|  |  |
| --- | --- |
| 试题名称： | ISBN号码 |
| 问题描述： | **问题描述**  　　每一本正式出版的图书都有一个ISBN号码与之对应，ISBN码包括9位数字、1位识别码和3位分隔符，其规定格式如“x-xxx-xxxxx-x”，其中符号“-”是分隔符（键盘上的减号），最后一位是识别码，例如0-670-82162-4就是一个标准的ISBN码。ISBN码的首位数字表示书籍的出版语言，例如0代表英语；第一个分隔符“-”之后的三位数字代表出版社，例如670代表维京出版社；第二个分隔之后的五位数字代表该书在出版社的编号；最后一位为识别码。 　　识别码的计算方法如下： 　　首位数字乘以1加上次位数字乘以2……以此类推，用所得的结果mod 11，所得的余数即为识别码，如果余数为10，则识别码为大写字母X。例如ISBN号码0-670-82162-4中的识别码4是这样得到的：对067082162这9个数字，从左至右，分别乘以1，2，…，9，再求和，即0×1+6×2+……+2×9=158，然后取158 mod 11的结果4作为识别码。 　　编写程序判断输入的ISBN号码中识别码是否正确，如果正确，则仅输出“Right”；如果错误，则输出是正确的ISBN号码。  **输入格式**  　　输入只有一行，是一个字符序列，表示一本书的ISBN号码（保证输入符合ISBN号码的格式要求）。  **输出格式**  　　输出一行，假如输入的ISBN号码的识别码正确，那么输出“Right”，否则，按照规定的格式，输出正确的ISBN号码（包括分隔符“-”）。  **样例输入**  0-670-82162-4  **样例输出**  Right  **样例输入**  0-670-82162-0  **样例输出**  0-670-82162-4 |

|  |  |
| --- | --- |
| 试题名称： | 窗口 |
| 问题描述： | **问题描述**  　　在某图形操作系统中,有 N 个窗口,每个窗口都是一个两边与坐标轴分别平行的矩形区域。窗口的边界上的点也属于该窗口。窗口之间有层次的区别,在多于一个窗口重叠的区域里,只会显示位于顶层的窗口里的内容。 　　当你点击屏幕上一个点的时候,你就选择了处于被点击位置的最顶层窗口,并且这个窗口就会被移到所有窗口的最顶层,而剩余的窗口的层次顺序不变。如果你点击的位置不属于任何窗口,则系统会忽略你这次点击。 　　现在我们希望你写一个程序模拟点击窗口的过程。  **输入格式**  　　输入的第一行有两个正整数,即 N 和 M。(1 ≤ N ≤ 10,1 ≤ M ≤ 10) 　　接下来 N 行按照从最下层到最顶层的顺序给出 N 个窗口的位置。 每行包含四个非负整数 x1, y1, x2, y2,表示该窗口的一对顶点坐标分别为 (x1, y1) 和 (x2, y2)。保证 x1 < x2,y1 2。 　　接下来 M 行每行包含两个非负整数 x, y,表示一次鼠标点击的坐标。 　　题目中涉及到的所有点和矩形的顶点的 x, y 坐标分别不超过 2559 和　　1439。  **输出格式**  　　输出包括 M 行,每一行表示一次鼠标点击的结果。如果该次鼠标点击选择了一个窗口,则输出这个窗口的编号(窗口按照输入中的顺序从 1 编号到 N);如果没有,则输出"IGNORED"(不含双引号)。  **样例输入**  3 4 0 0 4 4 1 1 5 5 2 2 6 6 1 1 0 0 4 4 0 5  **样例输出**  2 1 1 IGNORED  **样例说明**  　　第一次点击的位置同时属于第 1 和第 2 个窗口,但是由于第 2 个窗口在上面,它被选择并且被置于顶层。 　　第二次点击的位置只属于第 1 个窗口,因此该次点击选择了此窗口并将其置于顶层。现在的三个窗口的层次关系与初始状态恰好相反了。 　　第三次点击的位置同时属于三个窗口的范围,但是由于现在第 1 个窗口处于顶层,它被选择。 　　最后点击的 (0, 5) 不属于任何窗口。 |

|  |  |
| --- | --- |
| 试题名称： | 画图 |
| 问题描述： | **问题描述**  　　在一个定义了直角坐标系的纸上，画一个(x1,y1)到(x2,y2)的矩形指将横坐标范围从x1到x2，纵坐标范围从y1到y2之间的区域涂上颜色。 　　下图给出了一个画了两个矩形的例子。第一个矩形是(1,1) 到(4, 4)，用绿色和紫色表示。第二个矩形是(2, 3)到(6, 5)，用蓝色和紫色表示。图中，一共有15个单位的面积被涂上颜色，其中紫色部分被涂了两次，但在计算面积时只计算一次。在实际的涂色过程中，所有的矩形都涂成统一的颜色，图中显示不同颜色仅为说明方便。 http://img.blog.csdn.net/20170221102608763?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvdGlnZXJpc2xhbmQ0NQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center 　　给出所有要画的矩形，请问总共有多少个单位的面积被涂上颜色。  **输入格式**  　　输入的第一行包含一个整数n，表示要画的矩形的个数。 　　接下来n行，每行4个非负整数，分别表示要画的矩形的左下角的横坐标与纵坐标，以及右上角的横坐标与纵坐标。  **输出格式**  　　输出一个整数，表示有多少个单位的面积被涂上颜色。  **样例输入**  2 1 1 4 4 2 3 6 5  **样例输出**  15  **评测用例规模与约定**  　　1<=n<=100，0<=横坐标、纵坐标<=100。 |

|  |  |
| --- | --- |
| 试题名称： | Z字形扫描 |
| 问题描述： | **问题描述**  　　在图像编码的算法中，需要将一个给定的方形矩阵进行Z字形扫描(Zigzag Scan)。给定一个n×n的矩阵，Z字形扫描的过程如下图所示： http://img.blog.csdn.net/20170221103221799?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvdGlnZXJpc2xhbmQ0NQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center 　　对于下面的4×4的矩阵， 　　1 5 3 9 　　3 7 5 6 　　9 4 6 4 　　7 3 1 3 　　对其进行Z字形扫描后得到长度为16的序列： 　　1 5 3 9 7 3 9 5 4 7 3 6 6 4 1 3 　　请实现一个Z字形扫描的程序，给定一个n×n的矩阵，输出对这个矩阵进行Z字形扫描的结果。  **输入格式**  　　输入的第一行包含一个整数n，表示矩阵的大小。 　　输入的第二行到第n+1行每行包含n个正整数，由空格分隔，表示给定的矩阵。  **输出格式**  　　输出一行，包含n×n个整数，由空格分隔，表示输入的矩阵经过Z字形扫描后的结果。  **样例输入**  4 1 5 3 9 3 7 5 6 9 4 6 4 7 3 1 3  **样例输出**  1 5 3 9 7 3 9 5 4 7 3 6 6 4 1 3  **评测用例规模与约定**  　　1≤n≤500，矩阵元素为不超过1000的正整数。 |

|  |  |
| --- | --- |
| 试题名称： | 数字排序 |
| 问题描述： | **问题描述**  　　给定n个整数，请统计出每个整数出现的次数，按出现次数从多到少的顺序输出。  **输入格式**  　　输入的第一行包含一个整数n，表示给定数字的个数。 　　第二行包含n个整数，相邻的整数之间用一个空格分隔，表示所给定的整数。  **输出格式**  　　输出多行，每行包含两个整数，分别表示一个给定的整数和它出现的次数。按出现次数递减的顺序输出。如果两个整数出现的次数一样多，则先输出值较小的，然后输出值较大的。  **样例输入**  12 5 2 3 3 1 3 4 2 5 2 3 5  **样例输出**  3 4 2 3 5 3 1 1 4 1  **评测用例规模与约定**  　　1 ≤ n ≤ 1000，给出的数都是不超过1000的非负整数。 |

|  |  |
| --- | --- |
| 试题名称： | 日期计算 |
| 问题描述： | **问题描述**  　　给定一个年份y和一个整数d，问这一年的第d天是几月几日？ 　　注意闰年的2月有29天。满足下面条件之一的是闰年： 　　1） 年份是4的整数倍，而且不是100的整数倍； 　　2） 年份是400的整数倍。  **输入格式**  　　输入的第一行包含一个整数y，表示年份，年份在1900到2015之间（包含1900和2015）。 　　输入的第二行包含一个整数d，d在1至365之间。  **输出格式**  　　输出两行，每行一个整数，分别表示答案的月份和日期。  **样例输入**  2015 80  **样例输出**  3 21  **样例输入**  2000 40  **样例输出**  2 9 |

|  |  |
| --- | --- |
| 试题名称： | 消除类游戏 |
| 问题描述： | **问题描述**  　　消除类游戏是深受大众欢迎的一种游戏，游戏在一个包含有n行m列的游戏棋盘上进行，棋盘的每一行每一列的方格上放着一个有颜色的棋子，当一行或一列上有连续三个或更多的相同颜色的棋子时，这些棋子都被消除。当有多处可以被消除时，这些地方的棋子将同时被消除。 　　现在给你一个n行m列的棋盘，棋盘中的每一个方格上有一个棋子，请给出经过一次消除后的棋盘。 　　请注意：一个棋子可能在某一行和某一列同时被消除。  **输入格式**  　　输入的第一行包含两个整数n, m，用空格分隔，分别表示棋盘的行数和列数。 　　接下来n行，每行m个整数，用空格分隔，分别表示每一个方格中的棋子的颜色。颜色使用1至9编号。  **输出格式**  　　输出n行，每行m个整数，相邻的整数之间使用一个空格分隔，表示经过一次消除后的棋盘。如果一个方格中的棋子被消除，则对应的方格输出0，否则输出棋子的颜色编号。  **样例输入**  4 5 2 2 3 1 2 3 4 5 1 4 2 3 2 1 3 2 2 2 4 4  **样例输出**  2 2 3 0 2 3 4 5 0 4 2 3 2 0 3 0 0 0 4 4  **样例说明**  　　棋盘中第4列的1和第4行的2可以被消除，其他的方格中的棋子均保留。  **样例输入**  4 5 2 2 3 1 2 3 1 1 1 1 2 3 2 1 3 2 2 3 3 3  **样例输出**  2 2 3 0 2 3 0 0 0 0 2 3 2 0 3 2 2 0 0 0  **样例说明**  　　棋盘中所有的1以及最后一行的3可以被同时消除，其他的方格中的棋子均保留。  **评测用例规模与约定**  　　所有的评测用例满足：1 ≤ n, m ≤ 30。 |

|  |  |
| --- | --- |
| 试题名称： | 俄罗斯方块 |
| 问题描述： | **问题描述**  　　俄罗斯方块是俄罗斯人阿列克谢·帕基特诺夫发明的一款休闲游戏。 　　游戏在一个15行10列的方格图上进行，方格图上的每一个格子可能已经放置了方块，或者没有放置方块。每一轮，都会有一个新的由4个小方块组成的板块从方格图的上方落下，玩家可以操作板块左右移动放到合适的位置，当板块中某一个方块的下边缘与方格图上的方块上边缘重合或者达到下边界时，板块不再移动，如果此时方格图的某一行全放满了方块，则该行被消除并得分。 　　在这个问题中，你需要写一个程序来模拟板块下落，你不需要处理玩家的操作，也不需要处理消行和得分。 　　具体的，给定一个初始的方格图，以及一个板块的形状和它下落的初始位置，你要给出最终的方格图。  **输入格式**  　　输入的前15行包含初始的方格图，每行包含10个数字，相邻的数字用空格分隔。如果一个数字是0，表示对应的方格中没有方块，如果数字是1，则表示初始的时候有方块。输入保证前4行中的数字都是0。 　　输入的第16至第19行包含新加入的板块的形状，每行包含4个数字，组成了板块图案，同样0表示没方块，1表示有方块。输入保证板块的图案中正好包含4个方块，且4个方块是连在一起的（准确的说，4个方块是四连通的，即给定的板块是俄罗斯方块的标准板块）。 　　第20行包含一个1到7之间的整数，表示板块图案最左边开始的时候是在方格图的哪一列中。注意，这里的板块图案指的是16至19行所输入的板块图案，如果板块图案的最左边一列全是0，则它的左边和实际所表示的板块的左边是不一致的（见样例）  **输出格式**  　　输出15行，每行10个数字，相邻的数字之间用一个空格分隔，表示板块下落后的方格图。注意，你不需要处理最终的消行。  **样例输入**  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 3  **样例输出**  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 |

|  |  |
| --- | --- |
| 试题名称： | 火车购票 |
| 问题描述： | **问题描述**  　　请实现一个铁路购票系统的简单座位分配算法，来处理一节车厢的座位分配。 　　假设一节车厢有20排、每一排5个座位。为方便起见，我们用1到100来给所有的座位编号，第一排是1到5号，第二排是6到10号，依次类推，第20排是96到100号。 　　购票时，一个人可能购一张或多张票，最多不超过5张。如果这几张票可以安排在同一排编号相邻的座位，则应该安排在编号最小的相邻座位。否则应该安排在编号最小的几个空座位中（不考虑是否相邻）。 　　假设初始时车票全部未被购买，现在给了一些购票指令，请你处理这些指令。  **输入格式**  　　输入的第一行包含一个整数n，表示购票指令的数量。 　　第二行包含n个整数，每个整数p在1到5之间，表示要购入的票数，相邻的两个数之间使用一个空格分隔。  **输出格式**  　　输出n行，每行对应一条指令的处理结果。 　　对于购票指令p，输出p张车票的编号，按从小到大排序。  **样例输入**  4 2 5 4 2  **样例输出**  1 2 6 7 8 9 10 11 12 13 14 3 4  **样例说明**  　　1) 购2张票，得到座位1、2。 　　2) 购5张票，得到座位6至10。 　　3) 购4张票，得到座位11至14。 　　4) 购2张票，得到座位3、4。  **评测用例规模与约定**  　　对于所有评测用例，1 ≤ n ≤ 100，所有购票数量之和不超过100。 |

|  |  |
| --- | --- |
| 试题名称： | 工资计算 |
| 问题描述： | **问题描述**  　　小明的公司每个月给小明发工资，而小明拿到的工资为交完个人所得税之后的工资。假设他一个月的税前工资（扣除五险一金后、未扣税前的工资）为S元，则他应交的个人所得税按如下公式计算： 　　1） 个人所得税起征点为3500元，若S不超过3500，则不交税，3500元以上的部分才计算个人所得税，令A=S-3500元； 　　2） A中不超过1500元的部分，税率3%； 　　3） A中超过1500元未超过4500元的部分，税率10%； 　　4） A中超过4500元未超过9000元的部分，税率20%； 　　5） A中超过9000元未超过35000元的部分，税率25%； 　　6） A中超过35000元未超过55000元的部分，税率30%； 　　7） A中超过55000元未超过80000元的部分，税率35%； 　　8） A中超过80000元的部分，税率45%； 　　例如，如果小明的税前工资为10000元，则A=10000-3500=6500元，其中不超过1500元部分应缴税1500×3%=45元，超过1500元不超过4500元部分应缴税(4500-1500)×10%=300元，超过4500元部分应缴税(6500-4500)×20%=400元。总共缴税745元，税后所得为9255元。 　　已知小明这个月税后所得为T元，请问他的税前工资S是多少元。  **输入格式**  　　输入的第一行包含一个整数T，表示小明的税后所得。所有评测数据保证小明的税前工资为一个整百的数。  **输出格式**  　　输出一个整数S，表示小明的税前工资。  **样例输入**  9255  **样例输出**  10000  **评测用例规模与约定**  　　对于所有评测用例，1 ≤ T ≤ 100000。 |

|  |  |
| --- | --- |
| 试题名称： | 学生排队 |
| 问题描述： | **问题描述**  　　体育老师小明要将自己班上的学生按顺序排队。他首先让学生按学号从小到大的顺序排成一排，学号小的排在前面，然后进行多次调整。一次调整小明可能让一位同学出队，向前或者向后移动一段距离后再插入队列。 　　例如，下面给出了一组移动的例子，例子中学生的人数为8人。 　　0）初始队列中学生的学号依次为1, 2, 3, 4, 5, 6, 7, 8； 　　1）第一次调整，命令为“3号同学向后移动2”，表示3号同学出队，向后移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 5, 3, 6, 7, 8； 　　2）第二次调整，命令为“8号同学向前移动3”，表示8号同学出队，向前移动3名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 5, 8, 3, 6, 7； 　　3）第三次调整，命令为“3号同学向前移动2”，表示3号同学出队，向前移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 3, 5, 8, 6, 7。 　　小明记录了所有调整的过程，请问，最终从前向后所有学生的学号依次是多少？ 　　请特别注意，上述移动过程中所涉及的号码指的是学号，而不是在队伍中的位置。在向后移动时，移动的距离不超过对应同学后面的人数，如果向后移动的距离正好等于对应同学后面的人数则该同学会移动到队列的最后面。在向前移动时，移动的距离不超过对应同学前面的人数，如果向前移动的距离正好等于对应同学前面的人数则该同学会移动到队列的最前面。  **输入格式**  　　输入的第一行包含一个整数n，表示学生的数量，学生的学号由1到n编号。 　　第二行包含一个整数m，表示调整的次数。 　　接下来m行，每行两个整数p, q，如果q为正，表示学号为p的同学向后移动q，如果q为负，表示学号为p的同学向前移动-q。  **输出格式**  　　输出一行，包含n个整数，相邻两个整数之间由一个空格分隔，表示最终从前向后所有学生的学号。  **样例输入**  8 3 3 2 8 -3 3 -2  **样例输出**  1 2 4 3 5 8 6 7  **评测用例规模与约定**  　　对于所有评测用例，1 ≤ n ≤ 1000，1 ≤ m ≤ 1000，所有移动均合法。 |