|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 黑白图片旋转** |  | | |
| **来源** | 医学部计算概论2班助教--qxx |
| **描述** | |
| 黑白图片可以由一个二维的像素点阵表示，每个像素点是0到255之间的值，分别表示该点的黑白程度。  给你一个m行n列的黑白图片，将其顺时针旋转90度后输出。 | |
| **关于输入** | |
| 第一行输入m和n（100=>m>0,100=>n>0）  之后m行，每行包含n个数，表示图片矩阵的元素，元素值在0到255之间。 | |
| **关于输出** | |
| 输出n行，每行包含m个数，表示将矩阵顺时针旋转90度的结果。 | |
| **例子输入** | |
| 3 4  1 2 3 4  5 6 7 8  9 10 11 12 | |
| **例子输出** | |
| 9 5 1  10 6 2  11 7 3  12 8 4 | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 最大偏离值** |  | | |
| **来源** | 元培-From Whf |
| **描述** | |
| 输入 n 个无符号整数，找出偏离平均值最远的整数。如果有多个不同整数离平均值同时最远，则按从小到大输出这几个不同整数，其间以逗号间隔。其中，个数n 不大于300  例如，有6个数：3，6，7，1，5，4，平均值是 26/6=4.33，与 1 的差值最大。 | |
| **关于输入** | |
| 第1行，输入整数的个数 n；  第2行，输入 n 个整数，整数之间以空格间隔； | |
| **关于输出** | |
| 与平均值差值最大的整数，如果有多个不同整数同时达到最大差值，则从小到大输出相应的整数，其间以逗号间隔。 | |
| **例子输入** | |
| 7  1 2 3 4 5 6 7 | |
| **例子输出** | |
| 1,7 | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 有效的密码** |  | | |
| **来源** | 计算概论医学部2016-陶砺（熊瑞勤） |
| **描述** | |
| 由于北大未名BBS比较知名，容易引来黑客攻击。因此，对每个人的账户而言，密码越复杂，被黑客破解的可能性就越低。通常而言，密码至少由不少于六个字符组成，由以下规则判定密码是否安全：  1. 如果密码长度小于6位，则不安全  2. 如果组成密码的字符只有一类，则不安全  3. 如果组成密码的字符有两类，则为中度安全  4. 如果组成密码的字符有三类或以上，则为安全   通常，把数字、大写字母、小写字母和其它符号归为四类不同的字符。 | |
| **关于输入** | |
| 输入的第一行是一个整数n，表明后面有n组密码输入。随后的n行每行代表一个密码，每个密码的长度均小于50个字符（不含空格）。 | |
| **关于输出** | |
| 针对每一个密码判断并输出它是否安全。对于不安全的密码输出"Not safe"，对于中度安全的密码输出"Medium safe"，对于安全的密码输出"Safe" | |
| **例子输入** | |
| 5  1234  abcdefg  123safe  432\*fe+93  AD7doiuo31 | |
| **例子输出** | |
| Not safe  Not safe  Medium safe  Safe  Safe | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 满足条件的三个四位数** |  | | |
| **来源** | 计算概论A-by-dyf |
| **描述** | |
| A是一个四位数，且是一个完全平方数；B是一个4位数，且每一位的数字都相同；C也是一个4位数的完全平方数。已知C = A - B,请编程求出满足条件的所有A。 | |
| **关于输入** | |
| 此题无输入 | |
| **关于输出** | |
| 满足条件的A输出，如果有多个数满足条件，由大到小，每行输出一个。  此题不允许直接输出不经计算得出结果。 | |
| **例子输入** | |
| 无 | |
| **例子输出** | |
| 无 | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - Unix纪元** |  | | |
| **来源** | 百炼 2725 |
| **描述** | |
| 在著名的unix系统中，使用了一种简洁高效的时间表示方法，即：  将1970年1月1日0点作为“unix纪元”的原点，从1970年1月1日开始经过的秒数存储为一个32位整数   请编写一个程序，帮助把一个unix时间辍，转换成形如"YYYY-mm-dd HH:ii:ss"的格式，其中的字母分别代表   YYYY    4 位数字完整表示的年份  mm      数字表示的月份，有前导零的 2 位数字  dd        月份中的第几天，有前导零的2位数字  HH       小时，24 小时格式，有前导零  ii          有前导零的分钟数  ss         秒数，有前导零 | |
| **关于输入** | |
| 输入数据有若干行，每行包含一个整数t，(0<=t<2^31) | |
| **关于输出** | |
| 对每一行输入数据，输出一行，为形如“YYYY-mm-dd HH:ii:ss”格式的时间 | |
| **例子输入** | |
| 10  1234567890 | |
| **例子输出** | |
| 1970-01-01 00:00:10  2009-02-13 23:31:30 | |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 照亮房间** |  | |
| **描述** |
| 你需要放置一盏灯来照亮房间的每个角落，问这是否能办到？ |
| **关于输入** |
| 输入包含多组数据。  每组数据第一行是正整数n(4<=n<=100)，表示房间轮廓所形成的多边形的顶点个数。  接下来n行，每行表示房间的一个顶点。  顶点按顺时针的顺序给出，房间轮廓多边形的所有边都平行于坐标轴。  输入以n=0表示结束。 |
| **关于输出** |
| 对每组数据输出一行。  如果能有一种放置方法照亮房间的所有地方，输出"Yes"，否则输出"No" |
| **例子输入** |
| 4  0 0  0 1  1 1  1 0  8  0 0  0 2  1 2  1 1  2 1  2 2  3 2  3 0  0 |
| **例子输出** |
| Yes  No |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 布尔表达式** |  | |
| **描述** |
| 输入一个布尔表达式，请你输出它的真假值。  比如：( V | V ) & F & ( F | V )  V表示true，F表示false，&表示与，|表示或，!表示非。  上式的结果是F |
| **关于输入** |
| 输入包含多行，每行一个布尔表达式，表达式中可以有空格，总长度不超过1000 |
| **关于输出** |
| 对每行输入，如果表达式为真，输出"V",否则出来"F" |
| **例子输入** |
| ( V | V ) & F & ( F| V)  !V | V & V & !F & (F | V ) & (!F | F | !V & V)  (F&F|V|!V&!F&!(F|F&V)) |
| **例子输出** |
| F  V  V |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | **题目 - 债务** |  | | |
| **来源** | POJ 3634 |
| **描述** | |
| 有A、B、C三个人，他们之间互相欠钱，即为A欠B的钱，B欠C的钱，而C欠A的钱。现要把他们之间的债务关系清除掉。已知每个人手中持有每种人民币的张数，但每人总共最多有30张，且三人手中所有的钱加起来不超过1000元。现要求交换最少张数的人民币达到清除相互之间债务的目的。人民币只有100元，50元，20元，10元，5元，1元几种面值。 | |
| **关于输入** | |
| 第一行为一个整数t (1 ≤ t ≤ 50),表示下面包含的数据组数。对于每组数据，第一行有3个整数ab, bc, ca ≤ 1000，ab>0表示A欠B的钱（ab<0表示B欠A的钱），bc>0表示B欠C的钱（bc<0时表示C欠B的钱），ca>0表示C欠A的钱（ca<0时表示A欠C的钱）。接下来3行,每行有6个数，依次为A,B,C三人手中的钱的状况，即依次为100，50，20，10，5，1几种面值的数量。 | |
| **关于输出** | |
| 每组数据对应输出一行，即为最少需要交换的钱币张数，如果没有解，输出impossible。 | |
| **例子输入** | |
| 3  10 0 0  0 1 0 0 0 0  0 0 0 3 0 10  0 0 3 0 0 0  -10 -10 -10  0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  -10 10 10  3 0 0 0 2 0  0 2 0 0 0 1  0 0 1 1 0 3 | |
| **例子输出** | |
| 5  0  impossible | |
| **提示** | |
| 搜索 | |