第 1 季：水题

|  |  |
| --- | --- |
| id | 8000010000 |
| description | 计算 A+B |
| input | 输入整数 A 和整数 B 的值，用空格隔开 |
| output | 输出 A+B 的结果 |
| sample\_input | 1 2 |
| sample\_output | 3 |
| id | 8000010001 |
| description | 输入圆半径 r 和圆柱高 h，计算圆周长 l、圆面积 s、圆球表面积 sq、圆球体积 vq 和  圆柱体积 vz |
| input | 输入实型，用空格分隔 |
| output | 按顺序输出圆周长、圆面积、圆球表面积、圆球体积、圆柱体积，保留 2 位小数，结  果每行输出一个 |
| sample\_input | 1.5 3 |
| sample\_output | 9.42  7.07  28.27  14.14  21.21 |
| id | 8000010002 |
| description | 输入某学生的数学、英语和 C 语言课程成绩，输出该学生三门课总成绩和平均成绩 |
| input | 输入为实型，用空格分隔 |
| output | 输出为实型，保留六位小数，结果每行输出一个 |
| sample\_input | 90 82 74 |
| sample\_output | 246.000000  82.000000 |
| id | 8000010003 |
| description | 找出三个数据 A，B，C 中的最大数 |
| input | 输入为整型，用空格分隔 |
| output | 输出为整型 |
| sample\_input | 100 29 712 |
| sample\_output | 712 |
| id | 8000010004 |
| description | 如果一个整数逆序后得到的数值和原数值相同，则称之为幸运数。输入一个整数 N（N  ＜10000）判断是否为幸运数，是的话输出”yes”，否则输出”no”。 |
| input | 输入一个整数 |
| output | 输出为字符串 |
| sample\_input | 1234 |
| sample\_output | no |
| id | 8000010005 |
| description | 企业发放的奖金根据利润提成。利润(I)低于或等于 10 万元时，奖金可提 10%；利润高于 10 万元，低于 20 万元时，低于 10 万元的部分按 10%提成，高于 10 万元的部分， 可提成 7.5%；20 万到 40 万之间时，高于 20 万元的部分，可提成 5%；40 万到 60 万  之间时高于 40 万元的部分，可提成 3%；60 万到 100 万之间时，高于 60 万元的部分， |

|  |  |
| --- | --- |
|  | 可提成 1.5%，高于 100 万元时，超过 100 万元的部分按 1%提成，从键盘输入当月利  润 I，求应发放奖金总数？ |
| input | 输入为实型(单位为万元) |
| output | 输出为实型，保留六位小数(单位为万元) |
| sample\_input | 35 |
| sample\_output | 2.500000 |
| id | 8000010006 |
| description | 西安出租车的收费标准：起步价（2 公里以内，含 2 公里）为 7 元钱，超过 2 公里且在 15 公里以内（含 15 公里）时每公里收费 1.5 元，超过 15 公里时每公里收费 2.1 元，  不足 1 公里按照 1 公里收费。 |
| input | 输入为实型(单位为公里) |
| output | 输出为实型，保留六位小数(单位为元) |
| sample\_input | 9.2 |
| sample\_output | 19.000000 |
| id | 8000010007 |
| description | 输入一个日期(YYYY-MM-DD)输入一个日期，判断是这一年的第几天？ |
| input | 输入为整型，以”-”隔开 |
| output | 输出为整型 |
| sample\_input | 2013-6-17 |
| sample\_output | 168 |
| id | 8000010008 |
| description | 百分制成绩转换为五分制成绩，转换规则为：90～100：A；80～89：B；70～79：C；  60～69：D；60 分以下：E。输入百分制成绩，输出对应的五分制成绩。 |
| input | 输入为整型 |
| output | 输出为字符型 |
| sample\_input | 87 |
| sample\_output | B |
| id | 8000010009 |
| description | 有 4 个圆塔，圆心分别为(2，2)、(-2，2)、(2，-2)、(-2，-2)，圆半径为 1。这 4 个塔的高度为 10m。塔以外无建筑物。请编写程序，输入任一点的坐标，求该点的建筑高度(塔外的高度为零)。 |
| input | 输入为实型，以”,”隔开 |
| output | 输出为整型 |
| sample\_input | 2,2.5 |
| sample\_output | 10 |

第 2 季：循环

|  |  |
| --- | --- |
| id | 8000022000 |
| description | 一个数如果恰好等于它的因子之和，这个数就称为"*完*数"。*例*如，6 的因子为 1、2、  3，*而* 6=1+2+3，因*此* 6 是"*完*数"。请编写程序，找出 1000 之内的*所*有*完*数。 |
| input |  |
| output | 每行按格*式*输出*其*因子：6=1+2+3 |
| sample\_input |  |
| sample\_output | 6=1+2+3  28=1+2+4+7+14  496=1+2+4+8+16+31+62+124+248 |
| id | 8000022001 |
| description |  |
| input | 输入 a 为实型 |
| output | 输出根为实型，保留五位小数。 |
| sample\_input | 2 |
| sample\_output | 1.41421 |
| id | 8000022002 |
| description | 请编写程序，用*二*分*法*求下面*方*程在(*－*10，10)之间的根： |
| input | 输入*区*间数据为实型、用空格隔开输出均。 |
| output | 输出根为实型，保留*两*位小数。 |
| sample\_input | -10 10 |
| sample\_output | 2.00 |
| id | 8000022003 |
| description | VOL 大学有*两*个*乒乓*球*队进*行*比赛*，*各*出 3 *人*。*甲队*为 A、B、C 三*人*，*乙队*为 X、  Y、Z 三*人*，*已抽签决定比赛名*单，有*人向队员打听比赛*的*名*单，A *说他*不和 X *比*，  C *说他*不和 X、Z *比*，请编写程序找出 3 对*赛手*的*名*单。 |
| input |  |
| output | 输出*赛手名*单，一行一对。 |
| sample\_input |  |
| sample\_output | A=Z B=X  C=Y |
| id | 8000022004 |
| description | 编写程序，求任*意两*个整数之间*所*有的*素*数。 |
| input | 输入*两*个整数，用空格间隔。*注意*输入的*两*个整数*谁*大*谁*小是任*意*的。 |
| output | 输出数据*占*一行，用空格间隔。 |
| sample\_input | 100 130 |

|  |  |
| --- | --- |
| sample\_output | 101 103 107 109 113 127 |
| id | 8000022005 |
| description | 有一个分数数*列*：    求出这个数*列前* 20 *项*之和。 |
| input |  |
| output | 输出保留六位小数。 |
| sample\_input |  |
| sample\_output | 32.660261 |
| id | 8000022006 |
| description | *假设银*行整*存*整*取存款*不同期*限*的月*息*利*率*分别为：    利*息*=*本*金\*月*息*利*率*\*12\**存款*年*限*。  *现*在某*人手*中有 2000 元钱，请*通*过计算*选择*一*种存*钱*方案*，*使*得钱*存*入*银*行 20 年后得到的利*息*最*多*（*假定银*行对超过*存款*期*限*的*那*一部分时间不*付*利*息*）。 |
| input |  |
| output | 输出第 1 行为 8 年、5 年、3 年、2 年、1 年的*存款方案*（*各*利*率存款次*数），用空格  隔开。输出第 2 行为最大收*益*。 |
| sample\_input |  |
| sample\_output | 0 4 0 0 0  8841.01 |
| id | 8000022007 |
| description | 编写程序输入一个数，输出*其*整数部分的位数（*例*如输入 123.4 则输出 3，输入*－*0.6  则输出 0）。 |
| input | 输入为实型。 |
| output | 输出为整型。 |
| sample\_input | 123.4 |
| sample\_output | 3 |
| id | 8000022008 |
| description | 编写程序利用下面公*式*计算*π*的*近似*值： |
| input |  |
| output | 输出*π*为实型，保留六位小数。 |
| sample\_input |  |
| sample\_output | 3.141591 |
| id | 8000022009 |
| description | 编写程序求下面*级*数*前* n *项*的和，*其*中 n 从键盘*上*输入。 |

|  |  |
| --- | --- |
|  |  |
| input | 输入 n 为整型。 |
| output | 输出和为实型，保留六位小数。 |
| sample\_input | 5 |
| sample\_output | 0.338462 |
| id | 8000022010 |
| description | 编写程序计算 500～800 *区*间内*素*数的个数 cnt，*并*按*所*求*素*数的值从大到小的顺序，  *再*计算*其*间隔*减*、*加*之和，*即*第 1 个*素*数*－*第 2 个*素*数+第 3 个*素*数*－*第 4 个*素*数+ 第 5 个*素*数*……*的值 sum。 |
| input |  |
| output | 一行内输出 cnt 和 sum，用空格隔开。 |
| sample\_input |  |
| sample\_output | 44 130 |
| id | 8000022011 |
| description | 编写程序*验证*下*列*结*论*：任*何*一个*自然*数 n 的*立方都*等于 n 个*连续奇*数之和。*例*如：    *要*求程序对每个输入的*自然*数计算*并*输出相应的*连续奇*数。 |
| input | 输入*自然*数 n 为整数。 |
| output | 输出 n 个*连续奇*数之和，格*式*如 Sample Output *显示*。 |
| sample\_input | 5 |
| sample\_output | 5\*5\*5=125=21+23+25+27+29 |
| id | 8000022012 |
| description | 编写程序求一个整数的任*意次方*的最后三位数。*即*： |
| input | 输入均为整型，*首先*为 x，*其*后为 a，用空格隔开。 |
| output | 输出为整型。 |
| sample\_input | 13 13 |
| sample\_output | 253 |
| id | 8000022013 |
| description |  |
| input |  |
| output | 输出三行对应的 n 值。 |
| sample\_input |  |
| sample\_output | 6  9  11 |
| id | 8000022014 |
| description | 5 *只猴*子一起*摘*了 1 堆桃子。因为太累了，它们商量*决定*，*先*睡一觉*再*分。过了不知 |

|  |  |
| --- | --- |
|  | *多*久，1 *只猴*子来了。它见别的*猴*子没来，便将这 1 堆桃子平均分成 5 份，结果*多*了  1 个，就将*多*的这个吃了，拿走*其*中的 1 堆。又过了不知*多*久，第 2 *只猴*子来了。它不知道有 1 个同伴*已*经来过，还以为*自*己是第 1 个到的呢。于是将地*上*的桃子堆起来， 平均分成 5 份，发*现*也*多*了 1 个，同样吃了这 1 个，拿走*其*中的 1 堆。第 3 *只*、第 4 *只*、第 5 *只猴*子*都*是这样*……*问这 5 *只猴*子至少*摘*了*多*少个桃子？第 5 个*猴*子走后还剩下*多*少个桃子？ |
| input |  |
| output | 输出 5 *只猴*子至少*摘*了*多*少个桃子，第 5 个*猴*子走后还剩下*多*少个桃子。中间用空格  隔开。 |
| sample\_input |  |
| sample\_output | 3121 1020 |

第 3 季：枚举

|  |  |
| --- | --- |
| id | 8000022015 |
| description | *设* m，n 为一位正整数，含有数字 m 且不能被 m 整除的 n 位整数的个数为 g，这些整  数和为 s。计算 g、s。 |
| input | 输入 m 和 n，均为一位正整数，用空格隔开 |
| output | 输出 g、s，均为整数，用空格隔开 |
| sample\_input | 7 5 |
| sample\_output | 32152 1894711910 |
| id | 8000022016 |
| description | *设* n 为正整数，求解 n *使*不等*式*    成*立*。a,b 为*上*下*限*。*注意*：n 可能是一个值，也可能是一个*区*间 |
| input | 输入 a 和 b，均为整型，用空格隔开 |
| output | 输出 n，为整数；若为*区*间，则输出*区*间*上*下*限*，用空格隔开 |
| sample\_input | 2010 2011 |
| sample\_output | 18611 18621 |
| id | 8000022017 |
| description | 若一个世纪的 100 个年号中不*存*在一个*素*数，则这个世纪称为合数世纪。求第 n 个合  数世纪（公元 0 年起始）。 |
| input | 输入 n，为整数 |
| output | 输出合数世纪起始与结束年份，用空格隔开 |
| sample\_input | 1 |
| sample\_output | 1671800 1671899 |
| id | 8000022018 |
| description | 核反应堆中有α和β*两种*粒子，每秒钟内一个α粒子可以裂变为 3 个β粒子，*而*一个  β粒子可以裂变为 1 个α粒子和 2 个β粒子。若在 t=0 时刻的反应堆中*只*有一个α粒  子，求在 t 秒时反应堆裂变产生的α粒子和β粒子数。 |
| input | 输入 t，为整型 |
| output | 输出在 t 秒时反应堆裂变产生的α粒子和β粒子数，为整型，用空格隔开 |
| sample\_input | 6 |
| sample\_output | 183 546 |

|  |  |
| --- | --- |
| id | 8000022019 |
| description | 输入正整数 n，按从小到大的顺序输出*所*有形如 abcde/fghij=n 的表达*式*，*其*中 a～j 恰  好为数字 0～9 的一个排*列*，2≤n≤79。 |
| input | 输入正整数 n |
| output | 输出形如 abcde/fghij=n 的表达*式*，每行一个 |
| sample\_input | 62 |
| sample\_output | 79546/01283=62  94736/01528=62 |
| id | 8000022020 |
| description | 输入正整数 k，找出*所*有的正整数 x≥y，*使*得 |
| input |  |
| output |  |
| sample\_input | 4 |
| sample\_output | 1/4=1/20+1/5  1/4=1/12+1/6  1/4=1/8+1/8 |
| id | 8000022021 |
| description | 输入 n 个元*素*组成的序*列* S，你需*要*找出一个乘积最大的*连续*子序*列*。如果这个最大  的乘积不是正数，输出-1 表*示*无解。1≤n≤18，-10≤Si≤10。 |
| input | 第 1 行输入 n，整数  第 2 行 n 个元*素*的序*列* S，均为整型，用空格隔开 |
| output | 输出最大乘积，若无解输出-1 |
| sample\_input | 5  2 5 -1 2 -1 |
| sample\_output | 20 |
| id | 8000022022 |
| description | 有一些装有铀和铅的盒子，数量均足够*多*。*要*求把 n（n≤30）个盒子放成一行，但至  少有 3 个铀放在一起，问有*多*少*种方法*？ |
| input | 输入整数 n |
| output | 输出放置*方法*的数目 |
| sample\_input | 5 |
| sample\_output | 8 |
| id | 8000022023 |
| description | 相传韩信才智过*人*，从不直接清点*自*己军*队*的*人*数，*只要*让士兵*先*后以三*人*一排、五  *人*一排、七*人*一排地变换*队*形，*而他*每*次只*看一眼*队*伍的排尾就知道总*人*数了。输入  3 个非负整数 a、b、c，表*示*每*种队*形排尾的*人*数（a＜3、b＜5、c＜7），输出总*人*数的最小值或报告无解。  *已*知总*人*数不小于 10，不超过 100。 |
| input | 输入非负整数 a、b、c，用空格隔开 |
| output | 输出总*人*数，为整数，用空格隔开；若无解输出-1 |

|  |  |
| --- | --- |
| sample\_input | 2 1 6 |
| sample\_output | 41 |
| id | 8000022024 |
| description | 输入*两*个正整数 n＜m＜1000000，输出    保留 5 位小数。*注意*：*本*题有数据陷*进*。 |
| input | 输入*两*个正整数 n、m，用空格隔开 |
| output | 输出计算结果，保留 5 位小数。 |
| sample\_input | 2 4 |
| sample\_output | 0.42361 |
| id | 8000022025 |
| description | 输入正整数 a、b、c，输出 a/b 的小数形*式*，精确到小数点后 c 位。*其*中 a、b≤1000000，  c≤100。*注意*：*本*题有数据陷*进*。 |
| input | 输入正整数 a、b、c，用空格隔开 |
| output | 输出计算结果，小数点后 c 位 |
| sample\_input | 1 6 4 |
| sample\_output | 0.1667 |
| id | 8000022026 |
| description | 用 1、2、3、*…*、9 组成 3 个三位数：abc、def 和ghi，每个数字恰好*使*用了 1 *次*，*要*  求 abc：def：ghi=1：2：3。输出*所*有解。 |
| input |  |
| output | 输出*所*有解。 |
| sample\_input |  |
| sample\_output | 192 384 576  219 438 657  273 546 819  327 654 981 |
| id | 8000022027 |
| description | 计算 1 到N 中数字“1”出*现*的个数，*其*中 1≤N≤1000000000。 |
| input | 输入正整数 N。 |
| output | 输出“1”出*现*的个数 |
| sample\_input | 12 |
| sample\_output | 5 |
| id | 8000022028 |
| description | 青青草原*上*的美羊羊最*近*在网*上*相识了非洲部落的沸羊羊，它们聊得很开心，于是觉得有必*要*见一面。它们很高兴地发*现*它们住在同一条纬度线*上*，于是它们约*定各自*朝西走，直到碰面为止。  可是它们出发之*前*忘记了一件很重*要*的事情，既没有问清楚对*方*的特征，也没有约*定*见面的具体位置。不过羊羊们很乐观，它们觉得*只要*一直朝着某个*方向*走下去，总能碰到对*方*。但是除非这*两只*羊在同一时间走到同一点*上*，不*然*永远*都*不可能碰面。为  了帮助这*两只*乐观的羊，你被*要*求写一个程序来判断这*两只*羊是否能够碰面，会在什 |

|  |  |
| --- | --- |
|  | 么时候碰面。  规*定*纬度线*上*东经 0 度处为原点，由东往西为正*方向*，单位长度 1 米，这样我们就得到了一条*首*尾相接的数轴。*设*美羊羊的出发点坐标是 x，沸羊羊的出发点坐标是 y。美羊羊一小时能走 a 米，沸羊羊一*次*能走 b 米。纬度线总长 L 米。求出它们走了*多*少小时以后才会碰面。 |
| input | 输入一行 5 个整数 x，y，a，b，L，*其*中 x≠y ＜2000000000，0＜a、b＜2000000000，  0＜L＜2100000000。 |
| output | 输出碰面*所*需*要*的跳跃*次*数，如果永远不可能碰面则输出 impossible |
| sample\_input | 1 2 3 4 5 |
| sample\_output | 4 |
| id | 8000022029 |
| description | 输出 7 和 7 的倍数，还有包含 7 的数字*例*如（17，27，37...70，71，72，73...） |
| input | 一个整数 N。(N 不大于 30000) |
| output | 从小到大排*列*的不大于 N 的与 7 有关的数字，用空格隔开 |
| sample\_input | 20 |
| sample\_output | 7 14 17 |

第 4 季：函数

|  |  |
| --- | --- |
| id | 8000030000 |
| description | 用递归*法*将一个长整型数 n 逆序输出。*例*如输入 483，输出 384。n 的位数不确*定*，可  以是有效范围内的任*意*位数。 |
| input | 输入为整数 |
| output | 输出为整数 |
| sample\_input | 4325879 |
| sample\_output | 9785234 |
| id | 8000030001 |
| description | 编写函数计算从 n 个元*素*中*取* m 个元*素*的组合数 C(m,n)。 |
| input | 输入 n 和 m 为整型，用空格隔开。 |
| output | 输出为整型。若无解输出 wrong |
| sample\_input | 9 2 |
| sample\_output | 36 |
| id | 8000030002 |
| description | *已*知    编写函数用梯形*法*计算 f(x)在*区*间[a,b]的积分 |
| input | 输入 a 和 b，a＜b，均为 double 型 |
| output | 输出为 double 型 |
| sample\_input | 0.5 1 |
| sample\_output | 0.321751 |
| id | 8000030003 |
| description | 编写函数计算 |

|  |  |
| --- | --- |
|  | *其*中 x 为 x1,x2,*……*的平均数。请记住：不能*使*用数组。 |
| input | 第 1 行输入 n，为整型  第 2 行输入 x1,x2,*……*，均为 double 型，用空格隔开。 |
| output | 输出 s，double 型，小数点后 6 位。 |
| sample\_input | 5  1 2 3 4 5 |
| sample\_output | 10.000000 |
| id | 8000030004 |
| description | *已*知 ack 函数对于 m≥0 和 n≥0 有*定*义：ack(0,n)=n+1、ack(m,0)=ack(m-1,1)、  ack(m,n)=ack(m-1,ack(m,n-1))。输入 m 和 n，求解 ack 函数。 |
| input | 输入 m 和 n，均为整型，用空格隔开。 |
| output | 输出为整型 |
| sample\_input | 3 2 |
| sample\_output | 29 |
| id | 8000030005 |
| description | 某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是*加*密的。*加*密函数如下：每位数字*都加上* 5，*然*后用除以 10 的余数代替该数字，*再*将第一位和第四  位交换，第*二*位和第三位交换。 |
| input | 输入整型 |
| output | 输出整型 |
| sample\_input | 1998 |
| sample\_output | 3446 |
| id | 8000030006 |
| description | 编写内联函数 inline int xchg(unsigned char n)，计算将 unsigned char 型n 的低四位和高  四位交换后的结果。在主函数中输入数据调用函数输出结果。 |
| input | 输入整型 |
| output | 输出整型 |
| sample\_input | 194 |
| sample\_output | 44 |
| id | 8000030007 |
| description | 编写函数 getbit(n,k)；求出 n 从右边开始的第 k 位。在主函数中输入数据*并*调用该函  数输出结果。 |
| input | 输入整型 n 和 k（1≤k≤16），用空格分隔。 |
| output | 输出整型。 |
| sample\_input | 128 8 |
| sample\_output | 1 |
| id | 8000030008 |
| description | 编写函数实*现* value 左右循环移位（*即*移出的位在另一端填入）。函数原型为 int move(int value,int n)；*其*中 value 为*要*循环移位的数，n 为移位的位数，n 的绝对值不  大于 16，整型为 16 位。如果 n＜0 表*示*左移，n＞0 表*示*右移，n=0 表*示*不移位。在 |

|  |  |
| --- | --- |
|  | 主函数中输入数据*并*调用该函数输出结果。 |
| input | 输入 value 和n，均为整型，用空格隔开 |
| output |  |
| sample\_input | 134744064 -8 |
| sample\_output | 134742024 |
| id | 8000030009 |
| description | 编写函数 fceil(x)，返回大于等于 x 的最小整数，*例*如 fceil(2.8)为 3，fceil(-2.8)为-2。 |
| input | 输入 double 型 x。 |
| output | 输出整型。 |
| sample\_input | 2.8 |
| sample\_output | 3 |
| id | 8000030010 |
| description | 编写函数 getfloor(x)，返回小于等于 x 的最大整数，*例*如 getfloor(2.8)为 2，getfloor(-2.8)  为-3。 |
| input | 输入 double 型 x。 |
| output | 输出整型。 |
| sample\_input | 2.8 |
| sample\_output | 2 |
| id | 8000030011 |
| description | 豆豆今年 3 岁了，*现*在*他已*经能够认识 100 以内的非负整数，*并*且能够*进*行 100 以内的非负整数的*加法*计算。对于大于等于 100 的整数，豆豆仅保留该数的最后*两*位*进*行计算，如果计算结果大于等于 100，*那*么豆豆也仅保留计算结果的最后*两*位。  *例*如, 对于小明来*说*：  （1）1234 和 34 是相等的  （2）35+80=15  给*定*非负整数 A 和 B, 你的任务是代表豆豆计算出 A+B 的值。 |
| input | 输入数据的第 1 行为一个正整数 T，表*示*测试数据的组数，*然*后是 T 组测试数据。每  组测试数据包含*两*个非负整数 A 和 B（A 和B 均在int 型可表*示*的范围内）。 |
| output | 对于每组测试数据, 输出豆豆 A+B 的结果。 |
| sample\_input | 2  35 80  15 1152 |
| sample\_output | 15  67 |
| id | 8000030012 |
| description | POJ 公司的职*员*，最盼望的日子就是每月的 8 号，因为这一天是发工资的日子，养家糊口就靠它了，呵呵。但是对于公司财务部的职*员*来*说*，这一天则是最忙碌的一天。财务部的小明最*近*在考虑一个问题：如果每个*员*工的工资额*都*知道，最少需*要*准备*多*少张*人*民币，才能在给每位*员*工发工资的时候*都*不用*员*工找零呢？  这里*假设员*工的工资*都*是正整数，单位元，*人*民币一共有 100 元、50 元、10 元、5  元、2 元和 1 元六*种*。 |
| input | 输入数据第一行是一个整数 n（n＜100），表*示员*工的*人*数，*然*后是 n 个*员*工的工资。 |
| output | 输出一个整数 x，表*示*至少需*要*准备的*人*民币张数。 |
| sample\_input | 3 |

|  |  |
| --- | --- |
|  | 1 2 3 |
| sample\_output | 4 |
| id | 8000030013 |
| description | A+B 是 POJer 的最爱。这不，今天这个 A+B 希望能给大家带来好运，也希望这个题  目能唤起大家对 ACM 的热爱。  这个题目的 A 和B 不是简单的整数，*而*是*两*个时间，A 和B *都*是由 3 个整数组成， 分别表*示*时、分、秒。*比*如，*假设* A 为 34 45 56，就表*示* A *所*表*示*的时间是 34 小时  45 分钟 56 秒。 |
| input | 输入数据每行有 6 个整数 AH、AM、AS、BH、BM、BS，分别表*示*时间 A 和B *所*对  应的时分秒。题目保*证所*有的数据合*法*。 |
| output | 输出 A+B，每个输出结果也是由时分秒 3 部分组成，同时也*要*满足时间的规则（*即*分  和秒的*取*值范围在 0~59），每个输出*占*一行，*并*且*所*有的部分*都*可以用 32 位整数表*示*。 |
| sample\_input | 1 2 3 4 5 6 |
| sample\_output | 5 7 9 |
| id | 8000030014 |
| description | 古希腊数学家毕达哥拉斯在*自然*数研究中发*现*，220 的*所*有真约数（*即*不是*自*身的约数）之和为：  1+2+4+5+10+11+20+22+44+55+110＝284  *而* 284 的*所*有真约数为1+2+4+71+ 142  *加*起来恰好为 220。*人*们对这样的数感到很惊*奇*，称之为亲和数。一般地讲，如果*两*  个数中任*何*一个数*都*是另一个数的真约数之和，则这*两*个数就是亲和数。*现*在，编写一个程序，判断给*定*的*两*个数是否是亲和数。 |
| input | 输入数据包含*两*个整数 A、B，*其*中 0≤A，B≤600000 |
| output | 如果 A 和 B 是亲和数的话输出 YES，否则输出 NO。 |
| sample\_input | 220 284 |
| sample\_output | YES |

第 5 季：数组

|  |  |  |  |
| --- | --- | --- | --- |
| id | 8000038000 | | |
| description | 有*两*个 N（1≤N≤100）个元*素*的数组 A 和 B，*其*中 A 来*自*输入，将*其*“赋值”给 B  （*即*元*素*一一复制到 B 中），输出 B 数组下标为*奇*数的元*素*。 | | |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，元*素*之间用空格隔开。 | | |
| output | 输出 B 数组指*定*元*素*，元*素*之间用空格隔开。 | | |
| sample\_input | 5  1 2 3 4 5 | | |
| sample\_output | 2 4 | | |
| id |  | 80000 | 38001 |
| description | 输入*两*个 N（1≤N≤100）个元*素*的数组 A 和 B，实*现*它们的“*加法*”（*即*对应元*素*一  一相*加*），*并*“赋值”给同样的 C 数组，输出 C 数组。 | | |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，第 3 行给 B 输入 n 个整型数据，  元*素*之间用空格隔开。 | | |
| output | 输出“*加法*”结果，*即* C 数组，元*素*之间用空格隔开。 | | |
| sample\_input | 5  1 2 3 4 5 | | |

|  |  |
| --- | --- |
|  | 5 4 3 2 1 |
| sample\_output | 6 6 6 6 6 |
| id | 8000038002 |
| description | 输出一个 NxN（1≤N≤100）*二*维数组 A 左*上*角的元*素*（元*素*值不超过 9）。 |
| input | 第 1 行输入整型 n，第 2 行给A 输入 nxn 个整型数据，元*素*之间用空格隔开。 |
| output | 按*要*求输出 A 左*上*角的元*素*，确保行*列*对齐，元*素*之间用空格隔开。 |
| sample\_input | 3  1 2 3 4 5 6 7 8 9 |
| sample\_output | 1 2 3  4 5  7 |
| id | 8000038003 |
| description | 输出一个 NxN（1≤N≤100）*二*维数组 A 右*上*角的元*素*（元*素*值不超过 9）。 |
| input | 第 1 行输入整型 n，第 2 行给A 输入 nxn 个整型数据，元*素*之间用空格隔开。 |
| output | 按*要*求输出 A 左*上*角的元*素*，确保行*列*对齐，元*素*之间用空格隔开。 |
| sample\_input | 3  1 2 3 4 5 6 7 8 9 |
| sample\_output | 1 2 3  5 6  9 |
| id | 8000038004 |
| description | 输出一个 NxN（1≤N≤100）*二*维数组 A 左下角的元*素*（元*素*值不超过 9）。 |
| input | 第 1 行输入整型 n，第 2 行给A 输入 nxn 个整型数据，元*素*之间用空格隔开。 |
| output | 按*要*求输出 A 左*上*角的元*素*，确保行*列*对齐，元*素*之间用空格隔开。 |
| sample\_input | 3  1 2 3 4 5 6 7 8 9 |
| sample\_output | 1  4 5  7 8 9 |
| id | 8000038005 |
| description | 输出一个 NxN（1≤N≤100）*二*维数组 A 右下角的元*素*（元*素*值不超过 9）。 |
| input | 第 1 行输入整型 n，第 2 行给A 输入 nxn 个整型数据，元*素*之间用空格隔开。 |
| output | 按*要*求输出 A 左*上*角的元*素*，确保行*列*对齐，元*素*之间用空格隔开。 |
| sample\_input | 3  1 2 3 4 5 6 7 8 9 |
| sample\_output | 3  5 6  7 8 9 |
| id | 8000038006 |
| description | 计算一个 NxM（1≤N、M≤100）*二*维数组 A *所*有边沿元*素*的和 s1，*所*有内芯元*素*的  和 s2，输出 s1-s2。 |
| input | 第 1 行输入整型 n 和 m，接下来输入 n 行，每行 m 个元*素*给A，数据之间用空格隔开。 |
| output | 输出 s1-s2 为整型。 |
| sample\_input | 3 4 |

|  |  |  |
| --- | --- | --- |
|  | 1 1 1 1  1 2 2 1  1 1 1 1 | |
| sample\_output | 6 | |
| id | 8000038007 | |
| description | 编写一个函数 double avg(int A[],int s,int e)，计算一个 N（1≤N≤100）个元*素*的一维  数组A，从第 s 个（以 0 为开始，下同），到第 e 个元*素*的平均值，*其*中 0≤s＜e＜N。  在主函数输入输出，调用函数 avg 计算。 | |
| input | 第 1 行输入整型 n，第 2 行给 A 输入 n 个整型数据，第 3 行输入 s 和 e，数据之间用  空格隔开。 | |
| output | 输入 double 型，默认小数位。 | |
| sample\_input | 10  1 2 3 4 5 6 7 8 9 10  1 8 | |
| sample\_output | 5.500000 | |
| id | 8000038008 | |
| description | 编写一个函数 void BubbleSort(int A[],int s,int m)，能够从数组 A 第 s 个（以 0 为开始）  元*素*起始，*连续* m 个元*素使*用冒泡*法*降序排序。数组 A 最*多*有N（1≤N≤100）个元  *素*，0≤s＜N，且 s+m≤N。在主函数输入输出，调用函数 BubbleSort 求解。 | |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，第 3 行输入 s 和 m，数据之间用  空格隔开。 | |
| output | 输出排序后的数组 A，用空格隔开。 | |
| sample\_input | 10  1 2 3 4 5 6 7 8 9 10  1 8 | |
| sample\_output | 1 9 8 7 6 5 4 3 2 10 | |
| id | 8000038009 | |
| description | 编写一个函数 void Selecti | onSort(int A[],int s,int m)，能够从数组 A 第 s 个（以 0 为开 |
| 始）元*素*起始，*连续* m 个元*素使*用*选择法*降序排序。数组 A 最*多*有N（1≤N≤100） 个元*素*，0≤s＜N，且 s+m≤N。在主函数输入输出，调用函数 SelectionSort 求解。 | |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，第 3 行输入 s 和 m，数据之间用  空格隔开。 | |
| output | 输出排序后的数组 A，用空格隔开。 | |
| sample\_input | 10  1 2 3 4 5 6 7 8 9 10  1 8 | |
| sample\_output | 1 9 8 7 6 5 4 3 2 10 | |
| id | 8000038010 | |
| description | 编写一个函数 void Inserti | onSort(int A[],int s,int m)，能够从数组 A 第 s 个（以 0 为开始） |
| 元*素*起始，*连续* m 个元*素使*用插入*法*降序排序。数组 A 最*多*有N（1≤N≤100）个元  *素*，0≤s＜N，且 s+m≤N。在主函数输入输出，调用函数 InsertionSort 求解。 | |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，第 3 行输入 s 和 m，数据之间用  空格隔开。 | |
| output | 输出排序后的数组 A，用空格隔开。 | |

|  |  |
| --- | --- |
| sample\_input | 10  1 2 3 4 5 6 7 8 9 10  1 8 |
| sample\_output | 1 9 8 7 6 5 4 3 2 10 |
| id | 8000038011 |
| description | 编写一个函数 void QuickSort(int A[100],int s,int m)，能够从数组 A 第 s 个（以 0 为开  始）元*素*起始，至第 m 个元*素*结束*使*用快速排序降序排序。数组 A 最*多*有 N（1≤N  ≤100）个元*素*，0≤s＜N，且 s+m≤N。在主函数输入输出，调用函数 QuickSort 求  解。 |
| input | 第 1 行输入整型 n，第 2 行给A 输入 n 个整型数据，第 3 行输入 s 和 m，数据之间用  空格隔开。 |
| output | 输出排序后的数组 A，用空格隔开。 |
| sample\_input | 10  1 2 3 4 5 6 7 8 9 10  1 8 |
| sample\_output | 1 9 8 7 6 5 4 3 2 10 |
| id | 8000038012 |
| description | *比*基堡海滩有一个有n 个触*手*的恐怖水母，蟹老板希望雇佣一些海绵宝宝把它杀死（*即*砍掉*所*有触*手*）。*现*在有 m 个海绵宝宝可以雇佣，一个能力值为 x 的海绵宝宝可以砍掉恐怖水母一*只*直径不超过 x 的触*手*，且需*要*支*付* x 个金币。如*何*雇佣海绵宝宝才能杀死水母，*并*且支*付*的金币最少？需*要注意*一个海绵宝宝*只*能砍掉一*只*触*手*，*并*且不  能被雇佣*两次*。 |
| input | 第 1 行为正整数 n 和 m，第 2 行为水母 n *只*触*手*的直径，第 3 行为 m 个海绵宝宝的能  力值，*所*有数据用空格间隔。 |
| output | 输出最少金币数。如果无解，输出 NULL |
| sample\_input | 2 3  5 4  7 8 4 |
| sample\_output | 11 |
| id | 8000038013 |
| description | ft迪*要*出席一个周末表演晚会，*他*在会*上要*表演卡片魔术。*他*有 n（0＜n≤100）张卡片，每张卡片*上都*标明了 1～1000 之间的某个数字，这 n 张卡片*本*来是有序的，可是  ft迪的助*手*不小心把卡片*打*乱了。这可急坏了ft迪，忙令助*手*迅速*通*过一些操作把这些卡片变回有序的，*而*ft迪的助*手*是个思想简单的*人*，*他*能做的操作*只*有一*种*：交换任*意两*张卡片的位置。  *现*在，ft迪想知道助*手*最少交换几*次*可以达到目的，以便尽快*决定*是否替换这个魔术  表演，聪明的你能帮助*他*么？ |
| input | 第 1 行是一个正整数 n，第 2 行 n 个是*打*乱顺序后、每张卡片*上*的数字。 |
| output | 输出*所*需*要*的最少交换*次*数 |
| sample\_input | 5  1 4 7 8 3 |
| sample\_output | 3 |
| id | 8000038014 |
| description | 有一个 N（1≤N≤100）个元*素*的数组 A，按由小到大顺序*存*放。请编写程序，输入  一个数 m，用*二*分查找*法*找出该数在数组中的位置（*即*数组的下标）。如果该数不在 |

|  |  |
| --- | --- |
|  | 数组中，则输出 null。 |
| input | 第 1 行输入 n，第 2 行输入 n 个*已*排好序的数组元*素*，数据之间用空格分隔。第 3 行  输入 m。 |
| output | 若找到输出它数组的下标，否则输出 null |
| sample\_input | 15  1 4 9 13 21 34 55 89 144 233 377 570 671 703 812  34 |
| sample\_output | 5 |

第 6 季：字符串

|  |  |
| --- | --- |
| id | 8000046000 |
| description | Mr. B is a famous music composer. One of his most famous work was his set of preludes. These 24 pieces span the 24 musical keys (there are musically distinct 12 scale notes, and each may use major or minor tonality). The 12 distinct scale notes are:    Five of the notes have two alternate names, as is indicated above with equals sign. Thus, there are 17 possible names of scale notes, but only 12 musically distinct notes. When using one of these as the keynote for a musical key, we can further distinguish between major and minor tonalities. This gives 34 possible keys, of which 24 are musically distinct.  In naming his preludes, Mr. B used all the keys except the following 10, which were named instead by their alternate names:    Write a program that, given the name of a key, give an alternate name if it has one, or report the key name is unique. |
| input | Each test case is described by one line having the format “note tonality”, where “note” is one of the 17 names for the scale notes given above, and “tonality” is either “major” or  “minor” (quotes for clarify). |
| output | For each case output the required answer, following the format of the sample. |
| sample\_input | Ab minor |
| sample\_output | G# minor |
| id | 8000046001 |
| description | Mr. N, Mr. W, Mr.P and Mr. U are now in DongDa, ChangAn, for the 2012 ACM-ICPC Campus Contest. They’ve decided to take a 5 hours training every day before the contest. Also, they plan to start training at 10:00 each day since the contest will do so. The scenery in DongDa is so attractive that Mr. N would always like to take a walk outside for a while after breakfast. However, Mr. N have to go back before training starts, otherwise his teammates will be annoyed. Here is a problem: Mr. N does not have a watch. In order to know the exact time, he has bought a new watch in DongDa, but all the numbers on that watch are represented in Roman Numerals. Mr. N cannot understand such kind of numbers.  Can you translate for him? |
| input | Each test case contains a single line indicating a Roman Numerals that to be translated. All |

|  |  |
| --- | --- |
|  | the numbers can be found on clocks. That is, each number in the input represents an integer between 1 and 12. Roman Numerals are expressed by strings consisting of uppercase ‘I’, ‘V’ and ‘X’. See following the sample for further information.  I,II,III,IV,V,VI,VII,VIII,IX,X,XI,XII  1,2,3,4,5,6,7,8,9,10,11,12 |
| output | For each test case, display a single line containing a decimal number corresponding to the  given Roman Numerals. |
| sample\_input | VIII |
| sample\_output | 8 |
| id | 8000046002 |
| description | 编写实*现*下面字符串操作*要*求的函数。在主函数中输入字符串"[www.nwpu.edu.cn](http://www.nwpu.edu.cn/)"，调  用函数*并*得到结果。   1. 函数 void Left(char src[],int n,char dest[])将字符串 src 左边 n 个字符复制到 dest   中。   1. 函数 void Right(char src[],int n,char dest[])将字符串 src 右边 n 个字符复制到 dest   中。   1. 函数 void Mid(char src[],int loc,int n,char dest[])将字符串 src *自*下标 loc 开始的 n   个字符复制到 dest 中。 |
| input | 第 1 行输入字符串，第 2 行输入 n 和 loc 值，用空格分隔。 |
| output | 第 1 行输出调用函数 Left 的结果，第 2 行输出调用函数 Right 的结果，第 3 行输出调  用函数 Mid 的结果 |
| sample\_input | [www.nwpu.edu.cn](http://www.nwpu.edu.cn/)  5 5 |
| sample\_output | www.n du.cn  wpu.e |
| id | 8000046003 |
| description | As is known to all,if you throw a coin up and let it droped on the desk there are usually three results. Yes,just believe what I say ~it can be the right side or the other side or standing on the desk, If you don’t believe this,just try In the past there were some famous mathematicians working on this .They repeat the throwing job once again. But jacmy is a lazy boy.He is busy with dating or playing games.He have no time to throw a single coin for 100000 times. Here comes his idea,He just go bank and exchange thousands of dollars into coins and then throw then on the desk only once. The only job left for him is to count the number of coins with three conditions.  He will show you the coins on the desk to you one by one. Please tell him the possiblility of the coin on the right side as a fractional number if the possiblity between the result and 0.5 is no larger than 0.003. BE CAREFUL that even 1/2，50/100,33/66 are equal only 1/2 is accepted ! if the difference between the result and 0.5 is larger than 0.003,Please tell him  “Fail”.Or if you see one coin standing on the desk,just say “WA” any way. |
| input | The input is the result with N litters(1＜N＜1000).The letter are “U”,”D”,or “S”,”U”  means the coin is on the right side. “D” means the coin is on the other side .”S” means standing on the desk. |
| output | If test successeded,just output the possibility of the coin on the right side.If the test failed  please output “Fail”,If there is one or more”S”,please output “WA” |

|  |  |
| --- | --- |
| sample\_input | UUUDDD |
| sample\_output | 1/2 |
| id | 8000046004 |
| description | 编写一个程序实*现*将字符串中的*所*有”you”替换成”we” |
| input | 输入数据是一个字符串，长度不超过 1000 |
| output | 对于输入的每一行，输出替换后的字符串 |
| sample\_input | you are what you do |
| sample\_output | we are what we do |
| id | 8000046005 |
| description | 请编写一个函数 int stringcompare(char S1[],char S2[])，将*两*个字符串 S1 和 S2 *比*较。如果 S1＞S2，输出一个正数；S1=S2，输出 0；S1＜S2，输出一个负数。在主函数*两*个字符串用 gets 函数读入。输出的正数或负数的绝对值应是相*比*较的*两*个字符串相对应字符的ASCII 码的差值。*例*如，’A’与’C’相*比*，由于’A’＜’C’，应输出负数，由于’A’与’C’的码差值为 2，因*此*应输出“*－*2”。同理：“And”和“Aid:”  *比*较，根据第 2 个字符*比*较结果，’n’*比*’i’大 5，因*此*应输出“5”。 |
| input | 第 1 行输入 S1，第 2 行输入 S2。 |
| output | 输出为整型。 |
| sample\_input | And  Aid |
| sample\_output | 5 |
| id | 8000046006 |
| description | 输入任*意*一个字符串（包含 n 个字符，1≤n≤100），编写函数，将*此*字符串中从第 m  个字符开始（以 0 起始）的全部字符复制成为另一个字符串*并*输出（m＜n）。 |
| input | 第 1 行输入字符串，第 2 行输入整数 m。 |
| output | 输出为字符串。 |
| sample\_input | 112233445566778899  7 |
| sample\_output | 45566778899 |
| id | 8000046007 |
| description | 在主函数中输入 10 个等长的字符串（每个字符串最*多* 10 个字符），用另一个函数对  它们*进*行由小到大排序，*然*后在主函数中输出这 10 个*已*排好序的字符串。 |
| input | 输入 10 个等长的字符串，用空格分隔。 |
| output | 输出排序后的 10 个字符串，用空格分隔。 |
| sample\_input | she its can ibm bbc NBA nhk BOY jxf eat |
| sample\_output | BOY NBA bbc can eat ibm its jxf nhk she |
| id | 8000046008 |
| description | 输入一个字符串，内有数字和非数字字符。*例*如：a123x456 17960 302tab5876。将*其*  中*连续*的数字作为一个整数，依*次存*放到一维数组 a 中，*例*如 123 放在 a[0]，456 放在 a[1]*……*统计共有*多*少个整数，*并*输出这些数。 |
| input | 输入一个字符串（允许空格）。 |
| output | 第 1 行输出个数，第 2 行输出*多*个整数，用空格分隔。 |
| sample\_input | a123X456 7689?89njmk32lnk123 |
| sample\_output | 6  123 456 7689 89 32 123 |

|  |  |
| --- | --- |
| id | 8000046009 |
| description | 请编写程序，对键盘输入的字符串*进*行逆序，逆序后的字符串仍*然*保留在原来字符数  组中，最后输出。（不得调用任*何*字符串处理函数），*例*如：输入 输出。 |
| input | 输入字符串时，输入以等号（=）结束（*注意*不是回车） |
| output | 输出为字符串。 |
| sample\_input | hello world=  dlrow olleh |
| sample\_output | dlrow olleh |
| id | 8000046010 |
| description | 编写一个函数 void stringmerge(char S1[],char S2[])，对键盘输入的*两*个字符串 S1 和 S2  *进*行*连*接，结果送回到 S1 中。在主函数输入输出，调用这个函数合*并*字符串。 |
| input | 用 2 行分别输入 2 个字符串。 |
| output | 输出为字符串。 |
| sample\_input | hello  world |
| sample\_output | helloworld |
| id | 8000046011 |
| description | 编制函数 void deletechar(char S[],char c)，*其*功能是删除一个字符串 S 中指*定*的字符 c。  *要*求原始字符串在主函数中输入，处理后的字符串在主函数中输出。 |
| input | 第 1 行输入字符串  第 2 行输入删除字符 |
| output | 输出为字符串。 |
| sample\_input | Nikon: wfeel thew awir arouwnd the wuniverse  w |
| sample\_output | Nikon: feel the air around the universe |
| id | 8000046012 |
| description | 有一篇文章，共有 3 行文字，每行最*多*有 80 个字符。编写程序分别统计出文章中英  文大写字母、小写字母、数字、空格及*其他*字符的个数。 |
| input | 输入 3 行字符串 |
| output | 输出英文大写字母、小写字母、数字、空格、*其他*字符的个数，用空格分隔。 |
| sample\_input | Nikon at the frontiers of science.  Flash(Adobe Flash Media Rights Management Server) 21.03,-0.87,-3.97% |
| sample\_output | 8 62 10 10 11 |
| id | 8000046013 |
| description | 编写程序以字符串为单位，以空格或标点符号（字符串中仅含英文逗号','或小数点'.'  作为标点符号）作为分隔符，对字符串中*所*有单词*进*行倒排，*然*后把*已*处理的字符串  （应不含标点符号）*打*印出来。 |
| input | 输入一个字符串（包含大小写字母、空格、逗号或小数点） |
| output | 输出处理后的字符串。 |
| sample\_input | I am a student. I like study. |
| sample\_output | study like I student a am I |
| id | 8000046014 |
| description | 编写程序对字符串按下面给*定*的条件*进*行排序，排序后的结果仍按行重新*存*入字符串 |

|  |  |
| --- | --- |
|  | 中*并打*印出来。  条件：从字符串中间一分为*二*，左边部分按字符的 ASCII 值降序排序，右边部分按字符的ASCII 值升序排序；排序后，左边部分与右边部分*进*行交换。如果原字符串长度为*奇*数，则最中间的字符不参*加*排序，字符仍放在原位置*上*。 |
| input | 输入一行字符串。 |
| output | 输出结果字符串。 |
| sample\_input | abcd9876 |
| sample\_output | 6789dcba |

第 7 季：复杂数据

|  |  |
| --- | --- |
| id | 8000056000 |
| description | 编写一个 C 程序，实*现两*个分数的*加减法*。 |
| input | 每行数据是一个字符串，格*式*是“a/b+c/d”或“a/b-c/d”。*其*中 a, b, c, d 是一个 0-9 的  整数。b、d 不为 0。输入数据保*证*合*法*。 |
| output | 对于输入数据的每一行输出*两*个分数的运算结果。*注意*结果应符合书写习惯，没有*多*  余的符号、分子、分母，*并*且化简至最简分数。*例*如：“1/4-1/2”的结果是-1/4，“1/3-1/3”  的结果是 0。 |
| sample\_input | 1/8+3/8 |
| sample\_output | 1/2 |
| id | 8000056001 |
| description | ACM *队*的POJer 小 C 经常抱怨：“C 语言中格*式*输出中有十六、十、八*进*制输出，却没有*二进*制输出，哎，*谁*能帮我写一个程序实*现*输入一个十*进*制数 n，输出它的*二进*制数呀？”  你能帮帮*他*吗？ |
| input | 输入数据 n（0≤n≤1000010000） |
| output | 输出对应一个十*进*制数 n 的*二进*制数，*注意*：输出的*二进*制去掉任*何*一个*多*余的 0。 |
| sample\_input | 9 |
| sample\_output | 1001 |
| id | 8000056002 |
| description | *使*用字符指针编写程序，输入一个长度不超过 80 的字符串 a，在字符串 a 的 i 处（0  ＜i＜80）位置插入字符 x，输出插入后的字符串 a。*例*如：输入 nw world 在 1 处插入  e 输出 new world。 |
| input | 第 1 行输入字符串，第 2 行输入字符 x 和 i 值，用空格分隔。 |
| output | 输出插入字符后的字符串。 |
| sample\_input | nw world  e 1 |
| sample\_output | new world |
| id | 8000056003 |
| description | 编写函数 void strencode(char \*s)；函数的功能是将字符串中的大写字母*加* 3，小写字  母*减* 3。在主函数中输入字符串，调用函数后输出结果字符串。 |
| input | 输入一行字符串。 |
| output | 输出编码操作后的字符串。 |
| sample\_input | ABCDEF |
| sample\_output | DEFGHI |
| id | 8000056004 |

|  |  |
| --- | --- |
| description | 编写函数 void fun(char \*s,char \*t)，将参数 s *所*指字符串中除了下标为*奇*数，同时 ASCII值也为*奇*数的字符之外，*其*余的*所*有字符*都*删除，串中剩余字符*所*形成的一个新串放在参数 t *所*指的数组*并*返回给调用函数（*例*如：输入 0123456789，结果为 13579）。从  主函数中输入*并*调用函数得到结果。 |
| input | 输入一行字符串。 |
| output | 输出重组后的字符串。 |
| sample\_input | 0123456789 |
| sample\_output | 13579 |
| id | 8000056005 |
| description | 编写函数 char\* search(char \*cpsource,char ch)，该函数在一个字符串中找到可能的最长的子字符串，该字符串是由同一字符组成的。从主函数中输入"aabbcccddddeeeeeffffff"  和'e'，调用函数得到结果。 |
| input | 第 1 行输入字符串，第 2 行输入字符 ch。 |
| output | 输出子字符串 |
| sample\_input | aabbcccddddeeeeeffffff  e |
| sample\_output | eeeee |
| id | 8000056006 |
| description | 编写函数 void replace(char \*str,char \*fstr,char \*rstr)，将 str *所*指字符串中凡是与 fstr 字符串相同的字符替换成 rstr（rstr 与 fstr 的字符长度不一*定*相同）。从主函数中输入原始字符串"iffordowhileelsewhilebreak"、查找字符串"while"和替换字符串"struct"，调用  函数得到结果。 |
| input | 第 1 行输入*要*替换的字符串 str，第 2 行输入查找字符串 fstr，第 3 行输入替换字符串  rstr。 |
| output | 输出替换后的字符串 str。 |
| sample\_input | iffordowhileelsewhilebreak while  struct |
| sample\_output | iffordostructelsestructbreak |
| id | 8000056007 |
| description | *设*有学生信*息*如下：学号（长整型）、姓*名*（字符串）、年龄（整型），英语、数学、语文、政治、物理、化学、计算机成绩（均为实型），总分（实型）、平均分（实型）。编写程序输入 10 个学生信*息*，计算每个学生的总分、平均分，*然*后输出总分最高的  学生记录。 |
| input | 输入 10 行记录，用空格分隔。 |
| output | 输出总分最高的学生记录，成绩保留 2 位小数，用空格分隔。 |
| sample\_input | 101 zhang 18 76 83 74 80 87 95 78  102 chen 19 86 73 83 72 66 98 60  103 yuan 20 83 99 82 74 85 67 98  104 zhou 21 96 64 95 94 83 71 60  105 huang 22 84 94 83 98 77 95 95  106 jiang 23 70 63 75 71 90 67 84  107 ding 24 63 67 67 68 64 91 99  108 qin 25 68 64 67 78 66 88 77  109 deng 26 67 70 88 66 85 92 88 |

|  |  |
| --- | --- |
|  | 110 gao 27 73 95 84 77 79 82 64 |
| sample\_output | 105 huang 22 84.00 94.00 83.00 98.00 77.00 95.00 95.00 |
| id | 8000056008 |
| description | *定*义下面结构表*示*复数：    编写四个函数分别实*现*复数的和、差、积、商计算，在主函数中输入数据*并*调用这些  函数得到复数运算结果。 |
| input | 第 1 行输入第 1 个复数的实部和虚部，第 2 行输入第 2 个复数的实部和虚部 |
| output | 输出数据格*式*为%+.2lf，输出格*式*如下结果*显示*。 |
| sample\_input | -1 5  4 3 |
| sample\_output | (-1.00+5.00i)+(4.00+3.00i)=(3.00+8.00i)  (-1.00+5.00i)-(4.00+3.00i)=(-5.00+2.00i)  (-1.00+5.00i)\*(4.00+3.00i)=(-19.00+17.00i)  (-1.00+5.00i)/(4.00+3.00i)=(0.44+0.92i) |
| id | 8000056009 |
| description | 互联网*上*最流行三*种*图片格*式*为：JPEG、GIF、PNG。这三个格*式*的文件，包含特殊图像数据。如果是 JPEG 文件，*其*文件偏移第 6 个字节处（以 0 起始）为 JFIF；如果是 GIF 文件，*其*文件偏移第 0 个字节处（以 0 起始）为 GIF89a；如果是 PNG 文件， *其*文件偏移第 1 个字节处（以 0 起始）为 PNG。*现*在编写程序，判断一个数据文件  DATA5609.DAT 是什么格*式*？ |
| input | *自*行从互联网*上*下载一个图像文件，更*名*为 DATA5609.DAT 来测试。*例*如是一个 JPEG  文件。 |
| output | 根据格*式*分别 JPEG、GIF、PNG 字符串。若不是这三*种*格*式*，输出 NULL |
| sample\_input |  |
| sample\_output | JPEG |
| id | 8000056010 |
| description | 编写程序统计 DATA5610.TXT 文件中出*现*"if","while","for"单词的*次*数。 |
| input | 用记事*本自*行建*立*一个 DATA5610.TXT 文件，*存*放在源程序在相同的文件夹中，用来  测试*自*己的程序。  提交程序时无需提交 DATA5610.TXT 文件。 |
| output | 分别输出"if","while","for"单词的*次*数，用空格分隔。*例*如某个文件中包含 2 个 if,3 个  while,4 个 for 单词。 |
| sample\_input |  |
| sample\_output | 2 3 4 |
| id | 8000056011 |
| description | 24 位BMP 位图文件 DATA5611.BMP 由文件头、位图信*息*头和图形数据三部分组成。文件头主*要*包含文件大小、文件类型、图像数据偏离文件头的长度等信*息*；位图信*息*头包含图像尺寸信*息*、图像像*素*字节数、是否压缩、图像*所*用颜色数等信*息*  （<http://www.wotsit.org/>网站*上*有*各种*图形图像、音频视频、文档文件格*式*的*说*明）。  根据位图文件格*式定*义文件头、位图信*息*头结构体类型，从位图文件读*取*结构体数据从*而*得到位图文件信*息*。编写程序求 DATA5611.BMP 位图的长和宽。 |

|  |  |
| --- | --- |
| input | 用绘图工具*自*行创建一个 20x20 的 DATA5611.BMP 位图文件，*存*放在源程序在相同的文件夹中，用来测试*自*己的程序。  提交程序时无需提交 DATA5611.BMP 文件。 |
| output | 输出位图的长和宽，用空格分隔。 |
| sample\_input |  |
| sample\_output | 20 20 |
| id | 8000056012 |
| description | 编写程序给源程序文件 DATA5612.CPP *加上*行号后*存*储到另外一个文*本*文件  DATA5612.TXT 中。 |
| input | 用记事*本自*行建*立*一个 DATA5612.CPP 文件，输入一些源程序测试，*存*放在与程序相同的文件夹中。  提交程序时无需提交 DATA5612.CPP 文件。 |
| output | 程序运行结果是产生文件，无输出*显示*。下面是文件测试结果：    行号为 4 位，*其*后为一个空格，*然*后是程序行。 |
| sample\_input |  |
| sample\_output |  |
| id | 8000056013 |
| description | *已*有*两*个文*本*文件（DATA5613.TXT 和 DATA5613.CPP），请编写程序从这*两*个文件  中读出*各*行字符，逐个*比*较这*两*个文件中相应的行和*列上*的字符，如果遇到互不相同的字符，输出它是第几行第几*列*的字符。 |
| input | 用记事*本自*行建*立* DATA5613.TXT 和 DATA5613.CPP 文件，输入一些数据测试，*存*  放在与程序相同的文件夹中。  提交程序时无需提交 DATA5613.TXT 和 DATA5613.CPP 文件。  下面是文件测试数据： |

|  |  |
| --- | --- |
|  |  |
| output | 输出互不相同字符的行和*列*值，一行一个字符，用空格分隔。 |
| sample\_input |  |
| sample\_output | 1 7  3 6  7 9 |
| id | 8000056014 |
| description | 编写程序*完*成：①输入若干学生的数据：学号(int)，姓*名*(char [12])，成绩(int)，*并存*  储在文件 DATA5614.DB 中；②从文件中*再*读出学生数据，*打*印成绩最好的学生的相  关信*息*。 |
| input | 第 1 行输入学生*人*数 n，第 2 行输入 n 个*人*数的学号、姓*名*、成绩，用空格分隔。 |
| output | 输出成绩最好的学生的记录，用空格分隔。 |
| sample\_input | 5  101 zhang 78 106 wang 88 107 chen 78 103 wang 88 104 chen 98 |
| sample\_output | 104 chen 98 |

第 8 季：数据结构

|  |  |
| --- | --- |
| id | 8000070000 |
| description | In your job at DongDa Management Inc.(yes, it’s run by a bunch of clowns), you have just finished writing a program whose output is a list of names in nondescending order by length (so that each name is at least as long as the one preceding it). However, your boss does not like the way the output looks, and instead wants the output to appear more symmetric, with the shorter strings at the top and bottom and the longer strings in the middle. His rule is that each pair of names belongs on opposite ends of the list, and the first name in the pair is always in the top part of the list. In the example set below, Bo and Pat  are the first pair, Jean and Kevin the second pair, etc. |
| input | The input consists of one set of strings, Each set starts with a line containing an integer, n, which is the number of strings in the set, followed by n strings, NOT SORTED. None of the strings contain spaces. There is at least one and no more than 15 strings per set. Each string  is at most 25 characters long. |
| output | For each input set ,the output set as shown in the sample output.,If length of two strings is  equal,arrange them as the original order. |

|  |  |
| --- | --- |
| sample\_input | 7  Bo Pat Jean Kevin Claude William Marybeth |
| sample\_output | Bo Jean Claude Marybeth William Kevin Pat |
| id | 8000070001 |
| description | 编写程序建*立*一个链表，每个结点包括：学号、姓*名*、年龄，输入一个学号，如果链  表中的结点包括该学号，则输出该结点内容后，*并*将*其*结点删去。 |
| input | *连续*输入学号 int no、姓*名* char name[12]、年龄 int age，结束学号输入 0。*再*输入查找  节点学号。 |
| output | 输出第 1 行为原链表，第 2 行为删除节点后的链表。 |
| sample\_input | 1. zhang 18 2. wang 21 3. zhou 19 4. chen 20 5. huang 20   0  103 |
| sample\_output | 101 102 103 104 105  101 102 104 105 |
| id | 8000070002 |
| description | 编写一个程序实*现* A+B。不过与我们以*前完*成过的 A+B 不一样，A 和B *两*个数的位数有*近* 100 位。*注意*：需*要*处理正负数的情形，但不考虑小数。  提*示*：C 或C++没有直接表*示* 100 位的数据类型，我们得*自*己*先*构造一个合适的类型，  接下来还需*要*一些巧妙的*方法*处理 100 位数的输入和输出，因为标准输入输出同样没  有这样类型的处理功能。 |
| input | 分别用*两*行输入不超过 100 位的 A 和B（没有小数点，允许正负号）。 |
| output | 输出 A+B 的结果。 |
| sample\_input | 3333333333333333333333333333333333  -2222222222222222222222222222222222 |
| sample\_output | 1111111111111111111111111111111111 |
| id | 8000070003 |
| description | 编写一个程序实*现* A-B，A 和 B *两*个数的位数有*近* 100 位。*注意*：需*要*处理正负数的  情形，但不考虑小数。  提*示*：C 或C++没有直接表*示* 100 位的数据类型，我们得*自*己*先*构造一个合适的类型， 接下来还需*要*一些巧妙的*方法*处理 100 位数的输入和输出，因为标准输入输出同样没有这样类型的处理功能。 |
| input | 分别用*两*行输入不超过 100 位的 A 和B（没有小数点，允许正负号）。 |
| output | 输出 A-B 的结果。 |
| sample\_input | 3333333333333333333333333333333333  -2222222222222222222222222222222222 |
| sample\_output | 5555555555555555555555555555555555 |
| id | 8000070004 |
| description | 编写一个程序实*现* A\*B，A 和 B *两*个数的位数有*近* 100 位。*注意*：需*要*处理正负数的  情形，但不考虑小数。  提*示*：C 或C++没有直接表*示* 100 位的数据类型，我们得*自*己*先*构造一个合适的类型，  接下来还需*要*一些巧妙的*方法*处理 100 位数的输入和输出，因为标准输入输出同样没 |

|  |  |
| --- | --- |
|  | 有这样类型的处理功能。 |
| input | 分别用*两*行输入不超过 100 位的 A 和B（没有小数点，允许正负号）。 |
| output | 输出 A\*B 的结果。 |
| sample\_input | 3333333333333  2222222222222 |
| sample\_output | 7407407407405925925925926 |
| id | 8000070005 |
| description | 编写一个程序实*现* A/B，A 和 B *两*个数的位数有*近* 100 位。*注意*：需*要*处理正负数的  情形，但不考虑小数。  提*示*：C 或C++没有直接表*示* 100 位的数据类型，我们得*自*己*先*构造一个合适的类型， 接下来还需*要*一些巧妙的*方法*处理 100 位数的输入和输出，因为标准输入输出同样没有这样类型的处理功能。 |
| input | 分别用*两*行输入不超过 100 位的 A 和B（没有小数点，允许正负号）。 |
| output | 输出 A/B 的结果。 |
| sample\_input | 666666666666666666666666666666666  33 |
| sample\_output | 20202020202020202020202020202020 |
| id | 8000070006 |
| title | 创建与遍历职工链表 |
| description | 建*立*一个链表，每个结点包括的成*员*为：职工号。用一个 creat 函数来建*立*链表，用  list 函数来输出数据。 |
| input | 第 1 行输入 n，第 2 行输入 n 个职工号，用空格分隔。 |
| output | 从头节点开始输出每个结点职工号，用空格分隔。 |
| sample\_input | 7  101 102 103 104 105 106 107 |
| sample\_output | 101 102 103 104 105 106 107 |
| id | 8000070007 |
| description | 建*立*一个链表，每个结点包括的成*员*为：职工号。用一个 creat 函数来建*立*链表，用  list 函数来输出数据。*现*在新增*加*一个职工的数据，按职工号的顺序插入链表。写一函数 insert 来插入新结点。 |
| input | 第 1 行输入 n，第 2 行输入 n 个职工号（由小到大）用来创建链表，用空格分隔。第  3 行输入插入元*素*值。 |
| output | 输出插入后的链表。 |
| sample\_input | 7  101 102 103 104 105 107 108  106 |
| sample\_output | 101 102 103 104 105 106 107 108 |
| id | 8000070008 |
| description | 建*立*一个链表，每个结点数据包括：年龄。n *次*输入年龄，如果链表中的结点*所*包含  的年龄等于*此*年龄，则将*此*结点删去。否则增*加*一个新结点。 |
| input | 第 1 行输入 n，第 2 行输入 n 个年龄值，用空格分隔。 |
| output | 输出整个链表数据，用空格分隔。 |
| sample\_input | 10  12 16 16 17 19 20 22 27 34 22 |

|  |  |
| --- | --- |
| sample\_output | 12 17 19 20 27 34 |
| id | 8000070009 |
| description | 对一个实数 R（0.0＜R＜99.999），编写程序精确计算 R 的 n *次方*，*其*中 n 是整数且 0  ≤n≤25。 |
| input | 输入 R 和n，用空格分隔。 |
| output | 输出 R 的n *次方*精确值，输出需*要*去掉无用的 0。如果输出结果是整数，不*要*输出小  数点。 |
| sample\_input | 95.123 12 |
| sample\_output | 548815620517731830194541.899025343415715973535967221869852721 |
| id | 8000070010 |
| description | RLE（Run Length Encoding 行程编码）算*法*是一个简单高效的无损数据压缩算*法*，*其*基*本*思路是把数据看成一个线性序*列*，*而*这些数据序*列*组织*方式*分成*两种*情况：一*种*是*连续*的重复数据块，另一*种*是*连续*的不重复数据块。对于*连续*的重复数据快采用的压缩策略是用一个字节（我们称之为数据重数属性）表*示*数据块重复的*次*数，*然*后在这个数据重数属性字节后面*存*储对应的数据字节*本*身，*例*如某一个文件中有如下的数据序*列* AAAAA，在未压缩之*前占*用 5 个字节，*而*如果*使*用了压缩之后就变成了 5A， *只占*用*两*个字节，对于*连续*不重复的数据序*列*，表*示方法*和*连续*的重复数据块序*列*的表*示方法*一样，*只*不过*前*面的数据重数属性字节的内容为 1。一般的这里的数据块*取*一个字节，这篇文章中数据块*都*默认为一个字节。  具体来讲，字符串的编码规则如下：  在字符串中，2～9 个相同的字符组成的子字符串用 2 个字符来编码表*示*。第 1 个字符是这一字符串的长度，为 2～9。第 2 个字符是相同字符的值。如果一个字符串*存*在相同字符且*多*于 9 个的子串，就*先*对*前* 9 个字符*进*行编码，*然*后对*其*余相同字符组成的子串采用相同*方法进*行编码。*例*如 AAAAAABCCCC 编码为 6A1B14C。  在字符串中，如果*存*在某个子串，*其*中没有一个字符*连续*重复出*现*，就表*示*为以字符  1 开始，后面跟着这一子串，*再*以字符结束。如果在字符串中*存*在*只*有 1 个字符 1 出  *现*的子串，则以*两*个字符 1 作为输出，*例*如 12344 编码为 11123124。 |
| input | 输入一个字符串。 |
| output | 输出 RLE 编码后的字符串。 |
| sample\_input | AAAAAABCCCC12344 |
| sample\_output | 6A1B14C11123124 |
| id | 8000070011 |
| description | 有一个排序算*法* QuickSort。这个算*法*是将 n 个不同的整数由小到大*进*行排序，算*法*的操作是在需*要*的时候将相邻的 2 个数交换。*例*如，对于输入序*列* 9 1 0 5 4，QuickSort  产生 0 1 4 5 9 的结果。  你的任务是算出 QuickSort 最少需*要*用到*多*少*次*交换操作，才能对输入的序*列*由小到大排序。 |
| input | 第 1 行输入序*列*的个数 n，第 2 行输入 n 个序*列*元*素*。 |
| output | 输出*进*行排序*所*做的交换操作的最少*次*数。 |
| sample\_input | 5  9 1 0 5 4 |
| sample\_output | 6 |
| hint |  |
| id | 8000070012 |
| description | 在PushPop 城中有一个著*名*的火车站，车站铁路如图*所示*。 |

|  |  |
| --- | --- |
|  | 每辆火车*都*从 A *方向*驶入车站，*再*从 B *方向*驶出车站，同时它的车厢可以*进*行某*种*形*式*的重新组合。*假设*从 A *方向*驶来的火车有 N 节车厢（N≤1000），分别按顺序编号为 1，2，*…*，N。负责车厢调度的工作*人员*需*要*知道能否*使*它以 A1，A2，*…*，An 的顺序从 B *方向*驶出。  请你编写程序，用来判断能否得到指*定*的车厢顺序。*假定*在*进*入车站之*前*每节车厢之间*都*是不*连*着的，*并*且它们可以*自*行移动，直到处在 B *方向*的铁轨*上*。另外*假定*车站可以停放任*意多*的车厢。但是一旦车厢*进*入车站，它就不能*再*回到 A *方向*的铁轨*上*了，  *并*且一旦它驶入 B *方向*的铁轨后，它就不能*再*回到车站。 |
| input | 第 1 行输入 N，接下来的一行是任*意多*个的出站重组顺序（A1，A2，*…*，），以 0 为  结束标志，用空格分隔。 |
| output | 输出 Yes 表*示*可以把火车（1，2，*…*，N）火车编排成*所*需*要*的顺序（A1，A2，*…*，），  否则用 No 表*示*。 |
| sample\_input | 5  1 2 3 4 5 0 |
| sample\_output | Yes |
| id | 8000070013 |
| description | NWPU 长安校*区只*有一个复印店，每年 6 月份的时候它承担了繁重的毕业*设*计*论*文*打*印工作。有时候在*打*印机*队列*中有*上*百份的*论*文*要打*印，为*此*，你可能*要*等*上*几个小时才能得到一份*论*文*打*印。  因为有些*打*印工作*比*较重*要*，*所*以 Ten School 发明和实*现*了*打*印工作*队列*的一个简单优*先*系统。每个*打*印工作被赋予了一个从 1 到 9 的优*先级*（9 是最高优*先级*，1 是最低优*先级*），*打*印机操作如下：   1. 将*队列*中的第一个*打*印工作 J 从*队列*中*取*出； 2. 如果在*队列*中有优*先级*高于 J 的*打*印工作，则不*打*印 J，*而*是将 J 移到*队列*后端； 3. 否则*打*印 J（不将J 移到*队列*后端）；   用了这个*方法*，*所*有重*要*的文件能很快被*打*印。当*然*，令*人*烦恼的是*其他要*被*打*印的  *论*文*要*等*上*更*多*的时间。  *现*在，你的任务是编写程序计算你的*打*印工作什么时候被*完*成。给出当*前队列*和优*先级列*表，以及你的*论*文在*队列*中的位置，计算需*要多*长时间你的工作才被*打*印。*假定队列*中不会*加*入附*加*的工作，为了*使*事情简单化，*设定*一件*打*印工作花费 1 小时，*向队列*中添*加*一*项打*印工作和移走一件*打*印工作瞬间就能*完*成。 |
| input | 第 1 行输入*两*个整数 n 和m，*其*中 n 是*队列*中的对象个数（1≤n≤100），m 是你的*打* |

|  |  |
| --- | --- |
|  | 印工作的位置（0≤m≤n-1）。*队列*中第 1 个位置编号为 0，第 2 个位置编号为 1，以  *此*类推。  第 2 行给出 n 个整数，范围从 1 到 9，给出*队列*中*所*有工作的优*先级*。第 1 个整数给  出第 1 个*打*印工作的优*先级*，第 2 个整数给出第 2 个*打*印工作的优*先级*，以*此*类推。 |
| output | 输出一个整数，表*示*你的*打*印工作*完*成需*要多*少小时。*假定打*印工作*进*行的时候没有  附*加*的*打*印*加*入。 |
| sample\_input | 4 2  1 2 3 4 |
| sample\_output | 2 |
| id | 8000070014 |
| description | 游乐园准备*抽*奖*选择*一批幸运儿童（X 个*人*）*进*行玩具奖励，想请你帮忙处理这件事。这*次抽*奖是将*所*有儿童排成一排，*然*后从一叠卡片的顶部*取*卡片，卡片号为 N；从*队列*中由 1 到 N *进*行报数，每*次*报到 N 时，第 N 个儿童离开*队列*，*然*后下一个儿童*再*从 1 开始报数。当报数报到*队列*结束的时候，*再*从一叠卡片的顶部*取*下一张卡片，*再*从剩余的*队列*中从第 1 *人*开始根据新的卡片号*进*行报数。最后，*队列*中的 X 个*人*获奖。*然而*，*只*有到了游戏*前*的一分钟才知道有*多*少儿童参*加抽*奖。请你编写程序，基于卡片和*队列*中儿童的数量，求出*队列*中哪些儿童可以获奖。可以确*定*最*多*用 20 张卡片。*例*如：*队列*中有 10 *名*儿童，2 个幸运位置，卡片号码为 3、5、4、3、2，*队列*位置中  1 和 8 的儿童可以获奖，过程如下：  （1）*队列* 1、2、3、4、5、6、7、8、9、10，N=10，X=2，卡片*次*序为 3、5、4、3、  2、*……*  （2）3：划掉 3、6、9，剩下 1、2、4、5、7、8、10；  （3）5：划掉 7，剩下 1、2、4、5、8、10；  （4）4：划掉 5，剩下 1、2、4、8、10；  （5）3：划掉 4，剩下 1、2、8、10；  （6）2：划掉 2、10，剩下 1、8； |
| input | 第 1 行*先*输入一个整数 N（1≤N≤50）给出参*加抽*奖的儿童*人*数，*再*输入一个整数 X  （1≤X≤9）给出有*多*少个幸运位置。  第 2 行输入 20 个整数给出*前* 20 张卡片*上*的号码，卡片号码为 1～11 的整数，用空格隔开。 |
| output | 输出幸运位置*列*表，用空格隔开。 |
| sample\_input | 10 2  3 5 4 3 2 9 6 10 10 6 2 6 7 3 4 7 4 5 3 2 |
| sample\_output | 1 8 |

第 9 季：思维体操

|  |  |
| --- | --- |
| id | 8000075000 |
| description | When Tom was a child , he was always thinking about some simple math problems ,such as  “What it’s 1 cup of water plus 1 pile of dough ..” , “100 yuan buy 100 pig” .etc..  One day Tom met a old man in his dream , in that dream the man whose name was“RuLai” gave Tom a problem :Given an N , can you calculate how many ways to write N as i \* j + i  + j (0 < i <= j) ?  Tom found the answer when N was less than 10…but if N get bigger , he found it was too difficult for him to solve.  Well , you clever Cers ,could you help little Tom to solve this problem and let him have a  good dream ? |

|  |  |
| --- | --- |
| input | each line contain an integer N (0<=N <= 100000000000). |
| output | For each case, output the number of ways in one line |
| sample\_input | 11 |
| sample\_output | 2 |
| id | 8000075001 |
| description | *现*有一块草坪，长为 20 米，宽为 2 米，*要*在横中心线*上*放置半径为 Ri 的喷水装置， 每个喷水装置的效果*都*会让以它为中心的半径为实数 Ri(0<Ri<15)的圆被湿润，这有充足的喷水装置 i（1<i<600)个，*并*且一*定*能把草坪全部湿润，你*要*做的是：*选择*尽量  少的喷水装置，把整个草坪的全部湿润。 |
| input | 第 1 行输入一个整数 n，表*示*共有 n 个喷水装置。第 2 行有n 个实数 Ri，Ri 表*示*该喷  水装置能覆盖的圆的半径。 |
| output | 输出*所*用装置的个数 |
| sample\_input | 5  2 3.2 4 4.5 6 |
| sample\_output | 2 |
| id | 8000075002 |
| description | *进*行一*次*勇闯天涯帆船冒险活动。  帆船可以在港口租到，*并*且之间没有*区*别。一条帆船最*多只*能乘坐*两*个*人*，且乘客的总重量不能超过帆船的最大承载量。我们*要*尽量*减*少这*次*活动中的花销，*所*以*要*找出可以安置*所*有旅客的最少的帆船条数。*现*在请写一个程序，读入帆船的最大承载量、旅客数目和每位旅客的重量。根据给出的规则，计算*要*安置*所*有旅客必须的最少的帆船条数，*并*输出结果。 |
| input | 第 1 行包括*两*个整数 w，n，80≤w≤200,1≤n≤300，w 为一条帆船的最大承载量，n  为*人*数。  第 2 行一组数据为每个*人*的重量（不能大于船的承载量）； |
| output | 每组*人*数*所*需*要*的最少帆船的条数。 |
| sample\_input | 85 6  5 84 85 80 84 83 |
| sample\_output | 5 |
| id | 8000075003 |
| description | *假定*一个非负数整数 n，判断 n 是不是一些数（这些数不允许重复*使*用，且为正数）  的阶乘之和，如 9=1!+2!+3!，如果是，则输出 YES，否则输出 NO； |
| input | 输入一个正整数 n<1000000 |
| output | 如果符合条件，输出 YES，否则输出 NO |
| sample\_input | 840 |
| sample\_output | YES |
| id | 8000075004 |
| description | 输入一个字符串，求*其*中最长回文子串。子串的含义是：在字符串中*连续*出*现*的字符串片段。回文的含义是：正着看和倒着看是相同的，如 abba 和 abbebba。在判断时*要*求忽略*所*有的标点和空格，且忽略大小写，但输出时按原样输出（*首*尾不*要*输出*多*余  的字符串）。输入字符串长度大于等于 1 小于等于 5000，且单独*占*一行。 |
| input | 输入一行字符串。 |
| output | 输出*所要*求的回文子串。 |
| sample\_input | Last Week,todo level odot,King |

|  |  |
| --- | --- |
| sample\_output | k,todo level odot,K |
| id | 8000075005 |
| description | 路痴一旦不高兴，就必*然*一个*人*漫无目的的出去走走。今天被老师训了，*他*又不高兴了，怎么办？*那*就出去走呗，反正丢不了。这*次*幸好记下出来时的*方向*，*并*且在一张纸*上*密密麻麻的记下了*他*拐的弯（每*次*拐的弯*都*是 90 度的弯），0 代表左拐，1 代表右拐，*那*么*多* 0、1，*他*实在看不下去了，正好遇见聪明的你，你能告诉*他*，*他现*在面  *向*哪吗？ |
| input | 第 1 行输入*他*开始时的面对*方向*，和*他*拐弯*次*数 n（0<n<100）。  第 2 行接着 n 行个数字表*示*拐的弯，用空格隔开。 |
| output | *他现*在*所*面*向*的*方向*(West、East、North、South) |
| sample\_input | East 6  0 0 0 0 0 0 |
| sample\_output | West |
| id | 8000075006 |
| description | 从*前*有*两*个国家A 和B。*两*国*都*是兵强马壮，国王更是威猛无*比*。这*两*个国家*要*争*取*一片金矿，*都*不想放弃，由于*两*个国家的国王*都*是仁爱的，害怕劳民伤财不想*打*仗， 于是便有了个*决定*，*决定*分别派出 n 个*人*来*进*行*比赛* POJ 刷题。  国王们*都*知道田忌*赛*马这件事，于是出场的顺序*都*是从弱到强；每胜一场得 2 分，平一场得 1 分，输一场得 0 分；因为*他*们*都*不相信对*方*国家的*人*，于是就想让你来当裁  判，你一*定*能做到，是吗？ |
| input | 第 1 行有一个整数 n（1<n<100），紧随着*两*行，每行有 n 个数分别代表有 A 和B 国家  的参*赛人员*的水平，数据保*证都*在 int 范围内，用空格隔开。 |
| output | 如果*两*国之间的分数相等，则输出=，输出胜的一*方*（A 或B 或=）; |
| sample\_input | 5  1 2 3 4 5  2 7 1 1 2 |
| sample\_output | A |
| id | 8000075007 |
| description | 传*说*中能站在金字塔顶的*只*有*两种*动物，一*种*是鹰，一*种*是蜗牛。*比奇*堡的小蜗*听*了这个传*说*后，大受鼓舞，*立*志*要*去爬*上*金字塔。为了实*现自*己的梦想，小蜗找到了老鹰，老鹰告诉它金字塔高 H 米，小蜗牛知道：白天*自*己能*向上*爬 10 米，但由于晚*上要*休*息*，*自*己会下滑 5 米。它想知道*自*己在第几天能站在金字塔顶，你帮*他*写个程序  吧。 |
| input | 输入一个整数 H（0<H<1000000000）代表金字塔的高度。 |
| output | 输出一个整数 n 表*示*小蜗第 n 天站在金字塔顶*上*。 |
| sample\_input | 1000 |
| sample\_output | 199 |
| id | 8000075008 |
| description | 小鹏在小学时最喜欢*上*数学课。有一*次上*课，数学老师布置了一道题目：给出一组有规律的整数，*而*且*只*给出*前* 5 *项*，求出后面 5 *项*。*比*如：1，2，3，4，5，□，□，  □，□，□。*显然*这是等差数*列*，*所*以答*案*应该是 6，7，8，9，10。老师规*定*规律*只*  有三*种*：等差数*列*、等*比*数*列*、斐波*那*契数*列*（f[i]=f[i-2]+f[i-1]）；且公差、公*比*均为整数，*所*有出*现*的数字不会超出 int 范围。聪明的你，来*比*一*比*你和小鹏*谁*算的快。 |
| input | 每行 5 个整数，代表题目给出的*前* 5 *项*，用空格分隔。 |
| output | 如果能找到符合条件的规律，则输出 5 个整数，代表后 5 *项*的值。否则，请输出 NULL |

|  |  |
| --- | --- |
| sample\_input | 2 3 5 8 13 |
| sample\_output | 21 34 55 89 144 |
| id | 8000075009 |
| description | *现*在给你不共线的三个点 A、B、C 的坐标，它们一*定*能组成一个三角形，你需*要*判断 A、B、C 是顺时针给出的还是逆时针给出的？  如图为顺时针给出    如图为逆时针给出 |
| input | 输入 6 个整数 x1,y1,x2,y2,x3,y3 分别表*示* A、B、C 三个点的横纵坐标。（坐标值*都*在  0 到 10000 之间） |
| output | 如果这三个点是顺时针给出的，请输出 1，逆时针给出则输出 0 |
| sample\_input | 0 0 1 1 1 3 |
| sample\_output | 0 |
| id | 8000075010 |
| description | *设*计一个程序求出 A\*B，*然*后将*其*结果每一位相*加*得到 C，如果 C 的位数大于等于 2，  继*续*将 C 的*各*位数相*加*，直到结果是个一位数 k。*例*如： 6\*8=48  4+8=12  1+2=3  输出 3 *即*可。 |
| input | 输入给出*两*个非负整数 m，n（0≤m，n≤1000000000000)。 |
| output | 输出 k。 |
| sample\_input | 1234567 67 |
| sample\_output | 4 |
| id | 8000075011 |
| description | 在一个划分成网格的操场*上*，n 个士兵散乱地站在网格点*上*。网格点由整数坐标(x,y) 表*示*。士兵们可以沿网格边*上*、下、左、右移动一步，但在同一时刻任一网格点*上只*能有一*名*士兵。按照军官的命令，士兵们*要*整齐地*列*成一个水平*队列*，*即*排*列*成(x,y),(x+1,y),*…*,(x+n-1,y)。如*何选择* x 和 y 的值才能*使*士兵们以最少的总移动步数排成一*列*。  计算*使所*有士兵排成一行需*要*的最少移动步数。 |
| input | 第 1 行输入士兵数 n，1≤n≤10000。接下来 n 行是士兵的初始位置，每行 2 个整数 x  和 y，-10000≤x，y≤10000。 |
| output | 输出士兵排成一行需*要*的最少移动步数。 |
| sample\_input | 5  1 2 |

|  |  |
| --- | --- |
|  | 2 2  1 3  3 -2  3 3 |
| sample\_output | 8 |
| id | 8000075012 |
| description | 你能*使*一叠在桌子*上*的卡片*向*外伸出*多*远？  如果你有一个卡片，这张卡片最*多*可以*向*桌子外伸出卡片的一半长度（*假设*该卡片必须垂直于桌子），如果有*两*个卡片，就可以让*上*面的卡片*向*外伸出下面*那*张卡片的一半长度，*而*下面*那*张卡伸出桌子卡片的三分之一长度，*所*以*两*张卡片总的*向*外延伸 1/2  + 1/3 = 5/6 卡长度。以*此*类推，N 张卡片*向*外延伸 1/2+1/3+1/4+*…*+1/（n +1）卡片长度，最*上*面的*向*外延伸 1/2，第*二*张*向*外延伸 1/3，第三张*向*外延伸 1/4，*…*，最下面一张*向*外延伸 1 /（n +1）。如下图*所示*。 |
| input | 每个测试数据是一个 3 位正浮点数 c，最小值为 0.01，最大值为 5.20。 |
| output | 输出卡片的最小数目，为整型。 |
| sample\_input | 1.5 |
| sample\_output | 6 |
| id | 8000075013 |
| description | Problems in Computer Science are often classified as belonging to a certain class of problems (e.g.,NP,Unsolvable,Recursive).In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.  Consider the following algorithm:   1. input n 2. print n 3. if n = 1 then STOP 4. if n is odd then n <- 3n + 1 5. else n <- n / 2 6. GOTO 2   Given the input 22, the following sequence of numbers will be printed 22 11 34 17 52 26  13 40 20 10 5 16 8 4 2 1  It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that 0 < n < 1,000,000 (and, in fact, for many more numbers than this.)  Given an input n, it is possible to determine the number of numbers printed (including the 1). For a given n this is called the cycle-length of n. In the example above, the cycle length of 22 is 16.  For any two numbers i and j you are to determine the maximum cycle length over all |

|  |  |
| --- | --- |
|  | numbers between i and j. |
| input | The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.  You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j.  You can assume that no opperation overflows a 32-bit integer. |
| output | For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same  line). |
| sample\_input | 1 10 |
| sample\_output | 1 10 20 |
| id | 8000075014 |
| description | If you ever see a televised report on stock market activity, you'll hear the anchorperson say something like ``Gainers outnumbered losers 14 to 9,'' which means that for every 14 stocks that increased in value that day, approximately 9 other stocks declined in value. Often, as you hear that, you'll see on the screen something like this:  Gainers 1498  Losers 902  As a person with a head for numbers, you'll notice that the anchorperson could have said  ``Gainers outnumbered losers 5 to 3'', which is a more accurate approximation to what really happened. After all, the exact ratio of winners to losers is (to the nearest millionth) 1.660754, and he reported a ratio of 14 to 9, which is 1.555555, for an error of 0.105199; he could have said ``5 to 3'', and introduced an error of only 1.666667-1.660754=0.005913. The estimate ``5 to 3'' is not as accurate as ``1498 to 902'' of course; evidently, another goal is to use small integers to express the ratio. So, why did the anchorperson say ``14 to 9?'' Because his algorithm is to lop off the last two digits of each number and use those as the approximate ratio.  What the anchorman needs is a list of rational approximations of increasing accuracy, so that he can pick one to read on the air. Specifically, he needs a sequence {a\_1, a\_2, ..., a\_n} where a\_1 is a rational number with denominator 1 that most exactly matches the true ratio of winners to losers (rounding up in case of ties), a\_{i+1} is the rational number with least denominator that provides a more accurate approximation than a\_i, and a\_n is the exact ratio, expressed with the least possible denominator. Given this sequence, the anchorperson can decide which ratio gives the best tradeoff between accuracy and simplicity.  For example, if 5 stocks rose in price and 4 fell, the best approximation with denominator 1 is 1/1; that is, for every stock that fell, about one rose. This answer differs from the exact answer by 0.25 (1.0 vs 1.25). The best approximations with two in the denominator are 2/2 and 3/2, but neither is an improvement on the ratio 1/1, so neither would be considered. The best approximation with three in the denominator 4/3, is more accurate than any seen so far, so it is one that should be reported. Finally, of course, 5/4 is exactly the ratio, and so it is the last number reported in the sequence.  Can you automate this process and help the anchorpeople? |

|  |  |
| --- | --- |
| input | input contains one pair of positive integers. Each pair is on a line by itself, beginning in the first column and with a space between the two numbers. The first number of a pair is the number of gaining stocks for the day, and the second number is the number of losing stocks  for the day. The total number of stocks never exceeds 5000. |
| output | For each input pair, the standard output should contain a series of approximations to the ratio of gainers to losers. The first approximation has '1' as denominator, and the last is exactly the ratio of gainers to losers, expressed as a fraction with least possible denominator. The approximations in between are increasingly accurate and have increasing denominators, as described above.  The approximations for a pair are printed one to a line, beginning in column one, with the numerator and denominator of an approximation separated by a slash (``/''). A blank line  separates one sequence of approximations from another. |
| sample\_input | 5 4 |
| sample\_output | 1/1  4/3  5/4 |

第 10 季：挑战 ACM-ICPC

|  |  |
| --- | --- |
| id | 8000005000 |
| description | As part of an arithmetic competency program, your students will be given randomly generated lists of from 2 to 15 unique positive integers and asked to determine how many items in each list are twice some other item in the same list. You will need a program to help you with the grading. This program should be able to scan the lists and output the correct answer for each one. For example, given the list  1 4 3 2 9 7 18 22  your program should answer 3, as 2 is twice 1, 4 is twice 2, and 18 is twice 9. |
| input | There will be one list of numbers per line. Each list will contain from 2 to 15 unique positive integers. No integer will be larger than 99. Each line will be terminated with the  integer 0, which is not considered part of the list. |
| output | The output contain a count of the items that are double some other item. |
| sample\_input | 1 4 3 2 9 7 18 22 0 |
| sample\_output | 3 |
| id | 8000005001 |
| description | Some positive integers can be represented by a sum of one or more consecutive prime numbers. How many such representations does a given positive integer have? For example, the integer 53 has two representations 5 + 7 + 11 + 13 + 17 and 53. The integer 41 has three  representations 2+3+5+7+11+13, 11+13+17, and 41. The integer 3 has only one representation, which is 3. The integer 20 has no such representations. Note that summands must be consecutive prime  numbers, so neither 7 + 13 nor 3 + 5 + 5 + 7 is a valid representation for the integer 20. Your mission is to write a program that reports the number of representations for the given  positive integer. |
| input | The input is a sequence of positive integers each in a separate line. The integers are  between 2 and 10 000, inclusive. The end of the input is indicated by a zero. |
| output | The output should be composed of lines each corresponding to an input line except the last  zero. An output line includes the number of representations for the input integer as the sum |

|  |  |
| --- | --- |
|  | of one or more consecutive prime numbers. No other characters should be inserted in the  output. |
| sample\_input | 2  17  41  20  666  0 |
| sample\_output | 1  2  3  0  0 |
| id | 8000005002 |
| description | Fred Mapper is considering purchasing some land in Louisiana to build his house on. In the process of investigating the land, he learned that the state of Louisiana is actually shrinking by 50 square miles each year, due to erosion caused by the Mississippi River. Since Fred is hoping to live in this house the rest of his life, he needs to know if his land is going to be lost to erosion.  After doing more research, Fred has learned that the land that is being lost forms a semicircle(Red). This semicircle is part of a circle centered at (0,0), with the line that bisects the circle being the X axis. Locations below the X axis are in the water(Blue). The semicircle has an area of 0 at the beginning of year 1. (Semicircle illustrated in the Figure.) |
| input | The first line of input will be a positive integer indicating how many data sets will be included (N). Each of the next N lines will contain the X and Y Cartesian coordinates of the land Fred is considering. These will be floating point numbers measured in miles. The Y  coordinate will be non-negative. (0,0) will not be given. |
| output | For each data set, a single line of output should appear. This line should take the form of:  “ N Z” Where N is the data set (counting from 1), and Z is the first year (start from 1) this property will be within the semicircle AT THE END OF YEAR Z. Z must be an integer. |
| sample\_input | 2  1.0 1.0  25.0 0.0 |
| sample\_output | 1 1 |

|  |  |
| --- | --- |
|  | 2 20 |
| id | 8000005003 |
| description | Find and list all four-digit numbers in decimal notation that have the property that the sum of its four digits equals the sum of its digits when represented in hexadecimal (base 16) notation and also equals the sum of its digits when represented in duodecimal (base 12) notation.  For example, the number 2991 has the sum of (decimal) digits 2+9+9+1 = 21. Since 2991 = 1\*1728 + 8\*144 + 9\*12 + 3, its duodecimal representation is 189312, and these digits also sum up to 21. But in hexadecimal 2991 is BAF16, and 11+10+15 = 36, so 2991 should be rejected by your program.  The next number (2992), however, has digits that sum to 22 in all three representations (including BB016), so 2992 should be on the listed output. (We don't want decimal numbers with fewer than four digits -- excluding leading zeroes -- so that 2992 is the first  correct answer.) |
| input | There is no input for this problem |
| output | Your output is to be 2992 and all larger four-digit numbers that satisfy the requirements (in strictly increasing order), each on a separate line with no leading or trailing blanks, ending with a new-line character. There are to be no blank lines in the output. The first few lines of  the output are shown below. |
| sample\_input |  |
| sample\_output | 2992  2993  2994  … |

C/C++试题

|  |  |
| --- | --- |
| id | 8000005004 |
| description | A checksum is an algorithm that scans a packet of data and returns a single number. The idea is that if the packet is changed, the checksum will also change, so checksums are often used for detecting transmission errors, validating document contents, and in many other situations where it is necessary to detect undesirable changes in data.  For this problem, you will implement a checksum algorithm called Quicksum. A Quicksum packet allows only uppercase letters and spaces. It always begins and ends with an uppercase letter. Otherwise, spaces and letters can occur in any combination, including consecutive spaces.  A Quicksum is the sum of the products of each character's position in the packet times the character's value. A space has a value of zero, while letters have a value equal to their position in the alphabet. So, A=1, B=2, etc., through Z=26. Here are example Quicksum calculations for the packets "ACM" and "MID CENTRAL":  ACM: 1\*1 + 2\*3 + 3\*13 = 46  MID CENTRAL: 1\*13 + 2\*9 + 3\*4 + 4\*0 + 5\*3 + 6\*5 + 7\*14 + 8\*20 + 9\*18 + 10\*1 + 11\*12 = 650 |
| input | The input consists of one or more packets followed by a line containing only # that signals  the end of the input. Each packet is on a line by itself, does not begin or end with a space, and contains from 1 to 255 characters. |
| output | For each packet, output its Quicksum on a separate line in the output. |

|  |  |
| --- | --- |
| sample\_input | ACM  MID CENTRAL  REGIONAL PROGRAMMING CONTEST ACN  A C M ABC BBC  # |
| sample\_output | 46  650  4690  49  75  14  15 |
| id | 8000005005 |
| description | Judging a programming contest is hard work, with demanding contestants, tedious decisions,and monotonous work. Not to mention the nutritional problems of spending 12 hours with only donuts, pizza, and soda for food. Still, it can be a lot of fun.  Software that automates the judging process is a great help, but the notorious unreliability of some contest software makes people wish that something better were available. You are part of a group trying to develop better, open source, contest management software, based on the principle of modular design.  Your component is to be used for calculating the scores of programming contest teams and determining a winner. You will be given the results from several teams and must determine the winner.  Scoring  There are two components to a team's score. The first is the number of problems solved. The second is penalty points, which reflects the amount of time and incorrect submissions made before the problem is solved. For each problem solved correctly, penalty points are charged equal to the time at which the problem was solved plus 20 minutes for each incorrect submission. No penalty points are added for problems that are never solved.  So if a team solved problem one on their second submission at twenty minutes, they are charged 40 penalty points. If they submit problem 2 three times, but do not solve it, they are charged no penalty points. If they submit problem 3 once and solve it at 120 minutes, they are charged 120 penalty points. Their total score is two problems solved with 160 penalty points.  The winner is the team that solves the most problems. If teams tie for solving the most  problems,then the winner is the team with the fewest penalty points. |
| input | For the programming contest your program is judging, there are four problems. You are guaranteed that the input will not result in a tie between teams after counting penalty points.  Line 1 < nTeams >  Line 2 - n+1 < Name > < p1Sub > < p1Time > < p2Sub > < p2Time > ... < p4Time >  The first element on the line is the team name, which contains no whitespace.Following |

|  |  |
| --- | --- |
|  | that, for each of the four problems, is the number of times the team submitted a run for that problem and the time at which it was solved correctly (both integers). If a team did not solve a problem, the time will be zero. The number of submissions will be at least one if the  problem was solved. |
| output | The output consists of a single line listing the name of the team that won, the number of  problems they solved, and their penalty points. |
| sample\_input | 4  Stars 2 20 5 0 4 190 3 220  Rockets 5 180 1 0 2 0 3 100  Penguins 1 15 3 120 1 300 4 0  Marsupials 9 0 3 100 2 220 3 80 |
| sample\_output | Penguins 3 475 |
| id | 8000005006 |
| description | If a and d are relatively prime positive integers, the arithmetic sequence beginning with a and increasing by d, i.e., a, a + d, a + 2d, a + 3d, a + 4d, ..., contains infinitely many prime numbers. This fact is known as Dirichlet's Theorem on Arithmetic Progressions, which had been conjectured by Johann Carl Friedrich Gauss (1777 - 1855) and was proved by Johann Peter Gustav Lejeune Dirichlet (1805 - 1859) in 1837.  For example, the arithmetic sequence beginning with 2 and increasing by 3, i.e.,  2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 47, 50, 53, 56, 59, 62, 65, 68, 71,  74, 77, 80, 83, 86, 89, 92, 95, 98, ... ,  contains infinitely many prime numbers  2, 5, 11, 17, 23, 29, 41, 47, 53, 59, 71, 83, 89, ... .  Your mission, should you decide to accept it, is to write a program to find the nth prime number in this arithmetic sequence for given positive integers a, d, and n. |
| input | The input is a sequence of datasets. A dataset is a line containing three positive integers a, d, and n separated by a space. a and d are relatively prime. You may assume a <= 9307, d  <= 346, and n <= 210.  The end of the input is indicated by a line containing three zeros separated by a space. It is not a dataset. |
| output | The output should be composed of as many lines as the number of the input datasets. Each line should contain a single integer and should never contain extra characters.  The output integer corresponding to a dataset a, d, n should be the nth prime number among those contained in the arithmetic sequence beginning with a and increasing by d.  FYI, it is known that the result is always less than 1000000 (one million) under this input  condition. |
| sample\_input | 367 186 151  179 10 203  271 37 39  103 230 1  0 0 0 |
| sample\_output | 92809  6709  12037  103 |

|  |  |
| --- | --- |
| id | 8000005007 |
| description | Bill and Ted are taking a road trip. But the odometer in their car is broken, so they don't know how many miles they have driven. Fortunately, Bill has a working stopwatch, so they can record their speed and the total time they have driven. Unfortunately, their record keeping strategy is a little odd, so they need help computing the total distance driven. You are to write a program to do this computation.  For example, if their log shows    this means they drove 2 hours at 20 miles per hour, then 6-2=4 hours at 30 miles per hour, then 7-6=1 hour at 10 miles per hour. The distance driven is then (2)(20) + (4)(30) + (1)(10)  = 40 + 120 + 10 = 170 miles. Note that the total elapsed time is always since the beginning  of the trip, not since the previous entry in their log. |
| input | The input consists of one or more data sets. Each set starts with a line containing an integer n, 1 <= n <= 10, followed by n pairs of values, one pair per line. The first value in a pair, s, is the speed in miles per hour and the second value, t, is the total elapsed time. Both s and t are integers, 1 <= s <= 90 and 1 <= t <= 12. The values for t are always in strictly  increasing order. A value of -1 for n signals the end of the input. |
| output | For each input set, print the distance driven |
| sample\_input | 3  20 2  30 6  10 7  2  60 1  30 5  -1 |
| sample\_output | 170  180 |
| id | 8000005008 |
| description | In Africa there is a very special species of bee. Every year, the female bees of such species give birth to one male bee, while the male bees give birth to one male bee and one female bee, and then they die!  Now scientists have accidentally found one "magical female bee" of such special species to the effect that she is immortal, but still able to give birth once a year as all the other female bees. The scientists would like to know how many bees there will be after N years. Please write a program that helps them find the number of male bees and the total number of all  bees after N years. |
| input | Each line of input contains an integer N (≥ 0). Input ends with a case where N = -1. (This |

|  |  |
| --- | --- |
|  | case should NOT be processed.) |
| output | Each line of output should have two numbers, the first one being the number of male bees after N years, and the second one being the total number of bees after N years. (The two  numbers will not exceed 2e+32.) |
| sample\_input | 1  3  -1 |
| sample\_output | 1 2  4 7 |
| id | 8000005009 |
| description | George took sticks of the same length and cut them randomly until all parts became at most 50 units long. Now he wants to return sticks to the original state, but he forgot how many sticks he had originally and how long they were originally. Please help him and design a program which computes the smallest possible original length of those sticks. All lengths  expressed in units are integers greater than zero. |
| input | The input contains blocks of 2 lines. The first line contains the number of sticks parts after cutting, there are at most 64 sticks. The second line contains the lengths of those parts  separated by the space. The last line of the file contains zero. |
| output | The output should contains the smallest possible length of original sticks, one per line. |
| sample\_input | 9  5 2 1 5 2 1 5 2 1  4  1 2 3 4  0 |
| sample\_output | 6  5 |
| id | 8000005010 |
| description | The (Three peg) Tower of Hanoi problem is a popular one in computer science. Briefly the problem is to transfer all the disks from peg-A to peg-C using peg-B as intermediate one in such a way that at no stage a larger disk is above a smaller disk. Normally, we want the minimum number of moves required for this task. The problem is used as an ideal example for learning recursion. It is so well studied that one can find the sequence of moves for smaller number of disks such as 3 or 4. A trivial computer program can find the case of large number of disks also.  Here we have made your task little bit difficult by making the problem more flexible. Here |

|  |  |
| --- | --- |
|  | the disks can be in any peg initially.    If more than one disk is in a certain peg, then they will be in a valid arrangement (larger disk will not be on smaller ones). We will give you two such arrangements of disks. You will have to find out the minimum number of moves, which will transform the first arrangement into the second one. Of course you always have to maintain the constraint that smaller disks must be upon the larger ones. |
| input | The input file contains at most 100 test cases. Each test case starts with a positive integer N ( 1≤N≤60), which means the number of disks. You will be given the arrangements in next two lines. Each arrangement will be represented by N integers, which are 1, 2 or 3. If the i-th ( 1≤i≤N) integer is 1, you should consider that i-th disk is on Peg-A. The end of the  input is indicated by a zero.. This case should not be processed. |
| output | Output of each test case should consist of a line starting with `Case #: ' where # is the test case number. It should be followed by the minimum number of moves as specified in the  problem statement. |
| sample\_input | 3  1 1 1  2 2 2  3  1 2 3  3 2 1  0 |
| sample\_output | 7  3 |
| id | 8000005011 |
| description | Dexter is tired of Dee Dee. So he decided to keep Dee Dee busy in a game. The game he planned for her is quite easy to play but not easy to win at least not for Dee Dee. But Dexter does not have time to spend on this silly task, so he wants your help.  There will be a button, when it will be pushed a random number N will be chosen by computer. Then on screen there will be numbers from 1 to N. Dee Dee can choose any number of numbers from the numbers on the screen, and then she will command computer to subtract a positive number chosen by her (not necessarily on screen) from the selected numbers. Her objective will be to make all the numbers 0.  For example if N = 3, then on screen there will be 3 numbers on screen: 1, 2, 3. Say she now selects 1 and 2. Commands to subtract 1, then the numbers on the screen will be: 0, 1,  3. Then she selects 1 and 3 and commands to subtract 1. Now the numbers are 0, 0, 2. Now |

|  |  |
| --- | --- |
|  | she subtracts 2 from 2 and all the numbers become 0.  Dexter is not so dumb to understand that this can be done very easily, so to make a twist he will give a limit L for each N and surely L will be as minimum as possible so that it is still possible to win within L moves. But Dexter does not have time to think how to determine L for each N, so he asks you to write a code which will take N as input and give L as output. |
| input | Input consists of several lines each with N such that 1 ≤ N ≤ 1,000,000,000. The end of  the input is indicated by a zero. |
| output | For each N output L in separate lines. |
| sample\_input | 1  2  3  0 |
| sample\_output | 1  2  2 |
| id | 8000005012 |
| description | HanBo Favorite number is 1, So he will often use some ways to put some numbers to 1,and to this endless pride.  He would only two ways will be:  (1) Put a number m divided by a prime number p . of course, this number must be divisible,  i.e m = m / p  (2) The number m of one minus 1, i.e m = m-1  There are a whim, he wanted [a, b] interval all the numbers one by one into a 1, which is a huge boring works, so he wanted to know how much he was operating at least in order to achieve the purpose. |
| input | Input contains multiple sets of data (1000 data), the end of the input is indicated by 0 0. Each set of data begins with two integers: a, b (0 <a <= b <= 100000), meaning as the  meaning of problems described. |
| output | Each data output line, at least operand numbers. |
| sample\_input | 2 3  3 5  11 12  0 0 |
| sample\_output | 2  4  3 |
| id | 8000005013 |
| description | A Communist regime is trying to redistribute wealth in a village. They have have decided to sit everyone around a circular table. First, everyone has converted all of their properties to coins of equal value, such that the total number of coins is divisible by the number of people in the village. Finally, each person gives a number of coins to the person on his right and a number coins to the person on his left, such that in the end, everyone has the same number of coins. Given the number of coins of each person, compute the minimum number of coins that must be transferred using this method so that everyone has the same number  of coins. |

|  |  |
| --- | --- |
| input | There is a number of inputs. Each input begins with n(n<1000001), the number of people in the village. n lines follow, giving the number of coins of each person in the village, in counterclockwise order around the table. The total number of coins will fit inside an  unsigned 64 bit integer. |
| output | For each input, output the minimum number of coins that must be transferred on a single  line. |
| sample\_input | 3  100  100  100  4  1  2  5  4  0 |
| sample\_output | 0  4 |
| id | 8000005014 |
| description | Programming contests became so popular in the year 2397 that the governor of New Earck  — the largest human-inhabited planet of the galaxy — opened a special Alley of Contestant Memories (ACM) at the local graveyard. The ACM encircles a green park, and holds the holographic statues of famous contestants placed equidistantly along the park perimeter. The alley has to be renewed from time to time when a new group of memorials arrives.  When new memorials are added, the exact place for each can be selected arbitrarily along the ACM, but the equidistant disposition must be maintained by moving some of the old statues along the alley.  Surprisingly, humans are still quite superstitious in 24th century: the graveyard keepers believe the holograms are holding dead people souls, and thus always try to renew the ACM with minimal possible movements of existing statues (besides, the holographic equipment is very heavy). Statues are moved along the park perimeter. Your work is to find a renewal plan which minimizes the sum of travel distances of all statues. Installation of a  new hologram adds no distance penalty, so choose the places for newcomers wisely! |
| input | Input file contains two integer numbers: n — the number of holographic statues initially located at the ACM, and m — the number of statues to be added (2 ≤ n ≤ 1000, 1 ≤ m ≤ 1000). The length of the alley along the park perimeter is exactly 10 000 feet.Te end  of the input is indicated by 0 0. |
| output | Write a single real number to the output file — the minimal sum of travel distances of all  statues (in feet). The answer must be precise to at least 4 digits after decimal point. |
| sample\_input | 2 1  2 3  0 0 |
| sample\_output | 1666.6667  1000.0000 |