是假话，否则就是真话。

- 1) 当前的话与前面的某些真的话冲突，就是假话；
- 2) 当前的话中  $X$  或  $Y$  比  $N$  大，就是假话；
- 3) 当前的话表示  $X$  吃  $X$ ，就是假话。

你的任务是根据给定的  $N$  ( $1 \leq N \leq 50,000$ ) 和  $K$  句话 ( $0 \leq K \leq 100,000$ )，输出假话的总数。

#### 【输入】

第一行是两个整数  $N$  和  $K$ ，以一个空格分隔。

以下  $K$  行每行是三个正整数  $D, X, Y$ ，两数之间用一个空格隔开，其中  $D$  表示说法的种类。

若  $D=1$ ，则表示  $X$  和  $Y$  是同类。

若  $D=2$ ，则表示  $X$  吃  $Y$ 。

#### 【输出】

只有一个整数，表示假话的数目。

#### 【输入样例】

```
100 7
1 101 1
2 1 2
2 2 3
2 3 3
1 1 3
2 3 1
1 5 5
```

#### 【输出样例】

```
3
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
```

```

#define PI acos(-1.0)
#define N 1000001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
void Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
        father[y]=x;
}
int main()
{
    int n,k;
    cin>>n>>k;
    for(int i=1;i<=3*n;i++)
        father[i]=i;
    int sum=0;
    for(int i=1;i<=k;i++)
    {
        int t,x,y;
        cin>>t>>x>>y;

        if(x>n||y>n)
            sum++;
        else if(t==1)
        {
            if(Find(x+n)==Find(y)||Find(x+2*n)==Find(y))//x 吃 y、y 吃 x、非同类
                sum++;
            else
            {
                Union(x,y);
                Union(x+n,y+n);
                Union(x+2*n,y+2*n);
            }
        }
    }
}

```

```

else if(t==2)
{
    if(Find(x)==Find(y)||Find(x+2*n)==Find(y))//同类
        sum++;
    else
    {
        Union(x+n,y);
        Union(x+2*n,y+n);
        Union(x,y+2*n);
    }
}
}
cout<<sum<<endl;
return 0;
}

```

### 1.1.3.4.5 第五节 最小生成树

#### 1.1.3.4.5.1 T1348 城市公交网建设问题

##### T1348 城市公交网建设问题

原创 Alex\_McAvoy 发布于 2018-06-30 00:20:18 阅读数 1392 收藏  
展开

##### 【题目描述】

有一张城市地图，图中的顶点为城市，无向边代表两个城市间的连通关系，边上的权为在这两个城市之间修建高速公路的造价，研究后发现，这个地图有一个特点，即任一对城市都是连通的。现在的问题是，要修建若干高速公路把所有城市联系起来，问如何设计可使得工程的总造价最少？

##### 【输入】

$n$ （城市数， $1 \leq n \leq 100$ ）

$e$ （边数）

以下  $e$  行，每行 3 个数  $i, j, w_{ij}$ ，表示在城市  $i, j$  之间修建高速公路的造价。

##### 【输出】

$n-1$  行，每行为两个城市的序号，表明这两个城市间建一条高速公路。

## 【输入样例】

```

?5 8
1 2 2
2 5 9
5 4 7
4 1 10
1 3 12
4 3 6
5 3 3
2 3 8

```

## 【输出样例】

```

1 2
2 3
3 4
3 5

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
struct Node{
    int u;
    int v;
    int w;
}g[N*N],dis[N];
void quick_sort(int left,int right)
{
    int i=left,j=right;

```

```

int mid=g[(left+right)/2].w;
while(i<=j)
{
    while(g[i].w<mid)
        i++;
    while(g[j].w>mid)
        j--;
    if(i<=j)
    {
        swap(g[i],g[j]);
        i++;
        j--;
    }
}
if(i<right)
    quick_sort(i,right);
if(left<j)
    quick_sort(left,j);
}
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
        return 1;
    }
    return 0;
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=m;i++)
    {

```



```

    int u,v,w;
    cin>>u>>v>>w;
    if(u>v)//小点在前
        swap(u,v);
    g[i].u=u;
    g[i].v=v;
    g[i].w=w;
}

int sum=0;
quick_sort(1,m);
for(int i=1;i<=m;i++)
    sum+=g[i].w;

int cnt=0;
for(int i=1;i<=m;i++)
    if(Union(g[i].u,g[i].v))
    {
        cnt++;
        dis[cnt].u=g[i].u;
        dis[cnt].v=g[i].v;
        dis[cnt].w=g[i].w;
        if(cnt==n-1)
            break;
    }
for(int i=1;i<=cnt;i++)
    for(int j=i+1;j<=cnt;j++)
    {
        if(dis[i].u>dis[j].u)
            swap(dis[i],dis[j]);
        else if(dis[i].u==dis[j].u&&dis[i].v>dis[j].v)
            swap(dis[i],dis[j]);
    }

for(int i=1;i<=cnt;i++)
    cout<<dis[i].u<<" "<<dis[i].v<<endl;
return 0;
}

```

### 1.1.3.4.5.2 T1349 最优布线问题

#### T1349 最优布线问题

原创 Alex\_McAvoy 发布于 2018-06-30 00:20:22 阅读数 463 收藏  
展开

### 【题目描述】

学校有  $n$  台计算机，为了方便数据传输，现要将它们用数据线连接起来。两台计算机被连接是指它们有数据线连接。由于计算机所处的位置不同，因此不同的两台计算机的连接费用往往是不同的。

当然，如果将任意两台计算机都用数据线连接，费用将是相当庞大的。为了节省费用，我们采用数据的间接传输手段，即一台计算机可以间接的通过若干台计算机（作为中转）来实现与另一台计算机的连接。

现在由你负责连接这些计算机，任务是使任意两台计算机都连通（不管是直接的或间接的）。

### 【输入】

第一行为整数  $n$  ( $2 \leq n \leq 100$ )，表示计算机的数目。此后的  $n$  行，每行  $n$  个整数。第  $x+1$  行  $y$  列的整数表示直接连接第  $x$  台计算机和第  $y$  台计算机的费用。

### 【输出】

一个整数，表示最小的连接费用。

### 【输入样例】

```
?3
0 1 2
1 0 1
2 1 0
```

### 【输出样例】

```
2
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
```

```

#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N];
int dis[N],vis[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            cin>>g[i][j];

    memset(vis,0,sizeof(vis));
    for(int i=1;i<=n;i++)
        dis[i]=g[1][i];
    for(int i=1;i<=n;i++)
    {
        int k;
        int minn=INF;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]<minn)
            {
                minn=dis[j];
                k=j;
            }

        vis[k]=1;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]>g[k][j])
                dis[j]=g[k][j];
    }

    int sum=0;
    for(int i=1;i<=n;i++)
        sum+=dis[i];

    cout<<sum<<endl;
    return 0;
}

```

### 1.1.3.4.5.3 T1350 最短网络

#### T1350 最短网络

原创 Alex\_McAvoy 发布于 2018-06-29 22:02:28 阅读数 474 收藏  
展开

##### 【题目描述】

农民约翰被选为他们镇的镇长！他其中一个竞选承诺就是在镇上建立起互联网，并连接到所有的农场。当然，他需要你的帮助。约翰已经给他的农场安排了一条高速的网络线路，他想把这条线路共享给其他农场。为了用最小的消费，他想铺设最短的光纤去连接所有的农场。你将得到一份各农场之间连接费用的列表，你必须找出能连接所有农场并所用光纤最短的方案。每两个农场间的距离不会超过 100000。

##### 【输入】

第一行：农场的个数， $N$  ( $3 \leq N \leq 100$ )。

第二行..结尾:后来的行包含了一个  $N \times N$  的矩阵,表示每个农场之间的距离。理论上，他们是  $N$  行，每行由  $N$  个用空格分隔的数组成，实际上，他们限制在 80 个字符，因此，某些行会紧接着另一些行。当然，对角线将会是 0，因为不会有线路从第  $i$  个农场到它本身。

##### 【输出】

只有一个输出，其中包含连接到每个农场的光纤的最小长度。

##### 【输入样例】

```
?4
0 4 9 21
4 0 8 17
9 8 0 16
21 17 16 0
```

##### 【输出样例】

```
?28
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
```

```

#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N];
int dis[N],vis[N];
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            cin>>g[i][j];

    memset(vis,0,sizeof(vis));
    for(int i=1;i<=n;i++)
        dis[i]=g[1][i];
    for(int i=1;i<=n;i++)
    {
        int k;
        int minn=INF;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]<minn)
            {
                minn=dis[j];
                k=j;
            }

        vis[k]=1;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]>g[k][j])
                dis[j]=g[k][j];
    }

    int sum=0;
    for(int i=1;i<=n;i++)
        sum+=dis[i];

```

```

    cout<<sum<<endl;
    return 0;
}

```

点赞

### 1.1.3.4.5.4 T1351 家谱树

#### T1351 家谱树

原创 Alex\_McAvoy 发布于 2018-06-29 21:59:47 阅读数 917 收藏  
展开

##### 【题目描述】

有个人的家族很大，辈分关系很混乱，请你帮整理一下这种关系。

给出每个人的孩子的信息。

输出一个序列，使得每个人的后辈都比那个人后列出。

##### 【输入】

第 1 行一个整数  $N$  ( $1 \leq N \leq 100$ )，表示家族的人数；

接下来  $N$  行，第  $i$  行描述第  $i$  个人的儿子；

每行最后是 0 表示描述完毕。

##### 【输出】

输出一个序列，使得每个人的后辈都比那个人后列出；

如果有多解输出任意一解。

##### 【输入样例】

```

5
0
4 5 1 0
1 0
5 3 0
3 0

```

##### 【输出样例】

```

2 4 5 3 1

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
struct Node{
    int pre;
    int next;
}g[N*N];
int head[N],vis[N],q[N],dis[N];
int cnt;
void add(int x,int y)
{
    cnt++;
    g[cnt].pre=y;
    g[cnt].next=head[x];
    head[x]=cnt;
}
void top_sort(int k)
{
    vis[k]=1;

    int headd=1,tail=1;
    q[tail]=k;
    tail++;

    while(headd<tail)
    {
        int x=q[headd];
        cout<<x<<" ";
    }
}

```

```

        for(int b=head[x];b;b=g[b].next)
        {
            int y=g[b].pre;
            dis[y]--;
            if(!dis[y])
            {
                vis[y]=1;
                q[tail]=y;
                tail++;
            }
        }
        headd++;
    }
}
int main()
{
    int n;
    cin>>n;

    int j;
    for(int i=1;i<=n;i++)
        while(cin>>j&&j)
        {
            add(i,j);
            dis[j]++;
        }

    for(int i=1;i<=n;i++)
        if(!vis[i]&&!dis[i])
            top_sort(i);

    return 0;
}

```

### 1.1.3.4.5.5 T1391 局域网

#### T1391 局域网

原创 Alex\_McAvoy 发布于 2018-06-30 00:25:47 阅读数 518 收藏  
展开

##### 【题目描述】

某个局域网内有  $n(n \leq 100)$  台计算机，由于搭建局域网时工作人员的疏忽，现在局域网



内的连接形成了回路，我们知道如果局域网形成回路那么数据将不停的在回路内传输，造成网络卡的现象。因为连接计算机的网线本身不同，所以有一些连线不是很畅通，我们用  $f(i,j)$  表示  $i,j$  之间连接的畅通程度 ( $f(i,j) \leq 1000$ )， $f(i,j)$  值越小表示  $i,j$  之间连接越通畅， $f(i,j)$  为 0 表示  $i,j$  之间无网线连接。现在我们需要解决回路问题，我们将除去一些连线，使得网络中没有回路，并且被除去网线的  $\sum f(i,j)$  最大，请求出这个最大值。

#### 【输入】

第一行两个正整数  $n$   $k$

接下来的  $k$  行每行三个正整数  $i$   $j$   $m$  表示  $i,j$  两台计算机之间有网线联通，通畅程度为  $m$ 。

#### 【输出】

一个正整数， $\sum f(i,j)$  的最大值。

#### 【输入样例】

```
5 5
1 2 8
1 3 1
1 5 3
2 4 5
3 4 2
```

#### 【输出样例】

```
8
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
```

```

struct Node{
    int a;
    int b;
    int w;
}g[N*N];
void quick_sort(int left,int right)
{
    int i=left,j=right;
    int mid=g[(left+right)/2].w;
    while(i<=j)
    {
        while(g[i].w<mid)
            i++;
        while(g[j].w>mid)
            j--;
        if(i<=j)
        {
            swap(g[i],g[j]);
            i++;
            j--;
        }
    }
    if(i<right)
        quick_sort(i,right);
    if(left<j)
        quick_sort(left,j);
}
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
        return 1;
    }
    return 0;
}

```

```

int main()
{
    int n,k;
    cin>>n>>k;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=k;i++)
        cin>>g[i].a>>g[i].b>>g[i].w;

    int sum=0;
    quick_sort(1,k);
    for(int i=1;i<=k;i++)
        sum+=g[i].w;

    int cnt=0,ans=0;
    for(int i=1;i<=k;i++)
        if(Union(g[i].a,g[i].b))
        {
            ans+=g[i].w;
            cnt++;
            if(cnt==n-1)
                break;
        }

    cout<<sum-ans<<endl;
    return 0;
}

```

### 1.1.3.4.5.6 T1392 繁忙的都市

#### T1392 繁忙的都市

原创 Alex\_McAvoy 发布于 2018-06-30 00:20:36 阅读数 472 收藏  
展开

##### 【题目描述】

城市 C 是一个非常繁忙的大都市，城市中的道路十分的拥挤，于是市长决定对其中的道路进行改造。城市 C 的道路是这样分布的：城市中有  $n$  个交叉路口，有些交叉路口之间有道路相连，两个交叉路口之间最多有一条道路相连接。这些道路是双向的，且把所有的交叉路口直接或间接的连接起来了。每条道路都有一个分值，分值越小表示这个道路越繁忙，越需要进行改造。但是市政府的资金有限，市长希望进行改造的道路越少越好，于是他提出下面的要求：

1. 改造的那些道路能够把所有的交叉路口直接或间接的连通起来。
2. 在满足要求 1 的情况下，改造的道路尽量少。
3. 在满足要求 1、2 的情况下，改造的那些道路中分值最大值尽量小。

作为市规划局的你，应当作出最佳的决策，选择那些道路应当被修建。

#### 【输入】

第一行有两个整数  $n, m$  表示城市有  $n$  个交叉路口， $m$  条道路。接下来  $m$  行是对每条道路的描述， $u, v, c$  表示交叉路口  $u$  和  $v$  之间有道路相连，分值为  $c$ 。( $1 \leq n \leq 300$ ,  $1 \leq c \leq 10000$ )。

#### 【输出】

两个整数  $s, \max$ ，表示你选出了几条道路，分值最大的那条道路的分值是多少。

#### 【输入样例】

```
4 5
1 2 3
1 4 5
2 4 7
2 3 6
3 4 8
```

#### 【输出样例】

```
3 6
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1001
#define MOD 123
#define E 1e-6
using namespace std;
```

```

int g[N][N];
int dis[N],vis[N];
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            if(i==j)
                g[i][j]=0;
            else
                g[i][j]=INF;
        }

    for(int i=1;i<=m;i++)
    {
        int x,y,w;
        cin>>x>>y>>w;
        g[x][y]=w;
        g[y][x]=w;
    }

    memset(vis,0,sizeof(vis));
    for(int i=1;i<=n;i++)
        dis[i]=g[1][i];
    for(int i=1;i<=n;i++)
    {
        int k;
        int minn=INF;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]<minn)
            {
                minn=dis[j];
                k=j;
            }

        vis[k]=1;
        for(int j=1;j<=n;j++)
            if(!vis[j]&&dis[j]>g[k][j])
                dis[j]=g[k][j];
    }

    int maxx=-INF;

```

```

for(int i=1;i<=n;i++)
    maxx=max(maxx,dis[i]);

cout<<n-1<<" "<<maxx<<endl;
return 0;
}

```

点赞

### 1.1.3.4.5.7 T1393 联络员

#### T1393 联络员

原创 Alex\_McAvoy 发布于 2018-06-30 00:24:39 阅读数 600 收藏  
展开

##### 【题目描述】

Tyvj 已经一岁了，网站也由最初的几个用户增加到了上万个用户，随着 Tyvj 网站的逐步壮大，管理员的数目也越来越多，现在你身为 Tyvj 管理层的联络员，希望你找到一些通信渠道，使得管理员两两都可以联络（直接或者是间接都可以）。Tyvj 是一个公益性的网站，没有过多的利润，所以你要尽可能的使费用少才可以。

目前你已经知道，Tyvj 的通信渠道分为两大类，一类是必选通信渠道，无论价格多少，你都需要把所有的都选择上；还有一类是选择性的通信渠道，你可以从中挑选一些作为最终管理员联络的通信渠道。数据保证给出的通行渠道可以让所有的管理员联通。

##### 【输入】

第一行  $n, m$  表示 Tyvj 一共有  $n$  个管理员，有  $m$  个通信渠道；

第二行到  $m+1$  行，每行四个非负整数， $p, u, v, w$  当  $p=1$  时，表示这个通信渠道为必选通信渠道；当  $p=2$  时，表示这个通信渠道为选择性通信渠道； $u, v, w$  表示本条信息描述的是  $u, v$  管理员之间的通信渠道， $u$  可以收到  $v$  的信息， $v$  也可以收到  $u$  的信息， $w$  表示费用。

##### 【输出】

最小的通信费用。

##### 【输入样例】

```

5 6
1 1 2 1
1 2 3 1
1 3 4 1
1 4 1 1

```

2 2 5 10  
2 2 5 5

【输出样例】

9

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 10001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
struct Node{
    int p;
    int u;
    int v;
    int w;
}g[N*N];
int cmp(Node a,Node b)
{
    if(a.p==2&&b.p==2)
        return a.w<b.w;
    return a.p<b.p;
}
int Find(int x)
{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int Union(int x,int y)
```

```

{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
        return 1;
    }
    return 0;
}
int main()
{
    int n,m;
    cin>>n>>m;
    for(int i=1;i<=n;i++)
        father[i]=i;
    for(int i=1;i<=m;i++)
        cin>>g[i].p>>g[i].u>>g[i].v>>g[i].w;

    sort(g+1,g+1+m,cmp);
    int sum=0;
    for(int i=1;i<=m;i++)
    {
        if(g[i].p==1)
        {
            sum+=g[i].w;
            Union(g[i].u,g[i].v);
        }
        else if(g[i].p==2)
        {
            if(Union(g[i].u,g[i].v))
                sum+=g[i].w;
        }
    }
    cout<<sum<<endl;
    return 0;
}

```

点赞

### 1.1.3.4.5.8 T1394 连接格点

#### T1394 连接格点



原创 Alex\_McAvoy 发布于 2018-06-30 00:23:01 阅读数 475 收藏  
展开

### 【题目描述】

有一个  $M$  行  $N$  列的点阵，相邻两点可以相连。一条纵向的连线花费一个单位，一条横向的连线花费两个单位。某些点之间已经有连线了，试问至少还需要花费多少个单位才能使所有的点全部连通。

### 【输入】

第一行输入两个正整数  $m$  和  $n$ 。

以下若干行每行四个正整数  $x1,y1,x2,y2$ ，表示第  $x1$  行第  $y1$  列的点和第  $x2$  行第  $y2$  列的点已经有连线。输入保证  $|x1-x2|+|y1-y2|=1$ 。

### 【输出】

输出使得连通所有点还需要的最小花费。

### 【输入样例】

```
2 2
1 1 2 1
```

### 【输出样例】

```
3
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 1000001
#define MOD 123
#define E 1e-6
using namespace std;
int father[N];
int Find(int x)
```

```

{
    if(father[x]==x)
        return x;
    return father[x]=Find(father[x]);
}
int Union(int x,int y)
{
    x=Find(x);
    y=Find(y);
    if(x!=y)
    {
        father[y]=x;
        return 1;
    }
    return 0;
}
int main()
{
    int m,n;
    cin>>m>>n;

    for(int i=1;i<=m*n;i++)
        father[i]=i;

    int x1,x2,y1,y2;
    while(scanf("%d%d%d%d",&x1,&y1,&x2,&y2)==4)
        Union((x1-1)*n+y1,(x2-1)*n+y2);

    int sum=0;
    for(int j=1;j<=n;j++)//连竖线需要一个单位
        for(int i=1;i<=m;i++)
            if(Union((i-1)*n+j,(i)*n+j))
                sum+=1;
    for(int i=1;i<=m;i++)//连横线需要两个单位
        for(int j=1;j<=n;j++)
            if(Union((i-1)*n+j,(i-1)*n+j+1))
                sum+=2;

    printf("%d\n",sum);
    return 0;
}

```

## 1.1.3.4.6 第六节 拓扑排序与关键路径

### 1.1.3.4.6.1 T1352 奖金

#### T1352 奖金

原创 Alex\_McAvoy 发布于 2018-06-20 19:38:13 阅读数 923 收藏  
展开

##### 【题目描述】

由于无敌的凡凡在 2005 年世界英俊帅气男总决选中胜出,Yali Company 总经理 Mr.Z 心情好,决定给每位员工发奖金。公司决定以每个人本年在公司的贡献为标准来计算他们得到奖金的多少。

于是 Mr.Z 下令召开  $m$  方会谈。每位参加会谈的代表提出了自己的意见:“我认为员工  $a$  的奖金应该比  $b$  高!”Mr.Z 决定要找出一种奖金方案,满足各位代表的意见,且同时使得总奖金数最少。每位员工奖金最少为 100 元。

##### 【输入】

第一行两个整数  $n,m$ ,表示员工总数和代表数;

以下  $m$  行,每行 2 个整数  $a,b$ ,表示某个代表认为第  $a$  号员工奖金应该比第  $b$  号员工高。

##### 【输出】

若无法找到合理方案,则输出“Poor Xed”;否则输出一个数表示最少总奖金。

##### 【输入样例】

```
2 1
1 2
```

##### 【输出样例】

```
201
```

##### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
```

```

#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
int n,m;
int head[N],side[N];
int f[N],q[N];
int vis[N];
int cnt;
struct node{
    int pre;
    int next;
}a[N];
void add_edge(int x,int y)
{
    cnt++;
    a[cnt].pre=y;
    a[cnt].next=head[x];
    head[x]=cnt;
}
void topsort(int x)
{
    int headd=1,tail=1;
    vis[x]=1;
    q[tail]=x;
    tail++;
    while(headd<tail)
    {
        int u=q[headd];
        for(int b=head[u];b;b=a[b].next)
        {
            int v=a[b].pre;
            side[v]--;
            f[v]=max(f[v],f[u]+1);
            if(side[v]==0)
            {
                q[tail]=v;
                tail++;
                vis[v]=1;
            }
        }
    }
}

```

```

    }
    }
    headd++;
}
}
int main()
{
    int sum=0;
    cin>>n>>m;

    for(int i=1;i<=n;i++)
        f[i]=100;

    for(int i=1;i<=m;i++)
    {
        int x,y;
        cin>>y>>x;
        add_edge(x,y);
        side[y]++;
    }

    for(int i=1;i<=n;i++)
        if(!vis[i]&&!side[i])
            topsort(i);

    for(int i=1;i<=n;i++)
        if(!vis[i])
        {
            cout<<"Poor Xed"<<endl;
            return 0;
        }

    for(int i=1;i<=n;i++)
        sum+=f[i];

    cout<<sum<<endl;
    return 0;
}

```

点赞

收藏

分享

Alex\_McAvoy

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

## 1.1.3.4.6.2 T1395 烦人的幻灯片

### T1395 烦人的幻灯片

原创 Alex\_McAvoy 发布于 2018-06-20 19:20:40 阅读数 744 收藏  
展开

#### 【题目描述】

李教授将于今天下午作一次非常重要的演讲。不幸的事他不是一个非常爱整洁的人，他把自己演讲要用的幻灯片随便堆在了一起。因此，演讲之前他不得不去整理这些幻灯片。作为一个讲求效率的学者，他希望尽可能简单地完成它。教授这次演讲一共要用  $n$  张幻灯片 ( $n \leq 26$ )，这  $n$  张幻灯片按照演讲要使用的顺序已经用数字  $1 \sim n$  编了号。因为幻灯片是透明的，所以我们不能一下子看清每一个数字所对应的幻灯片。

现在我们用大写字母 A,B,C.....再次把幻灯片依次编号。你的任务是编写一个程序，把幻灯片的数字编号和字母编号对应起来，显然这种对应应该是唯一的；若出现多种对应的情况或是某些数字编号和字母编号对应不起来，我们称对应是无法实现的。

#### 【输入】

第一行只有一个整数  $n$ ，表示有  $n$  张幻灯片，接下来的  $n$  行每行包括 4 个整数  $xmin, xmax, ymin, ymax$ （整数之间用空格分开）为幻灯片的坐标，这  $n$  张幻灯片按其在文件中出现的顺序从前到后依次编号为 A,B,C.....，再接下来的  $n$  行依次为  $n$  个数字编号的坐标  $x,y$ ，显然在幻灯片之外是不会有数字的。

#### 【输出】

若是对应可以实现，输出文件应该包括  $n$  行，每一行为一个字母和一个数字，中间以一个空格隔开，并且每行以字母的升序排列，注意输出的字母要大写并且定格；反之，若是对应无法实现，在文件的第一行顶格输出 **None** 即可。首行末无多余的空格。

#### 【输入样例】

```
4
6 22 10 20
4 18 6 16
8 20 2 18
10 24 4 8
9 15
19 17
11 7
21 11
```

## 【输出样例】

A 4  
B 1  
C 2  
D 3

## 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 101
#define MOD 123
#define E 1e-6
using namespace std;
int g[N][N],side[N],sum[N];
int main()
{
    int n;
    int x1[27],x2[27],y1[27],y2[27];
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>x1[i]>>x2[i]>>y1[i]>>y2[i];

    for(int i=1;i<=n;i++)
    {
        int x,y;
        cin>>x>>y;
        for(int j=1;j<=n;j++)
            if(x1[j]<=x && x<=x2[j] && y1[j]<=y && y<=y2[j])
            {
                g[i][j]=1;
                side[j]++;
            }
    }

    int num=0;
```

```

int temp;
for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++)
        if(side[j]==1)
        {
            num++;
            side[j]--;
            for(int k=1;k<=n;k++)
                if(g[k][j])
                {
                    g[k][j]=0;
                    temp=k;
                    sum[j]=k;
                    break;
                }
            for(int k=1;k<=n;k++)
                if(g[temp][k])
                {
                    g[temp][k]=0;
                    side[k]--;
                }
        }

if(num==n)
    for(int i=1;i<=n;i++)
        cout<<(char)(i-1+'A')<<" "<<sum[i]<<endl;
else
    cout<<"None"<<endl;

return 0;
}

```

### 1.1.3.4.6.3 T1396 病毒

#### T1396 病毒

原创 Alex\_McAvoy 发布于 2018-06-20 19:09:53 阅读数 2247 收藏  
展开

##### 【题目描述】

有一天，小 y 突然发现自己的计算机感染了一种病毒！还好，小 y 发现这种病毒很弱，只是会把文档中的所有字母替换成其它字母，但并不改变顺序，也不会增加和删除字母。



现在怎么恢复原来的文档呢！小 y 很聪明，他在其他没有感染病毒的机器上，生成了一个由若干单词构成的字典，字典中的单词是按照字母顺序排列的，他把这个文件拷贝到自己的机器里，故意让它感染上病毒，他想利用这个字典文件原来的有序性，找到病毒替换字母的规律，再用来恢复其它文档。

现在你的任务是：告诉你被病毒感染了字典，要你恢复一个字母串。

### 【输入】

第一行为整数  $K$  ( $\leq 50000$ )，表示字典中的单词个数。

以下  $K$  行，是被病毒感染了字典，每行一个单词。

最后一行是需要你恢复的一串字母。

所有字母均为小写。

### 【输出】

输出仅一行，为恢复后的一串字母。当然也有可能出现字典不完整、甚至字典是错的情况，这时请输出一个 0。

### 【输入样例】

```
6
cebdbac
cac
ecd
dca
aba
bac
cedab
```

### 【输出样例】

```
abcde
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<vector>
#include<set>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 50001
```

```

#define MOD 123
#define E 1e-6
using namespace std;
int n;
int a[N][101],g[27][27];
int enter[27],vis[27],sum[27];
int cnt;
int p;
void judge(int x,int y)
{
    if(a[x][y]==0||a[x+1][y]==0)
        return ;
    if(a[x][y]!=a[x+1][y])
    {
        g[a[x][y]][a[x+1][y]]=1;
        enter[a[x+1][y]]++;
    }
    else
        judge(x,y+1);
}
bool topsort()
{
    for(int i=1;i<=27;++i)
        if(vis[i])
            cnt++;
    for(int i=1;i<n;++i)
        judge(i,1);

    int work,temp;
    while(p!=cnt)
    {
        work=0;
        for(int i=1;i<=cnt;++i)
        {
            if(enter[i]==0)
            {
                temp=i;
                enter[i]=-1;
                work++;
            }
        }
        if(work!=1)
            return false;
        sum[++p]=temp;
    }
}

```

```

        for(int i=1;i<=cnt;++i)
        {
            if(g[temp][i])
            {
                g[temp][i]=0;
                enter[i]--;
            }
        }
    }
    return true;
}
int main()
{
    cin>>n;
    for(int i=1;i<=(n+1);i++)
    {
        string str;
        cin>>str;
        for(int j=0;j<str.size();j++)
        {
            a[i][++a[i][0]]=int(str[j]+1-'a');
            vis[str[j]+1-'a']=1;
        }
    }

    if(!topsort())
        cout<<0;
    else
    {
        for(int i=1;i<=a[n+1][0];++i)
        {
            for(int j=1;j<=cnt;++j)
            {
                if(a[n+1][i]==sum[j])
                {
                    cout<<char(j-1+'a');
                    break;
                }
            }
        }
    }
    return 0;
}

```

点赞 2

收藏

分享

Alex\_McAvoy

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

### 1.1.1.3.4.7 新建项目

## 1.2 算法提高篇

### 1.2.1 第一部分：基础算法

#### 1.2.1.1 第一章：贪心算法

##### 1.2.1.1.1 活动安排（信息学奥赛一本通-T1422）

##### 活动安排（信息学奥赛一本通-T1422）

原创 Alex\_McAvoy 发布于 2019-07-20 10:52:21 阅读数 185 收藏  
展开

##### 【题目描述】

设有  $n$  个活动的集合  $E=\{1,2,\dots,n\}$ ，其中每个活动都要求使用同一资源，如演讲会场等，而在同一时间内只有一个活动能使用这一资源。每个活动  $i$  都有一个要求使用该资源的起始时间  $s_i$  和一个结束时间  $f_i$ ，且  $s_i < f_i$ 。如果选择了活动  $i$ ，则它在半开时间区间  $[s_i, f_i)$  内占用资源。若区间  $[s_i, f_i)$  与区间  $[s_j, f_j)$  不相交，则称活动  $i$  与活动  $j$  是相容的。

也就是说，当  $s_i \geq f_j$  或  $s_j \geq f_i$  时，活动  $i$  与活动  $j$  相容。选择出由相互兼容的活动组成的最大集合。

#### 【输入】

第 1 行一个整数  $n(n \leq 1000)$ ，接下来  $n$  行，每行两个整数  $s_i$  和  $f_i$ 。

#### 【输出】

输出尽可能多的互相兼容的活动个数。

#### 【输入样例】

```
4
1 3
4 6
2 5
1 7
```

#### 【输出样例】

```
2
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    int st,ed;
```

```

    bool operator < (const Node &rhs)const{
        return ed<rhs.ed;
    }
}a[N];
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d%d",&a[i].st,&a[i].ed);
        sort(a+1,a+1+n);
        int res=1;
        int time=a[1].ed;
        for(int i=2;i<=n;i++){
            if(a[i].st>=time){
                res++;
                time=a[i].ed;
            }
        }
        printf("%d\n",res);
        return 0;
    }
}

```

点赞

### 1.2.1.1.2 种树（信息学奥赛一本通-T1423）

#### 种树（信息学奥赛一本通-T1423）

原创 Alex\_McAvoy 发布于 2019-07-20 11:02:17 阅读数 115 收藏  
展开

##### 【题目描述】

现在我们国家开展新农村建设，农村的住房建设纳入了统一规划，统一建设，政府要求每一住户门口种些树。门口路边的地区被分割成块，并被编号成  $1..N$ 。每个部分为一个单位尺寸大小并最多可种一棵树。每个居民房子门前被指定了三个号码  $B, E, T$ 。这三个数表示该居民想在  $B$  和  $E$  之间最少种  $T$  棵树。当然， $B \leq E$ ，居民必须记住在指定区不能种多于区域地块数的树，所以  $T \leq E - B + 1$ 。居民们想种树的各自区域可以交叉。你的任务是求出能满足所有要求的最少的树的数量，尽量较少政府的支出。

##### 【输入】

第一行包含数据  $N, M$ ，区域的个数( $0 < N \leq 30000$ )，房子的数目( $0 < m \leq 5000$ );

下面的  $m$  行描述居民们的需要:  $B \ E \ T$ ,  $0 < B \leq E \leq 30000$ ,  $T \leq E - B + 1$ 。

**【输出】**

输出一个数，为满足所有居民的要求，所需要种树的最少数量。

**【输入样例】**

```
9 4
3 5 2
1 4 2
4 6 2
8 9 2
```

**【输出样例】**

```
5
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 30000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    int st,ed;
    int num;
    bool operator < (const Node &rhs)const{
        return ed<rhs.ed;
    }
}a[N];
```

```

bool vis[N];
int main(){
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=m;i++)
        scanf("%d%d%d",&a[i].st,&a[i].ed,&a[i].num);
    sort(a+1,a+1+m);

    int res=0;
    for(int i=1;i<=m;i++){
        int used=0;
        for(int j=a[i].st;j<=a[i].ed;j++)//从前向后统计已有多少树
            if(vis[j])
                used++;

        if(used<a[i].num){//若树的个数不到当前区间所需个数
            for(int j=a[i].ed;j>=a[i].st;j--){//从后向前在当前区间种树
                if(!vis[j]){
                    vis[j]=true;
                    used++;
                    res++;
                    if(used==a[i].num)
                        break;
                }
            }
        }
    }

    printf("%d\n",res);

    return 0;
}

```

### 1.2.1.1.3 喷水装置（信息学奥赛一本通-T1424）

#### 喷水装置（信息学奥赛一本通-T1424）

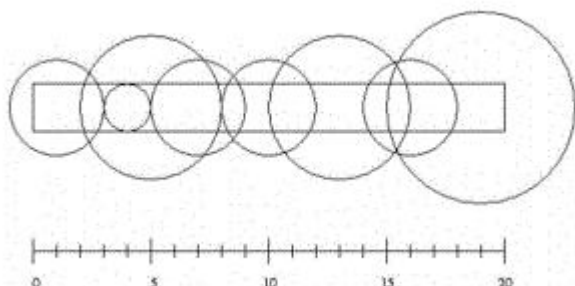
原创 Alex\_McAvoy 发布于 2019-07-21 20:32:44 阅读数 82 收藏  
展开

##### 【题目描述】

长  $L$  米，宽  $WW$  米的草坪里装有  $n$  个浇灌喷头。每个喷头都装在草坪中心线上（离



两边各  $W/2$  米)。我们知道每个喷头的位置（离草坪中心线左端的距离），以及它能覆盖到的浇灌范围。



请问：如果要同时浇灌整块草坪，最少需要打开多少个喷头？

### 【输入】

输入包含若干组测试数据。

第一行一个整数  $T$  表示数据组数；

每组数据的第一行是整数  $n$ 、 $L$  和  $W$ ；

接下来的  $n$  行，每行包含两个整数，给出一个喷头的位置和浇灌半径（上面的示意图是样例输入第一组数据所描述的情况）。

### 【输出】

对每组测试数据输出一个数字，表示要浇灌整块草坪所需喷头数目的最小值。如果所有喷头都打开也不能浇灌整块草坪，则输出  $-1$

### 【输入样例】

```
3
8 20 2
5 3
4 1
1 2
7 2
10 2
13 3
16 2
19 4
3 10 1
3 5
9 3
6 1
3 10 1
5 3
```

1 1  
9 1

【输出样例】

6  
2  
-1

【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 100000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    double pos;
    double num;
    bool operator < (const Node &rhs)const{
        return pos<rhs.pos;
    }
}node[N];
int main(){
    int t;
    scanf("%d",&t);
    while(t--){
        int n,l,w;
```

```

scanf("%d%d%d",&n,&l,&w);

int cnt=0;
for(int i=1;i<=n;i++){
    int pos,r;
    scanf("%d%d",&pos,&r);
    if(r>w/2){
        node[++cnt].pos=pos-sqrt(r*r-(w/2.0)*(w/2.0));
        node[cnt].num=pos+sqrt(r*r-(w/2.0)*(w/2.0));
    }
}
sort(node+1,node+1+cnt);

double len=0;
int res=0;
bool flag=true;
int i=1;
while(len<l){
    res++;
    double pos=len;
    while(node[i].pos<=pos&&i<=cnt){
        if(len<node[i].num)
            len=node[i].num;
        i++;
    }
    if(len==pos&&pos<l){
        printf("-1\n");
        flag=false;
        break;
    }
}
if(flag)
    printf("%d\n",res);
}
return 0;
}

```

点赞

收藏

分享

**Alex\_McAvoy**

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

展开阅读全文

## 1.2.1.1.4 加工生产调度（信息学奥赛一本通-T1425）

### 加工生产调度（信息学奥赛一本通-T1425）

原创 Alex\_McAvoy 发布于 2019-07-21 21:00:08 阅读数 96 收藏  
展开

#### 【题目描述】

某工厂收到了  $n$  个产品的订单，这  $n$  个产品分别在 A、B 两个车间加工，并且必须先 A 车间加工后才可以到 B 车间加工。

某个产品  $i$  在 A、B 两车间加工的时间分别为  $A_i, B_i$ 。怎样安排这  $n$  个产品的加工顺序，才能使总的加工时间最短。

这里所说的加工时间是指：从开始加工第一个产品到最后所有的产品都已在 A、B 两车间加工完毕的时间。

#### 【输入】

第一行仅一个数据  $n$ ，表示产品的数量；

接下来  $n$  个数据是表示这  $n$  个产品在 A 车间加工各自所要的时间；

最后的  $n$  个数据是表示这  $n$  个产品在 B 车间加工各自所要的时间。

#### 【输出】

第一行一个数据，表示最少的加工时间；

第二行是一种最小加工时间的加工顺序。

#### 【输入样例】

```
5
3 5 8 7 10
6 2 1 4 9
```

#### 【输出样例】

```
34
1 5 4 2 3
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
```

```

#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 100000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    int num;
    int id;
}m[N];
int a[N],b[N];
int res[N];
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%d",&a[i]);
    for(int i=1;i<=n;i++)
        scanf("%d",&b[i]);

    for(int i=1;i<=n;i++){
        m[i].num=min(a[i],b[i]);
        m[i].id=i;
    }
    for(int i=1;i<=n-1;i++){
        for(int j=i+1;j<=n;j++){
            if(m[i].num>m[j].num){
                swap(m[i],m[j]);
            }
        }
    }
}

```

```

int head=0,tail=n+1;
for(int i=1;i<=n;i++){
    int num=m[i].num;
    int id=m[i].id;
    if(num==a[id])
        res[++head]=id;
    else
        res[--tail]=id;
}

int timeA=0,timeB=0;
for(int i=1;i<=n;i++){
    timeA+=a[res[i]];
    if(timeB<timeA)
        timeB=timeA;
    timeB+=b[res[i]];
}

printf("%d\n",timeB);
for(int i=1;i<=n;i++)
    printf("%d ",res[i]);
printf("\n");
return 0;
}

```

### 1.2.1.1.5 智力大冲浪（信息学奥赛一本通-T1426）

#### 智力大冲浪（信息学奥赛一本通-T1426）

原创 Alex\_McAvoy 发布于 2019-07-21 21:10:21 阅读数 70 收藏  
展开

##### 【题目描述】

小伟报名参加中央电视台的智力大冲浪节目。本次挑战赛吸引了众多参赛者，主持人为了表彰大家的勇气，先奖励每个参赛者  $m$  元。先不要太高兴！因为这些钱还不一定都是你的。接下来主持人宣布了比赛规则：首先，比赛时间分为  $n$  个时段( $n \leq 500$ )，它又给出了很多小游戏，每个小游戏都必须在规定期限  $t_i$  前完成( $1 \leq t_i \leq n$ )。如果一个游戏没能在规定期限前完成，则要从奖励费  $m$  元中扣去一部分钱  $w_i$ ， $w_i$  为自然数，不同的游戏扣去的钱是不一样的。当然，每个游戏本身都很简单，保证每个参赛者都能在一个时段内完成，而且都必须从整时段开始。主持人只是想考考每个参赛者如何

安排组织自己做游戏的顺序。作为参赛者，小伟很想赢得冠军，当然更想赢取最多的钱！ 注意：比赛绝对不会让参赛者赔钱！

### 【输入】

输入共 4 行。

第一行为  $m$ ，表示一开始奖励给每位参赛者的钱；

第二行为  $n$ ，表示有  $n$  个小游戏； 第三行有  $n$  个数，分别表示游戏  $1 \sim n$  的规定完成期限；

第四行有  $n$  个数，分别表示游戏  $1 \sim n$  不能在规定期限前完成的扣款数。

### 【输出】

仅 1 行。表示小伟能赢取最多的钱。

### 【输入样例】

```
10000
7
4 2 4 3 1 4 6
70 60 50 40 30 20 10
```

### 【输出样例】

```
9950
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
```

```

const int N = 100000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    int t;
    int w;
    bool operator < (const Node &rhs)const{
        return w>rhs.w;
    }
}node[N];
bool bucket[N];
int main(){
    int n,m;
    scanf("%d%d",&m,&n);
    for(int i=1;i<=n;i++)
        scanf("%d",&node[i].t);
    for(int i=1;i<=n;i++)
        scanf("%d",&node[i].w);
    sort(node+1,node+1+n);

    int sum=0;
    for(int i=1;i<=n;i++){
        bool flag=true;
        for(int j=node[i].t;j>=1;j--){
            if(!bucket[j]){
                flag=false;
                bucket[j]=true;
                break;
            }
        }

        if(flag){
            for(int j=n;j>=1;j--){
                if(!bucket[j]){
                    bucket[j]=true;
                    break;
                }
            }
            sum+=node[i].w;
        }
    }

    printf("%d\n",m-sum);

```



```

    return 0;
}

```

点赞 1

### 1.2.1.1.6 数列极差（信息学奥赛一本通-T1427）

#### 数列极差（信息学奥赛一本通-T1427）

原创 Alex\_McAvoy 发布于 2019-07-22 07:31:40 阅读数 62 收藏  
展开

##### 【题目描述】

在黑板上写了  $N$  个正整数作成的一个数列，进行如下操作：每一次擦去其中的两个数  $a$  和  $b$ ，然后在数列中加入一个数  $a \times b + 1$ ，如此下去直至黑板上剩下一个数，在所有按这种操作方式最后得到的数中，最大的  $\max$ ，最小的为  $\min$ ，则该数列的极差定义为  $M = \max - \min$ 。

##### 【输入】

第一行，一个数为  $N$ ；

第二行， $N$  个数。

##### 【输出】

输出极差。

##### 【输入样例】

```

3
1 2 3

```

##### 【输出样例】

```

2

```

##### 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>

```

```

#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 100000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int a[N],b[N];
int cmp(int a,int b){
    return a>b;
}
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d",&a[i]);
        b[i]=a[i];
    }

    //最小值
    sort(a+1,a+1+n,cmp);
    for(int i=2;i<=n;i++){
        a[i]=a[i]*a[i-1]+1;
    }
    int minn=a[n];
    //最大值
    sort(b+1,b+1+n);
    for(int i=2;i<=n;i++){
        b[i]=b[i]*b[i-1]+1;
    }
    int maxx=b[n];

    printf("%d\n",maxx-minn);

    return 0;
}

```

## 1.2.1.1.7 数列分段（信息学奥赛一本通-T1428）

### 数列分段（信息学奥赛一本通-T1428）

原创 Alex\_McAvoy 发布于 2019-07-22 07:46:49 阅读数 56 收藏  
展开

#### 【题目描述】

对于给定的一个长度为  $N$  的正整数数列  $A[i]$ ，现要将其分成连续的若干段，并且每段和不超过  $M$ （可以等于  $M$ ），问最少能将其分成多少段使得满足要求。

#### 【输入】

第 1 行包含两个正整数  $N, M$ ，表示了数列  $A[i]$  的长度与每段和的最大值；

第 2 行包含  $N$  个空格隔开的非负整数  $A[i]$ ，如题目所述。

#### 【输出】

一个正整数，输出最少划分的段数。

#### 【输入样例】

```
5 6
4 2 4 5 1
```

#### 【输出样例】

```
3
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
```

```

#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 100000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int a[N];
int main(){
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;i++)
        scanf("%d",&a[i]);

    int sum=0;
    int res=1;
    for(int i=1;i<=n;i++){
        if(sum+a[i]>m){
            sum=a[i];
            res++;
        }
        else
            sum+=a[i];
    }

    printf("%d\n",res);

    return 0;
}

```

点赞

### 1.2.1.1.8 线段（信息学奥赛一本通-T1429）

#### 线段（信息学奥赛一本通-T1429）

原创 Alex\_McAvoy 发布于 2019-07-22 08:00:49 阅读数 62 收藏  
展开

##### 【题目描述】

在一个数轴上有  $n$  条线段，现选取其中  $k$  条线段使得这  $k$  条线段两两没有重合部分，问最大的  $k$  为多少？

**【输入】**

第一行为一个正整数  $n$ ，下面  $n$  行每行 2 个数字  $a_i$ ,  $b_i$ , 描述每条线段。

**【输出】**

输出文件仅包括 1 个整数，为  $k$  的最大值。

**【输入样例】**

```
3
0 2
2 4
1 3
```

**【输出样例】**

```
2
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Line{
    int x,y;
    bool operator < (const Line &rhs)const{
        return y<rhs.y;
    }
}
```

```

}line[N];
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d%d",&line[i].x,&line[i].y);
        sort(line+1,line+1+n);

        int res=1;
        int temp=line[1].y;
        for(int i=2;i<=n;i++){
            if(line[i].x>=temp){
                temp=line[i].y;
                res++;
            }
        }

        printf("%d\n",res);

        return 0;
    }
}

```

### 1.2.1.1.9 家庭作业（信息学奥赛一本通-T1430）

#### 家庭作业（信息学奥赛一本通-T1430）

原创 Alex\_McAvoy 发布于 2019-07-22 11:16:51 阅读数 185 收藏  
展开

##### 【题目描述】

老师在开学第一天就把所有作业都布置了，每个作业如果在规定的时间内交上来的话才有学分。每个作业的截止日期和学分可能是不同的。例如如果一个作业学分为 10，要求在 6 天内交，那么要想拿到这 10 学分，就必须在第 6 天结束前交。

每个作业的完成时间都是只有一天。例如，假设有 7 次作业的学分和完成时间如下：

作业号	1	2	3	4	5	6	7
期限	1	1	3	3	2	2	6
学分	6	7	2	1	4	5	1

最多可以获得 15 学分，其中一个完成作业的次序为 2，6，3，1，7，5，4，注意可能 d 还有其他方法。

你的任务就是找到一个完成作业的顺序获得最大学分。

### 【输入】

第一行一个整数  $N$ ，表示作业的数量。

接下来  $N$  行，每行包括两个整数，第一个整数表示作业的完成期限，第二个数表示该作业的学分。

### 【输出】

输出一个整数表示可以获得的最大学分。保证答案不超过 `longint` 范围。

### 【输入样例】

```
7
1 6
1 7
3 2
3 1
2 4
2 5
6 1
```

### 【输出样例】

```
15
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000000+5;
```

```

const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    int t,w;
    Node(){}
    Node(int t,int w):t(t),w(w){}
    bool operator < (const Node &rhs)const{
        if(t!=rhs.t)
            return t>rhs.t;
        return w<rhs.w;
    }
}node[N];

priority_queue<pair<int,int> > Q;
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d%d",&node[i].t,&node[i].w);
        sort(node+1,node+1+n);

        int j=1;
        int sum=0;
        for(int i=node[1].t;i>=1;i--){
            while(node[j].t==i&&j<=n){
                Q.push(make_pair(node[j].w,node[j].t));
                j++;
            }
            if(!Q.empty()){
                pair<int,int> temp=Q.top();
                sum+=temp.first;
                Q.pop();
            }
        }

        printf("%d\n",sum);

        return 0;
    }
}

```

[点赞](#)  
[收藏](#)  
[分享](#)



## 1.2.1.1.10 钓鱼（信息学奥赛一本通-T1431）

### 钓鱼（信息学奥赛一本通-T1431）

原创 Alex\_McAvoy 发布于 2019-07-22 13:47:15 阅读数 156 收藏  
展开

#### 【题目描述】

在一条水平路边，有  $n$  个钓鱼湖，从左到右编号为  $1, 2, \dots, n$ 。佳佳有  $H$  个小时的空余时间，他希望利用这个时间钓到更多的鱼。他从 1 出发，向右走，有选择的在一些湖边停留一定的时间（是 55 分钟的倍数）钓鱼。最后在某一个湖边结束钓鱼。佳佳从第  $i$  个湖到第  $i+1$  个湖需要走  $5 \times T_i$  分钟路，还测出在第  $i$  个湖停留，第一个 5 分钟可以钓到  $F_i$  条鱼，以后每再钓 5 分钟，可以钓到的鱼量减少  $D_i$ ，若减少后的鱼量小于 0，则减少后的鱼量为 0。为了简化问题，佳佳假定没有其他人钓鱼，也没有其他因素影响他钓到期望数量的鱼。请编程求出佳佳最多能钓鱼的数量。

#### 【输入】

第一行一个整数  $n$ ，表示湖的个数

第二行一个整数  $H$ ，表示佳佳的空闲时间

第三行有  $n$  个整数，依次表示每个湖第一个 5 分钟能钓到鱼的数量

第四行有  $n$  个整数，依次表示以后的每 5 分钟钓鱼数量比前一个 5 分钟钓鱼数量减少的数量

第五行有  $n-1$  个整数， $T_i$  表示由第  $i$  个湖到第  $i+1$  个湖需要花  $5 \times T_i$  分钟的路程

#### 【输出】

输出只有一行，表示佳佳最多能钓鱼的数量。

#### 【输入样例】

```
3
1
4 5 6
1 2 1
1 2
```

#### 【输出样例】

```
35
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
```

```

#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int fish[N],d[N],t[N];
priority_queue<pair<int,int> > Q;
int main(){
    int n,h;
    scanf("%d%d",&n,&h);
    h*=60;
    for(int i=1;i<=n;i++)
        scanf("%d",&fish[i]);
    for(int i=1;i<=n;i++)
        scanf("%d",&d[i]);
    for(int i=1;i<=n-1;i++)
        scanf("%d",&t[i]);

    int walkTime=0;
    int maxx=-INF;
    for(int i=1;i<=n;i++){//可到达的池塘
        int restTime=h-walkTime;
        int res=0;

        for(int j=1;j<=i;j++)
            Q.push(make_pair(fish[j],j));
        while(restTime>0&&Q.top().first>0){//选鱼最多的池塘

```

```

    pair<int,int> temp=Q.top();
    Q.pop();

    res+=temp.first;
    temp.first-=d[temp.second];
    Q.push(temp);
    restTime-=5;
}
maxx=max(maxx,res);
walkTime+=t[i]*5;
}

printf("%d\n",maxx);

return 0;
}

```

点赞

### 1.2.1.1.11 糖果传递（信息学奥赛一本通-T1432）

#### 糖果传递（信息学奥赛一本通-T1432）

原创 Alex\_McAvoy 发布于 2019-07-22 15:26:58 阅读数 71 收藏  
展开

##### 【题目描述】

有  $n$  个小朋友坐成一圈，每人有  $a_i$  个糖果。每人只能给左右两人传递糖果。每人每次传递一个糖果代价为 1。

##### 【输入】

第一行一个正整数  $n \leq 1000000$ ，表示小朋友的个数。

接下来  $n$  行，每行一个整数  $a_i$ ，表示第  $i$  个小朋友得到的糖果的颗数。

##### 【输出】

求使所有人获得均等糖果的最小代价。

##### 【输入样例】

```

4
1
2
5
4

```

## 【输出样例】

4

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
LL a[N];
LL next[N];//给下个人的纸牌数
int main(){
    int n;
    scanf("%d",&n);
    LL sum=0;
    for(int i=1;i<=n;i++){
        scanf("%lld",&a[i]);
        sum+=a[i];
    }

    LL ava=sum/n;//平均数
    for(int i=2;i<=n;i++)//next[i]=a[i]+next[i-1]-sum/n
        next[i]=next[i-1]+a[i]-ava;
    /*
    next[i]-next[i-1]=a[i]-sum/n
    所以 next[i]-next[i-1]+next[i-1]-next[i-2]+...-next[1]=sigma(i)(a[i]-sum/n)

```

```

    即 next[i]=sigma(i)(a[i]-sum/n)+next[1]
    显然 sigma(i)(a[i]-sum/n)是定值
    所以 next[1]取 sigma(i)(a[i]-sum/n)的中位数即可
    */
    sort(next+1,next+1+n);

    LL res=0;
    for(int i=1;i<=n;i++)
        res+=abs(next[i]-next[(n+1)/2]);
    printf("%lld\n",res);

    return 0;
}

```

## 1.2.1.2 第二章：二分与三分

### 1.2.1.2.1 愤怒的牛（信息学奥赛一本通-T1433）

#### 愤怒的牛（信息学奥赛一本通-T1433）

原创 Alex\_McAvoy 发布于 2019-08-05 09:35:28 阅读数 118 收藏  
展开

##### 【题目描述】

农夫 John 建造了一座很长的畜栏，它包括  $N(2 \leq N \leq 100,000)$  个隔间，这些小隔间依次编号为  $x_1, \dots, x_N(0 \leq x_i \leq 1,000,000,000)$ 。但是，John 的  $C(2 \leq C \leq N)$  头牛们并不喜欢这种布局，而且几头牛放在一个隔间里，他们就要发生争斗。为了不让牛互相伤害。John 决定自己给牛分配隔间，使任意两头牛之间的最小距离尽可能的大，那么，这个最大的最小距离是什么呢

##### 【输入】

第一行：空格分隔的两个整数  $N$  和  $C$ ；

第二行---第  $N+1$  行：  $i+1$  行指出了  $x_i$  的位置。

##### 【输出】

一个整数，最大的最小值。

##### 【输入样例】

```

5 3
1 2 8 4 9

```

## 【输出样例】

3

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
#include<string>
#include<cstdlib>
#include<queue>
#include<set>
#include<map>
#include<stack>
#include<vector>
#define INF 0x3f3f3f3f
#define PI acos(-1.0)
#define N 100001
#define MOD 123
#define E 1e-6
using namespace std;
int n,c;
int stall[N];
bool judge(int x) {
    int temp=stall[0];
    int cnt=1;
    for(int i=1; i<n; i++) {
        if(stall[i]-temp>=x) {
            cnt++;
            temp=stall[i];
            if(cnt>=c)
                return true;
        }
    }
    return false;
}
int main() {
    scanf("%d%d",&n,&c);
    for(int i=0; i<n; i++)
        scanf("%d",&stall[i]);

    sort(stall,stall+n);

    int left=0,right=stall[n-1]-stall[0];

```

```

while(left<=right) {
    int mid=(left+right)/2;
    if(judge(mid))
        left=mid+1;
    else
        right=mid-1;
}

printf("%d\n",right);

return 0;
}

```

点赞

## 1.2.1.2.2 Best Cow Fences（信息学奥赛一本通-T1434）

### Best Cow Fences（信息学奥赛一本通-T1434）

原创 Alex\_McAvoy 发布于 2019-08-05 10:08:10 阅读数 59 收藏  
展开

#### 【题目描述】

给定一个长度为  $n$  的正整数序列  $A$ 。求一个平均数最大的，长度不小于  $L$  的子序列。

#### 【输入】

第一行， $n$  和  $L$ ；

$n$  个正整数，表示  $A$ 。

#### 【输出】

一个整数，表示答案的 1000 倍（不用四舍五入，直接输出）。

#### 【输入样例】

```

10 6
6 4 2 10 3 8 5 9 4 1

```

#### 【输出样例】

```

6500

```

#### 【源程序】

```

#include<iostream>
#include<cstdio>

```

```

#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E6;
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

int n,f;
double a[N];
double sub[N];
double sum[N];
int main() {
    scanf("%d%d",&n,&f);
    for(int i=1;i<=n;i++)
        scanf("%lf",&a[i]);

    double left=-1E6,right=1E6;
    while(right-left>1E-5){
        double mid=(left+right)/2.0;
        for(int i=1;i<=n;i++){
            sub[i]=a[i]-mid;//作差
            sum[i]=sum[i-1]+sub[i];//前缀和
        }

        double ans=-1E10;
        double minn=1E10;
        for(int i=f;i<=n;i++){

```



```

        minn=min(minn,sum[i-f]);
        ans=max(ans,sum[i]-minn);
    }

    if(ans>=0)
        left=mid;
    else
        right=mid;
    }
    printf("%d\n",right*1000);

    return 0;
}

```

点赞

### 1.2.1.2.3 曲线（信息学奥赛一本通-T1435）

#### 曲线（信息学奥赛一本通-T1435）

原创 Alex\_McAvoy 发布于 2019-06-04 21:58:02 阅读数 194 收藏  
展开

##### 【题目描述】

明明做作业的时候遇到了  $n$  个二次函数  $S_i(x) = ax^2 + bx + c$ ，他突发奇想设计了一个新的函数  $F(x) = \max(S_i(x)), i = 1 \dots n$ 。

明明现在想求这个函数在  $[0, 1000]$  的最小值，要求精确到小数点后四位四舍五入。

##### 【输入】

输入包含  $T$  组数据 ( $T < 10$ )，每组第一行一个整数  $n (n \leq 10000)$ ，之后  $n$  行，每行 3 个整数  $a (0 \leq a \leq 100)$ ,  $b (|b| \leq 5000)$ ,  $c (|c| \leq 5000)$ ，用来表示每个二次函数的 3 个系数，注意二次函数有可能退化一次。

##### 【输出】

每组数据一个输出，表示新函数  $F(x)$  的在区间  $[0, 1000]$  上的最小值。精确到小数点后四位，四舍五入。

##### 【输入样例】

```

2
1
2 0 0

```

2  
2 0 0  
2 -4 2

**【输出样例】**

0.0000  
0.5000

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E-12;
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int n;
double a[N],b[N],c[N];
double cal(double x){
    double maxx=-INF;
    for(int i=1;i<=n;i++){
        maxx=max(maxx,a[i]*x*x+b[i]*x+c[i]);
    }
    return maxx;
}
int main() {
    int t;
    scanf("%d",&t);
```

```

while(t--){
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%lf%lf%lf",&a[i],&b[i],&c[i]);
        double left=0,right=1000;
        while(right-left>=EPS){
            double lmid=left+(right-left)/3.0;
            double rmid=right-(right-left)/3.0;
            if(cal(lmid)<=cal(rmid))
                right=rmid;
            else
                left=lmid;
        }
        printf("%.4lf\n",cal(left));
    }

    return 0;
}

```

## 1.2.1.2.4 数列分段 II（信息学奥赛一本通-T1436）

### 数列分段 II（信息学奥赛一本通-T1436）

原创 Alex\_McAvoy 发布于 2019-08-05 10:20:05 阅读数 375 收藏  
展开

#### 【题目描述】

对于给定的一个长度为  $N$  的正整数数列  $A[i]$ ，现要将其分成  $M$  ( $M \leq N$ ) 段，并要求每段连续，且每段和的最大值最小。

关于最大值最小：

例如一数列 4 2 4 5 1 要分成 3 段

将其如下分段：

[4 2][4 5][1]

第一段和为 6，第 2 段和为 9，第 3 段和为 1，和最大值为 9。

将其如下分段：

[4][2 4][5 1]

第一段和为 4，第 2 段和为 6，第 3 段和为 6，和最大值为 6。

并且无论如何分段，最大值不会小于 6。

所以可以得到要将数列 4 2 4 5 1 要分成 3 段，每段和的最大值最小为 6。

#### 【输入】

第 1 行包含两个正整数  $N, M$ ，第 2 行包含  $N$  个空格隔开的非负整数  $A[i]$ ，含义如题目所述。

#### 【输出】

仅包含一个正整数，即每段和最大值最小为多少。

#### 【输入样例】

```
5 3
4 2 4 5 1
```

#### 【输出样例】

```
6
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E6;
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
```

```

int n,m,a[N];
bool judge(int x) {
    int sum=0,group=1;
    for(int i=1; i<=n; i++) {
        sum+=a[i];
        if(sum>x) {
            sum=a[i];
            group++;
        }
    }
    if(group<=m)
        return true;
    else
        return false;
}

int main() {
    int left=0,right=0;
    scanf("%d%d",&n,&m);
    for(int i=1; i<=n; i++) {
        scanf("%d",&a[i]);
        left=max(left,a[i]);
        right+=a[i];
    }

    while(left+1<right) {
        int mid=(left+right)/2;
        if(judge(mid))
            right=mid;
        else
            left=mid;
    }

    if(judge(left))
        printf("%d\n",left);
    else
        printf("%d\n",right);
    return 0;
}

```

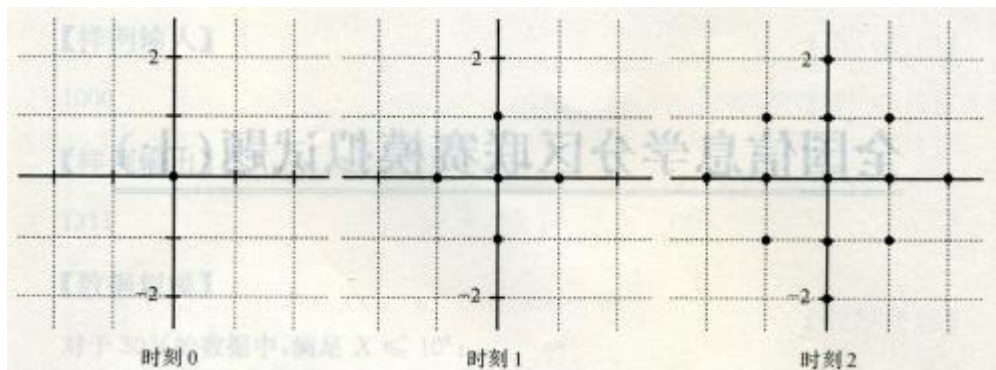
### 1.2.1.2.5 扩散（信息学奥赛一本通-T1437）

扩散（信息学奥赛一本通-T1437）

原创 Alex\_McAvoy 发布于 2019-08-06 09:21:52 阅读数 120 收藏  
展开

### 【题目描述】

一个点每过一个单位时间就会向四个方向扩散一个距离，如图。



两个点  $a$ 、 $b$  连通，记作  $e(a,b)$ ，当且仅当  $a$ 、 $b$  的扩散区域有公共部分。连通块的定义是块内的任意两个点  $u$ 、 $v$  都必定存在路径  $e(u,a_0), e(a_0,a_1), \dots, e(a_k,v)$ 。给定平面上的  $n$  个点，问最早什么时刻它们形成一个连通块。

### 【输入】

第一行一个数  $n$ ，以下  $n$  行，每行一个点坐标。

### 【输出】

一个数，表示最早的时刻所有点形成连通块。

### 【输入样例】

```
2
0 0
5 5
```

### 【输出样例】

```
5
```

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
```

```

#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E-10;
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Edge{
    int x,y;
    int dis;
    Edge(){}
    Edge(int x,int y,int dis):x(x),y(y),dis(dis){}
    bool operator < (const Edge &rhs)const{
        return dis<rhs.dis;
    }
}edge[N];
struct Node{
    int x,y;
    Node(){}
    Node(int x,int y):x(x),y(y){}
}node[N];
int tot;
int father[N];
int getDis(int i,int j){
    return abs(node[i].x-node[j].x)+abs(node[i].y-node[j].y);
}
int Find(int x){
    return father[x]==x?x:father[x]=Find(father[x]);
}
int Kruskal(int n){
    for(int i=1;i<=n;i++)
        father[i]=i;
    int num=0;
    int res=-INF;
    for(int i=1;i<=tot;i++){
        int x=edge[i].x;

```

```

    int y=edge[i].y;
    x=Find(x);
    y=Find(y);
    if(x!=y){
        res=max(res,edge[i].dis);
        num++;
        father[x]=y;
    }
    if(num==n)
        break;
}
return res;
}
int main(){
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%d%d",&node[i].x,&node[i].y);

    for(int i=1;i<n;i++){
        for(int j=i+1;j<=n;j++){
            edge[++tot].x=i;
            edge[tot].y=j;
            edge[tot].dis=getDis(i,j);
        }
    }

    sort(edge+1,edge+1+tot);
    int res=Kruskal(n);
    res=(res+1)/2;
    printf("%d\n",res);
    return 0;
}

```

[点赞](#)  
[收藏](#)  
[分享](#)  
[站内首发文章](#)

**Alex\_McAvoy**

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

[他的留言板](#)

[已关注](#)

[展开阅读全文](#)



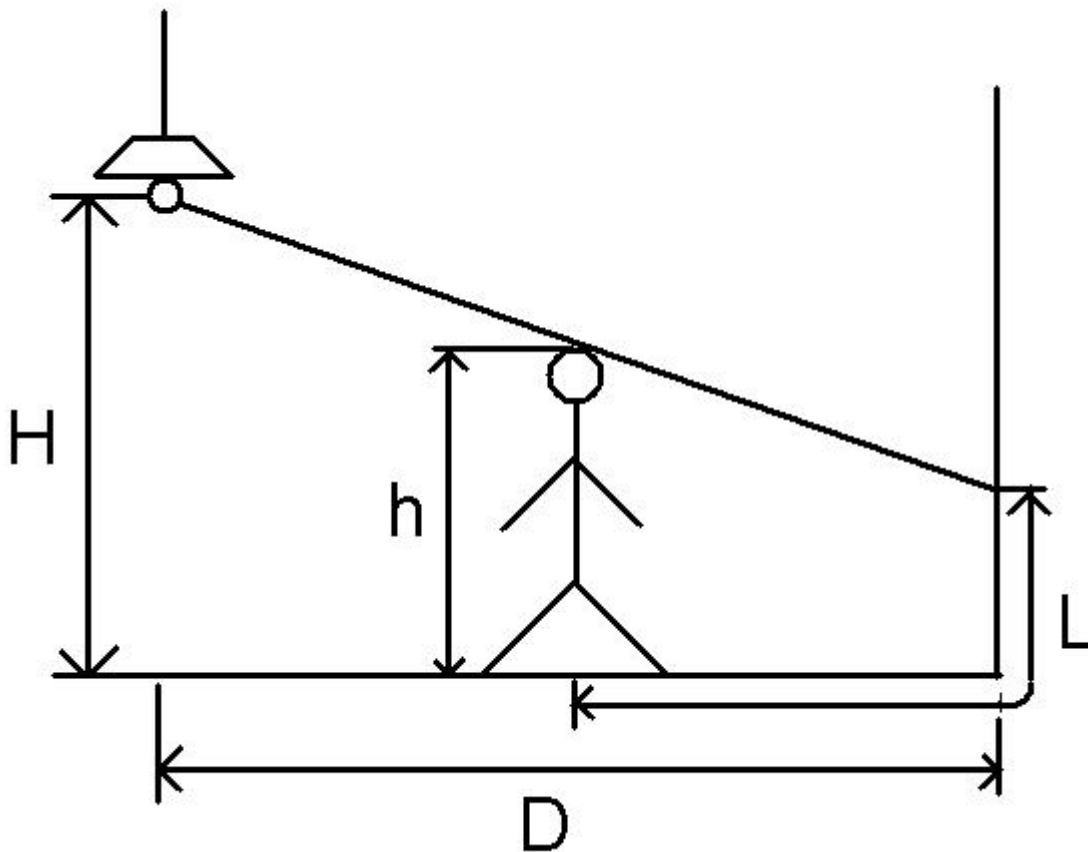
### 1.2.1.2.6 灯泡（信息学奥赛一本通-T1438）

#### 灯泡（信息学奥赛一本通-T1438）

原创 Alex\_McAvoy 发布于 2019-08-05 15:54:26 阅读数 125 收藏  
展开

##### 【题目描述】

相比 Wildleopard 的家，他的弟弟 Mildleopard 比较穷，他的房子是狭窄的，而且在他的房间里只有一个灯泡，每天晚上，他徘徊在自己狭小的房子里，思考如何赚更多的钱。有一天，他发现他的影子的长度随着他在灯泡和墙壁之间走动时会发生变化。一个突然的想法出现在他的脑海里，他想知道在房间里他影子的最大长度



##### 【输入】

第一行包含一个整数  $T$  ( $T \leq 100$ ), 表示测试数据的组数

对于每组测试数据仅一行包含三个实数  $H, h, D$ ,  $H$  表示灯泡的高度,  $h$  表示 Mildleopard 的身高,  $D$  表示灯泡和墙的水平距离。  $10^{-2} \leq H, h, d \leq 10^3$ ,  $H - h \geq 10^{-2}$

**【输出】**

T 行，每行一个数，表示影子的最大长度，保留三位小数

**【输入样例】**

```
3
2 1 0.5
2 0.5 3
4 3 4
```

**【输出样例】**

```
1.000
0.750
4.000
```

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E-10;
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
double H,h,D;
double cal(double x){
    double tanA=(H-h)/x;
    double d=H/tanA;
    if(d<=D)
```

```

        return d-x;
    d=d-D;
    return D-x+d*tanA;
}
int main() {
    int t;
    scanf("%d",&t);
    while(t--){
        scanf("%lf%lf%lf",&H,&h,&D);
        double left=0,right=D;
        while(right-left>=EPS){
            double lmid=left+(right-left)/3.0;
            double rmid=right-(right-left)/3.0;
            if(cal(lmid)<cal(rmid))
                left=lmid;
            else
                right=rmid;
        }
        printf("%.3lf\n",cal(left));
    }
    return 0;
}

```

### 1.2.1.2.7 传送带（信息学奥赛一本通-T1439）

#### 传送带（信息学奥赛一本通-T1439）

原创 Alex\_McAvoy 发布于 2019-08-06 10:54:25 阅读数 90 收藏  
展开

##### 【题目描述】

对于给定的一个长度为  $N$  的正整数数列  $A[i]$ ，现要将其分成  $M$  ( $M \leq N$ ) 段，并要求每段连续，且每段和的最大值最小。

关于最大值最小：

例如一数列 4 2 4 5 1 要分成 3 段

将其如下分段：

[4 2][4 5][1]

第一段和为 6，第 2 段和为 9，第 3 段和为 1，和最大值为 9。

将其如下分段：

[4][2 4][5 1]

第一段和为 4，第 2 段和为 6，第 3 段和为 6，和最大值为 6。

并且无论如何分段，最大值不会小于 6。

所以可以得到要将数列 4 2 4 5 1 要分成 3 段，每段和的最大值最小为 6。

#### 【输入】

第 1 行包含两个正整数  $N$ ， $M$ ，第 2 行包含  $N$  个空格隔开的非负整数  $A[i]$ ，含义如题目所述。

#### 【输出】

仅包含一个正整数，即每段和最大值最小为多少。

#### 【输入样例】

```
5 3
4 2 4 5 1
```

#### 【输出样例】

```
6
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const double EPS = 1E-10;
const int MOD = 1E9+7;
const int N = 10000+5;
```

```

const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
struct Node{
    double x,y;
    Node(){}
    Node(double x,double y):x(x),y(y){}
}a,b,c,d;
double p,q,r;
double getDis(double x1,double y1,double x2,double y2){
    return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
}
double judge(Node e){
    Node left(c.x,c.y),right(d.x,d.y);
    int times=100;//控制三分次数
    while(times--){
        Node lmid,rmid;
        lmid.x=left.x+(right.x-left.x)/3;
        lmid.y=left.y+(right.y-left.y)/3;
        rmid.x=right.x-(right.x-left.x)/3;
        rmid.y=right.y-(right.y-left.y)/3;
        double dis1=getDis(lmid.x,lmid.y,d.x,d.y)/q+getDis(lmid.x,lmid.y,e.x,e.y)/r;
        double dis2=getDis(rmid.x,rmid.y,d.x,d.y)/q+getDis(rmid.x,rmid.y,e.x,e.y)/r;

        if(dis1<dis2){
            right.x=rmid.x;
            right.y=rmid.y;
        }
        else{
            left.x=lmid.x;
            left.y=lmid.y;
        }
    }

    return
    getDis(left.x,left.y,d.x,d.y)/q+getDis(left.x,left.y,e.x,e.y)/r+getDis(e.x,e.y,a.x,a.y)/p;
}
int main(){
    scanf("%lf%lf",&a.x,&a.y);
    scanf("%lf%lf",&b.x,&b.y);
    scanf("%lf%lf",&c.x,&c.y);
    scanf("%lf%lf",&d.x,&d.y);
    scanf("%lf%lf%lf",&p,&q,&r);

```

```

Node left(a.x,a.y),right(b.x,b.y);
int times=100;//控制三分次数
while(times--){
    Node lmid,rmid;
    lmid.x=left.x+(right.x-left.x)/3;
    lmid.y=left.y+(right.y-left.y)/3;
    rmid.x=right.x-(right.x-left.x)/3;
    rmid.y=right.y-(right.y-left.y)/3;

    if(judge(lmid)<judge(rmid)){
        right.x=rmid.x;
        right.y=rmid.y;
    }
    else{
        left.x=lmid.x;
        left.y=lmid.y;
    }
}
printf("%.2lf",judge(left));

return 0;
}

```

点赞

收藏

分享

Alex\_McAvoy

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+

他的留言板

已关注

展开阅读全文

### 1.2.1.3 第三章：深搜的剪枝技巧

Help > [信息学奥赛一本通](#) > [算法提高篇](#) > [第一部分：基础算法](#) >

第三章：深搜的剪枝技巧

#### 1.2.1.3.1 数的划分（信息学奥赛一本通-T1440）

数的划分（信息学奥赛一本通-T1440）

原创 Alex\_McAvoy 发布于 2019-07-23 14:28:22 阅读数 154 收藏  
展开

### 【题目描述】

将整数  $n$  分成  $k$  份，且每份不能为空，任意两份不能相同(不考虑顺序)。

例如：  $n=7$ ，  $k=3$ ，下面三种分法被认为是相同的。

1, 1, 5; 1, 5, 1; 5, 1, 1;

问有多少种不同的分法。 输出一个整数，即不同的分法

### 【输入】

两个整数  $n$ ，  $k$  ( $6 < n \leq 200$ ，  $2 \leq k \leq 6$ )，中间用单个空格隔开。

### 【输出】

一个整数，即不同的分法。

### 【输入样例】

7 3

### 【输出样例】

4

### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000000+5;
```

```

const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int res;
int a[N];
int n,k;
void dfs(int sum,int step){
    if(step==k){
        if(sum>=a[step-1])
            res++;
        return;
    }

    for(int i=a[step-1];i<=sum/(k-step+1);i++){//上下界剪枝
        a[step]=i;
        sum-=i;
        dfs(sum,step+1);
        sum+=i;
    }
}

int main() {
    scanf("%d%d",&n,&k);
    a[0]=1;
    dfs(n,1);
    printf("%d\n",res);
    return 0;
}

```

点赞

### 1.2.1.3.2 生日蛋糕（信息学奥赛一本通-T1441）

#### 生日蛋糕（信息学奥赛一本通-T1441）

原创 Alex\_McAvoy 发布于 2019-07-24 11:23:55 阅读数 85 收藏  
展开

##### 【题目描述】

7 月 17 日是 Mr.W 的生日，ACM-THU 为此要制作一个体积为  $N\pi$  的  $M$  层生日蛋糕，每层都是一个圆柱体。设从下往上数第  $i$  ( $1 \leq i \leq M$ ) 层蛋糕是半径为  $R_i$ ，高度为  $H_i$  的圆柱。



当  $i < M$  时, 要求  $R_i > R_{i+1}$  且  $H_i > H_{i+1}$ 。由于要在蛋糕上抹奶油, 为尽可能节约经费, 我们希望蛋糕外表面 (最下一层的下底面除外) 的面积  $Q$  最小。

令  $Q = S\pi$ , 请编程对给出的  $N$  和  $M$ , 找出蛋糕的制作方案 (适当的  $R_i$  和  $H_i$  的值), 使  $S$  最小。

(除  $Q$  外, 以上所有数据皆为正整数)

#### 【输入】

有两行, 第一行为  $N (N \leq 10000)$ , 表示待制作的蛋糕的体积为  $N\pi$ ; 第二行为  $M (M \leq 20)$ , 表示蛋糕的层数为  $M$ 。

#### 【输出】

仅一行, 是一个正整数  $S$  (若无解则  $S=0$ )。

#### 【输入样例】

100  
2

#### 【输出样例】

68

#### 【提示】

附: 圆柱公式

体积  $V = \pi R^2 H$

侧面积  $A = 2\pi R H$

底面积  $A = \pi R^2$

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
```

```

#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 500000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

int n,m;
int minn=INF;
//step 为前 step 层, v、s 为前 step 层体积、面积, r、h 为第 step 层半径、高
void dfs(int step,int v,int s,int r,int h){
    if(step==m){//层数搜索完, 返回
        if(v==n)//体积为所给体积, 更新最小面积
            minn=s;
        return;
    }

    if(v+(r-1)*(r-1)*(h-1)*(m-step)<n)//当前层体积与每层体积最大值之和小于 n, 不符合, 返回
        return;
    if(v+1*1*1*(m-step)>n)//当前层体积与每层体积最小值之和大于 n, 不符合, 返回
        return;
    if(2*(n-v)/r+s>minn)//侧面积加上前 step 层面积大于最小值, 不符合, 返回
        return;

    for(int i=r-1;i>=m-step;i--){//上一层的半径的最小值要保证大于当前层半径的最小值
        for(int j=h-1;j>=m-step;j--){//上一层高度的最小值要保证大于当前层高度的最小值
            int area=s+2*i*j;//面积
            int volume=v+i*i*j;//体积
            if(area<minn&&volume<=n)
                dfs(step+1,volume,area,i,j);
        }
    }
}

int main(){
    scanf("%d%d",&n,&m);
    /*
    i 为半径,j 为高
    第 1 层为 1,第 2 层为 2,...,第 m 层为 m, 故半径最小为 m, 同理, 高最小也是 m
    根据题意, 蛋糕体积  $V=n*\pi$ , 则半径 i 的平方  $i*i$  为底面积, 乘以层数 m 即为体积
    则有不等式:  $m*(i*i*\pi)<=n*\pi$ , 即:  $i*i*m<=n$ 
    */
}

```

```

    同理，对于高度 j，有：  $i*j \leq n$ 
    */
    for(int i=m;i*i*m<=n;i++){
        for(int j=m;j*i*i<=n;j++){
            int flankArea=2*i*j;//侧面积
            int bottomArea=i*i;//底面积
            int area=flankArea+bottomArea;
            int volume=i*i*j;//体积
            if(area<minn)
                dfs(1,volume,area,i,j);
        }
    }
    printf("%d\n",minn);
    return 0;
}

```

### 1.2.1.3.3 小木棍（信息学奥赛一本通-T1442）

#### 小木棍（信息学奥赛一本通-T1442）

原创 Alex\_McAvoy 发布于 2019-07-24 10:19:55 阅读数 101 收藏  
展开

##### 【题目描述】

乔治有一些同样长的小木棍，他把这些木棍随意砍成几段，直到每段的长都不超过 50。现在，他想把小木棍拼接成原来的样子，但是却忘记了自己开始时有多少根木棍和它们的长度。给出每段小木棍的长度，编程帮他找出原始木棍的最小可能长度。

##### 【输入】

第一行为一个单独的整数  $N$  表示砍过以后的小木棍的总数，其中  $N \leq 60$ ，第二行为  $N$  个用空格隔开的正整数，表示  $N$  根小木棍的长度。

##### 【输出】

仅一行，表示要求的原始木棍的最小可能长度。

##### 【输入样例】

```

9
5 2 1 5 2 1 5 2 1

```

##### 【输出样例】

```

6

```

## 【源程序】

```

#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int a[N];
bool vis[N];
int cnt;
bool cmp(int x,int y){
    return x>y;
}
bool dfs(int k,int step,int rest,int len){
    //k 为已经拼接好的个数,step 为上一个的编号,rest 为剩余的长度,len 为木棍的长度
    if(k==cnt+1 && rest==0)//拼接好个数为要求数且无剩余
        return true;
    else if(k==cnt+1)//拼接好个数为要求数但有剩余
        return false;
    else if(rest==0){//拼接好个数不为要求数但无剩余, 重新开始
        rest=len;//剩余数变为长度
        step=0;//编号归零
    }

    for(int i=step+1;i<=cnt;i++){
        if(!vis[i]){

```

```

    if(rest-a[i]>=0){//保证剩余值不为负数
        vis[i]=true;
        if(dfs(k+1,i,rest-a[i],len))
            return true;
        vis[i]=false;
        if(a[i]==rest || len==rest)//头尾剪枝，此时已在回溯之后，需要判断头尾两
种情况
            break;
        while(a[i]==a[i+1])//去重剪枝，用当前长度搜索无结果时，对同样长度的可
以忽略
            i++;
    }
}

return 0;
}

int main(){
    int n;
    scanf("%d",&n);

    int sum=0;
    for(int i=1;i<=n;i++){
        int x;
        scanf("%d",&x);
        if(x<=50){
            a[++cnt]=x;
            sum+=x;
        }
    }
    sort(a+1,a+1+cnt,cmp);

    for(int i=a[1];i<=sum;i++){
        if(sum%i==0){
            if(dfs(1,0,i,i)){
                printf("%d\n",i);
                break;
            }
        }
    }
    return 0;
}

```

## 1.2.1.3.4 Addition Chains（信息学奥赛一本通-T1443）

### Addition Chains（信息学奥赛一本通-T1443）

原创 Alex\_McAvoy 发布于 2019-07-24 14:33:15 阅读数 69 收藏  
展开

#### 【题目描述】

已知一个数列  $a_0, a_1, \dots, a_m$ ，其中  $a_0=1, a_m=n; a_0 < a_1 < a_2 < \dots < a_{m-1} < a_m$ 。对于每个  $k(1 \leq k \leq m)$  满足  $a_k = a_i + a_j (0 \leq i, j \leq k-1)$ ，这里  $i$  与  $j$  可以相等。

现给定  $n$  的值，要求  $m$  的最小值（并不要求输出）及这个数列的值（可能存在多个数列，只输出任意一个满足条件的就可以）。

#### 【输入】

多组数据，每行给定一个正整数  $n$ 。输入以 0 结束。

#### 【输出】

对于每组数据，输出满足条件的长度最小的数列。

#### 【输入样例】

```
5
7
12
15
77
0
```

#### 【输出样例】

```
1 2 4 5
1 2 4 6 7
1 2 4 8 12
1 2 4 5 10 15
1 2 4 8 9 17 34 68 77
```

#### 【源程序】

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
```

```

#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
const int N = 1000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

int n;
int a[N];
bool vis[N];
int cnt;
bool dfs(int step){
    if(step>cnt){
        if(a[cnt]==n)
            return true;
        return false;
    }

    for(int i=step-1;i>=1;i--){//搜索顺序优化
        for(int j=i;j>=1;j--){
            int temp=a[i]+a[j];
            if(temp>n)//可行性剪枝
                continue;
            if(!vis[temp]){
                if(temp<=a[step-1])
                    return false;

                vis[temp]=true;
                a[step]=temp;
                if(dfs(step+1))
                    return true;
                vis[temp]=false;
            }
        }
    }
}

```

```

    }
}
int main(){
    while(scanf("%d",&n)!=EOF&&n){
        memset(a,0,sizeof(a));

        if(n==1)
            printf("1\n");
        else if(n==2)
            printf("1 2\n");
        else{
            a[1]=1;
            a[2]=2;
            cnt=3;
            while(!dfs(3)){
                memset(vis,false,sizeof(vis));
                cnt++;
            }
            for(int i=1;i<=cnt;i++)
                printf("%d ",a[i]);
            printf("\n");
        }
    }
    return 0;
}

```

点赞

收藏

分享

### 1.2.1.3.5 埃及分数（信息学奥赛一本通-T1444）

#### 埃及分数（信息学奥赛一本通-T1444）

原创 Alex\_McAvoy 发布于 2019-07-29 17:30:52 阅读数 95 收藏  
展开

##### 【题目描述】

在古埃及，人们使用单位分数的和(形如  $1/a$  的,  $a$  是自然数)表示一切有理数。如： $2/3=1/2+1/6$ ,但不允许  $2/3=1/3+1/3$ ,因为加数中有相同的。对于一个分数  $a/b$ ,表示方法有很多种，但是哪种最好呢？首先，加数少的比加数多的好，其次，加数个数相同的，最小的分数越大越好。



如:  $19/45=1/3 + 1/12 + 1/180$

$19/45=1/3 + 1/15 + 1/45$

$19/45=1/3 + 1/18 + 1/30,$

$19/45=1/4 + 1/6 + 1/180$

$19/45=1/5 + 1/6 + 1/18.$

最好的是最后一种, 因为  $1/18$  比  $1/180, 1/45, 1/30, 1/180$  都大。

给出  $a, b(0 < a < b < 1000)$ , 编程计算最好的表达方式。

**【输入】**

输入:  $a\ b$

**【输出】**

若干个数, 自小到大排列, 依次是单位分数的分母。

**【输入样例】**

19 45

**【输出样例】**

5 6 18

**【源程序】**

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
const int MOD = 1E9+7;
```

```

const int N = 200+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

LL maxDeep;
LL temp[N];
LL res[N];
LL GCD(LL a,LL b){
    return b==0?a:GCD(b,a%b);
}
bool judge(LL step){//当最小分数分母比原来大进行更新
    if(res[step]==-1)
        return true;
    else if(temp[step]>res[step])
        return false;
    else if(temp[step]<res[step])
        return true;
    else
        return false;
}
LL getLimit(LL x,LL y){
    for(LL i=2;;i++){
        if(y<x*i)// 1/i<=x/y
            return i;
    }
}
bool dfs(LL step,LL minn,LL x,LL y){
    if(step==maxDeep){
        if(y%x)
            return false;
        else{
            temp[step]=y/x;
            if(judge(step))//存在更优解，更新答案
                memcpy(res,temp,sizeof(temp));
            return true;
        }
    }
    minn=max(minn,getLimit(x,y));//取得新下界，注意这里取的是 max（分母），以防止漏除枚举。

    bool flag=false;
    for(LL i=minn;;i++){
        if((maxDeep-step+1)*y<=x*i)//剪枝，(maxDeep-step+1)*(1/i)<=x/y
            break;

```

```

temp[step]=i;
LL ny=y*i;
LL nx=x*i-y;

LL gcd=GCD(nx,ny);
if(dfs(step+1,minn+1,nx/gcd,ny/gcd))
    flag=true;
}
return flag;
}
int main(){
    LL n,m;
    scanf("%lld%lld",&n,&m);
    for(maxDeep=1;;maxDeep++){
        LL limit=getLimit(n,m);
        memset(temp,0,sizeof(temp));
        memset(res,-1,sizeof(res));
        if(dfs(0,limit,n,m))//找到第一个可行解即退出
            break;
    }
    for(LL i=0;i<=maxDeep;i++)
        printf("%lld ",res[i]);
    return 0;
}

```

点赞

### 1.2.1.3.6 平板涂色（信息学奥赛一本通-T1445）

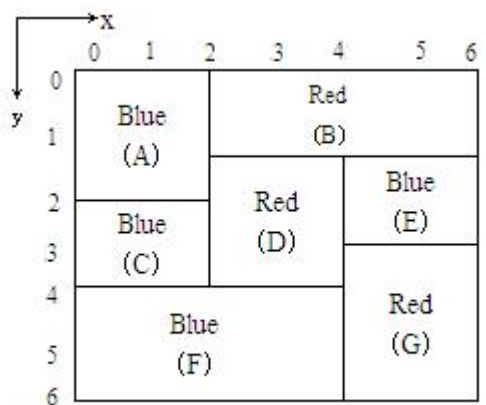
#### 平板涂色（信息学奥赛一本通-T1445）

原创 Alex\_McAvoy 发布于 2019-08-03 17:00:00 阅读数 83 收藏  
展开

##### 【题目描述】

CE 数码公司开发了一种名为自动涂色机（APM）的产品。它能用预定的颜色给一块由不同尺寸且互不覆盖的矩形构成的平板涂色。

为了涂色，APM 需要使用一组刷子。每个刷子涂一种不同的颜色 C。APM 拿起一把有颜色 C 的刷子，并给所有颜色为 C 且符合下面限制的矩形涂色：



为了避免颜料渗漏使颜色混合，一个矩形只能在所有紧靠它上方的矩形涂色后，才能涂色。例如图中矩形 **F** 必须在 **C** 和 **D** 涂色后才能涂色。注意，每一个矩形必须立刻涂满，不能只涂一部分。

写一个程序求一个使 **APM** 拿起刷子次数最少的涂色方案。注意，如果一把刷子被拿起超过一次，则每一次都必须记入总数中。

**【输入】**

第一行为矩形的个数 **N**。下面有 **N** 行描述了 **N** 个矩形。每个矩形有 5 个整数描述，左上角的 **y** 坐标和 **x** 坐标，右下角的 **y** 坐标和 **x** 坐标，以及预定颜色。

颜色号为 1 到 20 的整数。

平板的左上角坐标总是(0, 0)。

坐标的范围是 0..99。N 小于 16。

**【输出】**

拿起刷子的最少次数。

**【输入样例】**

```
7
0 0 2 2 1
0 2 1 6 2
2 0 4 2 1
1 2 4 4 2
1 4 3 6 1
4 0 6 4 1
3 4 6 6 2
```

**【输出样例】**

```
3
```

**【源程序】**

```
#include<iostream>
```

```

#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
#define Pair pair<int,int>
const int MOD = 1E9+7;
const int N = 1000000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;

struct Rectangle{
    int x1,y1,x2,y2;
    int color;
}rect[20];
bool vis[20];
bool G[40][40];
int res=15;
bool judge(int x) {
    if(rect[x].x1==0)
        return true;
    for(int i=rect[x].y1+1;i<=rect[x].y2;i++)
        if(G[rect[x].x1][i]==0)
            return false;
    return true;
}
void update(int x,int sta){
    for(int i=rect[x].x1+1;i<=rect[x].x2;i++)
        for(int j=rect[x].y1+1;j<=rect[x].y2;j++)
            G[i][j]=sta;
}

```

```

}
void dfs(int p,int k,int num,int n){//第 p 块板子刷第 k 次，累积了 num 块板子
    if(k>res)//可行性剪枝
        return;
    if(num==n)
        res=min(res,k);

    for(int i=1;i<=n;i++){
        if(!vis[i]&&judge(i)){
            update(i,1);
            vis[i]=true;
            num++;
            if(rect[i].color==rect[p].color)
                dfs(i,k,num,n);
            else{
                k++;
                dfs(i,k,num,n);
                k--;
            }
            update(i,0);
            vis[i]=false;
            num--;
        }
    }
}

int main() {
    int n;
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%d%d%d%d%d",&rect[i].x1,&rect[i].y1,&rect[i].x2,&rect[i].y2,&rect[i].color);
    dfs(0,0,0,n);
    printf("%d\n",res);
    return 0;
}

```

点赞

收藏

分享

Alex\_McAvoy

发布了 1870 篇原创文章 · 获赞 663 · 访问量 187 万+  
他的留言板

已关注  
展开阅读全文

### 1.2.1.3.7 靶形数独（信息学奥赛一本通-T1447）

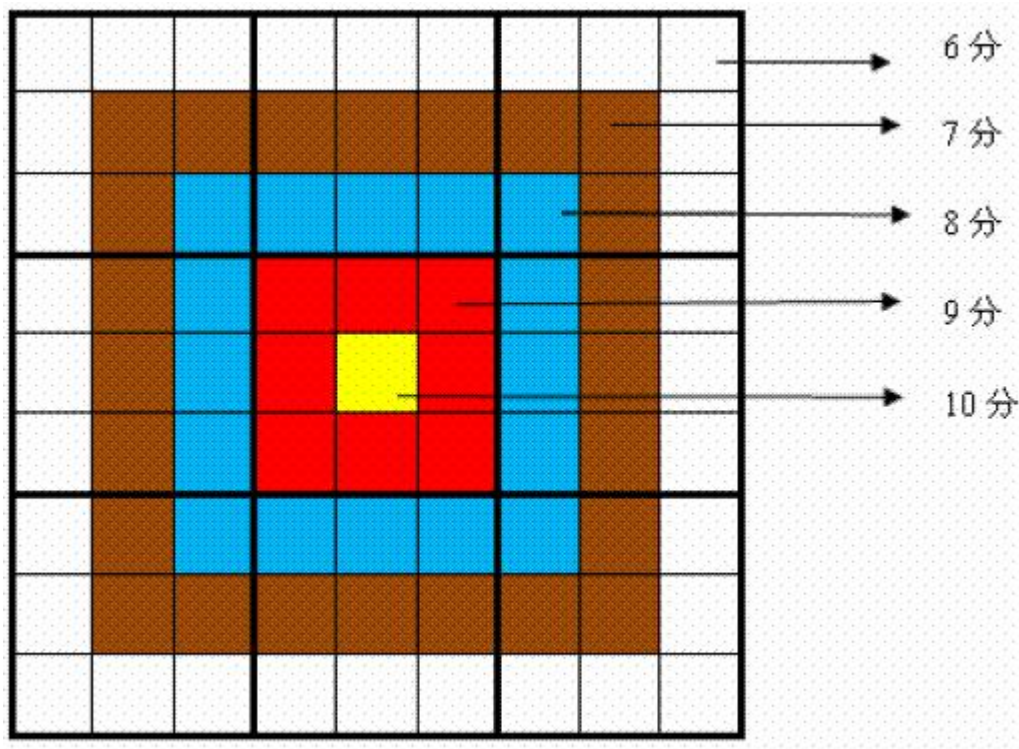
#### 靶形数独（信息学奥赛一本通-T1447）

原创 Alex\_McAvoy 发布于 2019-04-29 20:22:43 阅读数 307 收藏  
展开

##### 【题目描述】

小城和小华都是热爱数学的好学生，最近，他们不约而同地迷上了数独游戏，好胜的他们想用数独来一比高低。但普通的数独对他们来说都过于简单了，于是他们向 Z 博士请教，Z 博士拿出了他最近发明的“靶形数独”，作为这两个孩子比试的题目。

靶形数独的方格同普通数独一样，在 9 格宽×9 格高的大九宫格中有 9 个 3 格宽×3 格高的小九宫格（用粗黑色线隔开的）。在这个大九宫格中，有一些数字是已知的，根据这些数字，利用逻辑推理，在其他的空格上填入 1 到 9 的数字。每个数字在每个小九宫格内不能重复出现，每个数字在每行、每列也不能重复出现。但靶形数独有一点和普通数独不同，即每一个方格都有一个分值，而且如同一个靶子一样，离中心越近则分值越高。（如图）



上图具体的分值分布是：最里面一格（黄色区域）为 10 分，黄色区域外面的一圈（红色区域）每个格子为 9 分，再外面一圈（蓝色区域）每个格子为 8 分，蓝色区域外面一圈（棕色区域）每个格子为 7 分，最外面一圈（白色区域）每个格子为 6 分，如上图所示。比赛的要求是：每个人必须完成一个给定的数独（每个给定数独可能有不同的填法），而且要争取更高的总分数。而这个总分数即每个方格上的分值和完成这个数独时填在相应格上的数字的乘积的总和。如图，在以下的这个已经填完数字的靶形数独游戏中，总分数为 2829。游戏规定，将以总分的高低决出胜负。

7	5	4	9	3	8	2	6	1
1	2	8	6	4	5	9	3	7
6	3	9	2	1	7	4	8	5
8	6	5	4	2	9	1	7	3
9	7	2	3	5	1	6	4	8
4	1	3	8	7	6	5	2	9
5	4	7	1	8	2	3	9	6
2	9	1	7	6	3	8	5	4
3	8	6	5	9	4	7	1	2

由于求胜心切，小城找到了善于编程的你，让你帮他求出，对于给定的靶形数独，能够得到的最高分数。

#### 【输入】

一共 9 行。每行 9 个整数（每个数都在 0—9 的范围内），表示一个尚未填满的数独方格，未填的空格用“0”表示。每两个数字之间用一个空格隔开。

#### 【输出】

共 1 行。输出可以得到的靶形数独的最高分数。如果这个数独无解，则输出整数-1。

#### 【输入样例】

样例 1

```
7 0 0 9 0 0 0 0 1
1 0 0 0 0 5 9 0 0
0 0 0 2 0 0 0 8 0
0 0 5 0 2 0 0 0 3
0 0 0 0 0 0 6 4 8
```



4 1 3 0 0 0 0 0  
 0 0 7 0 0 2 0 9 0  
 2 0 1 0 6 0 8 0 4  
 0 8 0 5 0 4 0 1 2

样例 2

0 0 0 7 0 2 4 5 3  
 9 0 0 0 0 8 0 0 0  
 7 4 0 0 0 5 0 1 0  
 1 9 5 0 8 0 0 0 0  
 0 7 0 0 0 0 0 2 5  
 0 3 0 5 7 9 1 0 8  
 0 0 0 6 0 1 0 0 0  
 0 6 0 9 0 0 0 0 1  
 0 0 0 0 0 0 0 0 6

【输出样例】

样例 1

2829

样例 2

2852

源程序

```
#include<iostream>
#include<cstdio>
#include<cstdlib>
#include<string>
#include<cstring>
#include<cmath>
#include<ctime>
#include<algorithm>
#include<utility>
#include<stack>
#include<queue>
#include<vector>
#include<set>
#include<map>
#include<bitset>
#define EPS 1e-9
#define PI acos(-1.0)
#define INF 0x3f3f3f3f
#define LL long long
```

```

const int MOD = 1E9+7;
const int N = 1000+5;
const int dx[] = {-1,1,0,0,-1,-1,1,1};
const int dy[] = {0,0,-1,1,-1,1,-1,1};
using namespace std;
int g[N][N];
int grade[10][10] { //得分
    {0,0,0,0,0,0,0,0,0},
    {0,6,6,6,6,6,6,6,6},
    {0,6,7,7,7,7,7,7,6},
    {0,6,7,8,8,8,8,8,7,6},
    {0,6,7,8,9,9,9,9,8,7,6},
    {0,6,7,8,9,10,9,8,7,6},
    {0,6,7,8,9,9,9,8,7,6},
    {0,6,7,8,8,8,8,8,7,6},
    {0,6,7,7,7,7,7,7,7,6},
    {0,6,6,6,6,6,6,6,6,6},
};
int area[10][10] { //宫
    {0,0,0,0,0,0,0,0,0,0},
    {0,1,1,1,2,2,2,3,3,3},
    {0,1,1,1,2,2,2,3,3,3},
    {0,1,1,1,2,2,2,3,3,3},
    {0,4,4,4,5,5,5,6,6,6},
    {0,4,4,4,5,5,5,6,6,6},
    {0,4,4,4,5,5,5,6,6,6},
    {0,7,7,7,8,8,8,9,9,9},
    {0,7,7,7,8,8,8,9,9,9},
    {0,7,7,7,8,8,8,9,9,9},
};
struct Node {
    int x;
    int y;
} point[N*10];
int vis_x[N][N],vis_y[N][N],vis_g[N][N];
int vis[N*10];
int n,sum;
void dfs(int k) {
    if(k>n) { //当确定的数字个数大于未填入的数字总数时
        int cnt=0;

        for(int i=1; i<=9; i++) //记录分值
            for(int j=1; j<=9; j++)
                cnt+=g[i][j]*grade[i][j];
    }
}

```

```

sum=max(sum,cnt);//取最大分值

return;

}

int w=INF,temp;
for(int i=1; i<=n; i++){ //确定搜索的起点
    if(!vis[i]) {
        int ww=0;
        for(int j=1; j<=9; j++)
            if( !vis_x[point[i].x][j] && !vis_y[point[i].y][j]
&& !vis_g[area[point[i].x][point[i].y]][j] )
                if(++ww==w)
                    break;
        if(ww<w) {
            w=ww;
            temp=i;
        }
    }
}

vis[temp]=1;
int x=point[temp].x;
int y=point[temp].y;

for(int i=1; i<=9; i++){ //枚举每一层可能的状态
    if( !vis_x[x][i] && !vis_y[y][i] && !vis_g[area[x][y]][i] ) { //如果当前数字在整个图
的横向纵向小区域内未出现过
        g[x][y]=i;
        vis_x[x][i]=1;
        vis_y[y][i]=1;
        vis_g[area[x][y]][i]=1;

        dfs(k+1);

        vis_x[x][i]=0;
        vis_y[y][i]=0;
        vis_g[area[x][y]][i]=0;
    }
}

vis[temp]=0;

```

```
}  
int main() {  
    for(int i=1; i<=9; i++){  
        for(int j=1; j<=9; j++) {  
            scanf("%d",&g[i][j]);  
            if(g[i][j]) {  
                vis_x[i][g[i][j]]=1;//标记行  
                vis_y[j][g[i][j]]=1;//标记列  
                vis_g[area[i][j]][g[i][j]]=1;//标记图  
            }  
            else { //如果当前点未填入数字  
                /*记录未填入数字*/  
                n++;  
                point[n].x=i;  
                point[n].y=j;  
            }  
        }  
    }  
  
    sum=-1;  
    dfs(1);  
    cout<<sum<<endl;  
  
    return 0;  
}
```