

## 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	<p>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)。</p> 
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	请编写程序，用迭代法求 $x = \sqrt{a}$ 。求平方根的迭代公式为： $x_{n+1} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right)$ 要求前后两次求出的 x 的差的绝对值少于 $10^{-5}$ 。
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： $2x^3 - 4x^2 + 3x - 6 = 0$
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	<p>有一个分数数列：</p> $\frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \dots$ <p>求出这个数列前 20 项之和。</p>
input	
output	输出保留六位小数。
sample_input	
sample_output	32.660261
id	8000022006
description	<p>假设银行整存整取存款不同期限的月息利率分别为：</p> <p>0.63% 期限=1 年      0.66% 期限=2 年      0.69% 期限=3 年</p> <p>0.75% 期限=5 年      0.84% 期限=8 年</p> <p>利息=本金*月息利率*12*存款年限。</p> <p>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）。</p>
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	<p>编写程序利用下面公式计算 <math>\pi</math> 的近似值：</p> $\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots)$ <p>直到括号中最后一项的绝对值小于 <math>10^{-6}</math> 为止。</p>
input	
output	输出 $\pi$ 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。

	$\sum_{j=1}^n \frac{(-1)^{j-1} 2^j}{[2^j + (-1)^j][2^{j+1} + (-1)^{j+1}]}$
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： $1^3=1$ ； $2^3=3+5$ ； $3^3=7+9+11$ 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： 求 $x^a$ 的最后三位数，其中 x，a 从键盘上输入。
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	某级数的前两项 $A_1=1$ ， $A_2=1$ ，以后各项具有如下关系： $A_n=A_{n-2}+2A_{n-1}$ 。 编写程序要求依次对于整数 M=100, 1000 和 10000 求出对应的 n 值， 使其满足： $S_n < M$ 且 $S_{n+1} \geq M$ 这里 $S_n = A_1 + A_2 + \dots + A_n$
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	<p>设 <math>n</math> 为正整数，求解 <math>n</math> 使不等式</p> $a < 1 + \frac{1}{1+1/2} + \frac{1}{1+1/2+1/3} + \cdots + \frac{1}{1+1/2+\cdots+1/n} < b$ <p>成立。<math>a, b</math> 为上下限。注意：<math>n</math> 可能是一个值，也可能是一个区间</p>
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 $\alpha$ 和 $\beta$ 两种粒子，每秒钟内一个 $\alpha$ 粒子可以裂变为 3 个 $\beta$ 粒子，而一个 $\beta$ 粒子可以裂变为 1 个 $\alpha$ 粒子和 2 个 $\beta$ 粒子。若在 $t=0$ 时刻的反应堆中只有一个 $\alpha$ 粒子，求在 $t$ 秒时反应堆裂变产生的 $\alpha$ 粒子和 $\beta$ 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 $\alpha$ 粒子和 $\beta$ 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 $a\sim j$ 恰好为数字 $0\sim 9$ 的一个排列， $2\leq n\leq 79$ 。
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 $x\geq y$ ，使得 $\frac{1}{k} = \frac{1}{x} + \frac{1}{y}$
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。 $1\leq n\leq 18$ ， $-10\leq S_i\leq 10$ 。
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 $n$ ( $n\leq 30$ ) 个盒子放成一行，但至少要有 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	<p>输入两个正整数 <math>n &lt; m &lt; 1000000</math>，输出</p> $\frac{1}{n^2} + \frac{1}{(n+1)^2} + \dots + \frac{1}{m^2}$ <p>保留 5 位小数。注意：本题有数据陷进。</p>
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	<p>输入正整数 <math>a</math>、<math>b</math>、<math>c</math>，输出 <math>a/b</math> 的小数形式，精确到小数点后 <math>c</math> 位。其中 <math>a, b \leq 1000000</math>，<math>c \leq 100</math>。注意：本题有数据陷进。</p>
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	<p>用 1、2、3、<math>\dots</math>、9 组成 3 个三位数：<math>abc</math>、<math>def</math> 和 <math>ghi</math>，每个数字恰好使用了 1 次，要求 <math>abc: def: ghi = 1: 2: 3</math>。输出所有解。</p>
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	<p>计算 1 到 <math>N</math> 中数字“1”出现的个数，其中 <math>1 \leq N \leq 1000000000</math>。</p>
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	<p>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。</p> <p>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</p>



	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 $x \neq y < 2000000000$ ， $0 < a, b < 2000000000$ ， $0 < L < 2100000000$ 。
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 $f(x) = 1 / (1 + x^2)$ 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

	$s = \sum_{i=1}^n (x_i - x)^2$ <p>其中 <math>x</math> 为 <math>x_1, x_2, \dots</math> 的平均数。请记住：不能使用数组。</p>
input	第 1 行输入 $n$ ，为整型 第 2 行输入 $x_1, x_2, \dots$ ，均为 <code>double</code> 型，用空格隔开。
output	输出 $s$ ， <code>double</code> 型，小数点后 6 位。
sample_input	5 1 2 3 4 5
sample_output	10.000000
id	8000030004
description	已知 <code>ack</code> 函数对于 $m \geq 0$ 和 $n \geq 0$ 有定义： <code>ack(0,n)=n+1</code> 、 <code>ack(m,0)=ack(m-1,1)</code> 、 <code>ack(m,n)=ack(m-1,ack(m,n-1))</code> 。输入 $m$ 和 $n$ ，求解 <code>ack</code> 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 <code>inline int xchg(unsigned char n)</code> ，计算将 <code>unsigned char</code> 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 <code>getbit(n,k)</code> ；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 $k$ ( $1 \leq k \leq 16$ )，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 <code>value</code> 左右循环移位（即移出的位在另一端填入）。函数原型为 <code>int move(int value,int n)</code> ；其中 <code>value</code> 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	<p>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。</p> <p>例如，对于小明来说：</p> <p>（1）1234 和 34 是相等的</p> <p>（2）35+80=15</p> <p>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。</p>
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	<p>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？</p> <p>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。</p>
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 $0 \leq A, B \leq 600000$
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 $N$ ( $1 \leq N \leq 100$ ) 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	8000038001
description	输入两个 $N$ ( $1 \leq N \leq 100$ ) 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 $N \times N$ ( $1 \leq N \leq 100$ ) 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 $n \times n$ 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 $N \times N$ ( $1 \leq N \leq 100$ ) 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 $n \times n$ 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 $N \times N$ ( $1 \leq N \leq 100$ ) 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 $n \times n$ 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 $N \times N$ ( $1 \leq N \leq 100$ ) 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 $n \times n$ 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 $N \times M$ ( $1 \leq N, M \leq 100$ ) 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 <code>double avg(int A[],int s,int e)</code> ，计算一个 $N$ ( $1 \leq N \leq 100$ ) 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 $0 \leq s < e < N$ 。在主函数输入输出，调用函数 <code>avg</code> 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 <code>double</code> 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 <code>void BubbleSort(int A[],int s,int m)</code> ，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 $N$ ( $1 \leq N \leq 100$ ) 个元素， $0 \leq s < N$ ，且 $s+m \leq N$ 。在主函数输入输出，调用函数 <code>BubbleSort</code> 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 <code>void SelectionSort(int A[],int s,int m)</code> ，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 $N$ ( $1 \leq N \leq 100$ ) 个元素， $0 \leq s < N$ ，且 $s+m \leq N$ 。在主函数输入输出，调用函数 <code>SelectionSort</code> 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 <code>void InsertionSort(int A[],int s,int m)</code> ，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 $N$ ( $1 \leq N \leq 100$ ) 个元素， $0 \leq s < N$ ，且 $s+m \leq N$ 。在主函数输入输出，调用函数 <code>InsertionSort</code> 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（ $1 \leq N \leq 100$ ）个元素， $0 \leq s < N$ ，且 $s+m \leq N$ 。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	山迪要出席一个周末表演晚会，他在会上要表演 <b>卡片魔术</b> 。他有 n（ $0 < n \leq 100$ ）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，可是山迪的助手不小心把卡片打乱了。这可急坏了山迪，忙令助手迅速通过一些操作把这些卡片变回有序的，而山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，山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（ $1 \leq N \leq 100$ ）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 <b>二分查找法</b> 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 <code>null</code> 。
input	第 1 行输入 <code>n</code> ，第 2 行输入 <code>n</code> 个已排好序的数组元素，数据之间用空格分隔。第 3 行输入 <code>m</code> 。
output	若找到输出它数组的下标，否则输出 <code>null</code>
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	<p>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:</p> <p>A   A#=Bb   B   C   C#=Db   D   D#=Eb   E   F   F#=Gb   G   G#=Ab</p> <p>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.</p> <p>In naming his preludes, Mr. B used all the keys except the following 10, which were named instead by their alternate names:</p> <p>Ab minor   A# major   A# minor   C# major   Db minor D# major   D# minor   Gb major   Gb minor   G# major</p> <p>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.</p>
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	<p>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?</p>
input	Each test case contains a single line indicating a Roman Numerals that to be translated. All



	<p>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.</p> <p>I,II,III,IV,V,VI,VII,VIII,IX,X,XI,XII</p> <p>1,2,3,4,5,6,7,8,9,10,11,12</p>
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	<p>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"<u>www.nwpu.edu.cn</u>",调用函数并得到结果。</p> <p>(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。</p> <p>(2) 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。</p> <p>(3) 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。</p>
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	<u>www.nwpu.edu.cn</u> 5 5
sample_output	www.n du.cn wpu.e
id	8000046003
description	<p>As is known to all,if you throw a coin up and let it dropped 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.</p> <p>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 possiblility 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.</p>
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 successedded,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

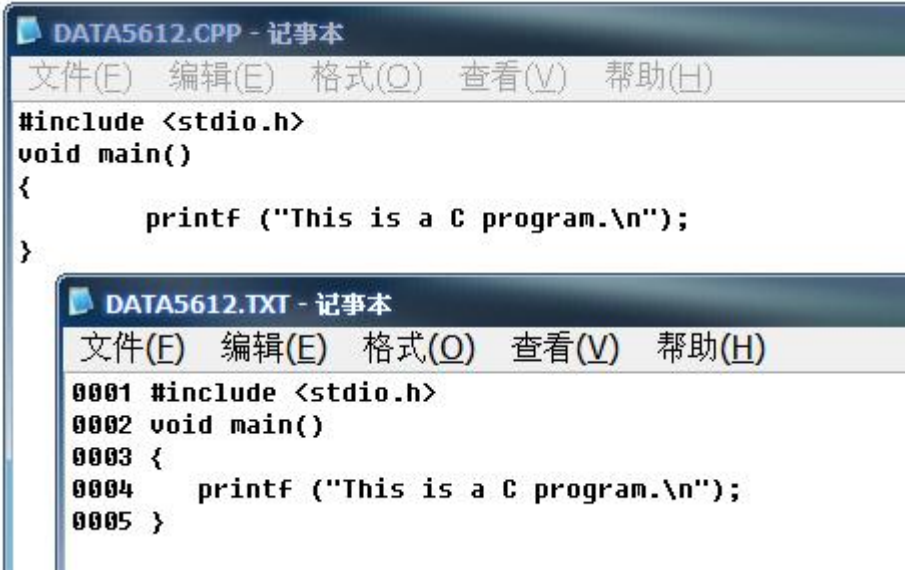
	中并打印出来。 条件：从字符串中间一分为二，左边部分按字符的 ASCII 值降序排序，右边部分按字符的 ASCII 值升序排序；排序后，左边部分与右边部分进行交换。如果原字符串长度为奇数，则最中间的字符不参加排序，字符仍放在原位置上。
input	输入一行字符串。
output	输出结果字符串。
sample_input	abcd9876
sample_output	6789dcba

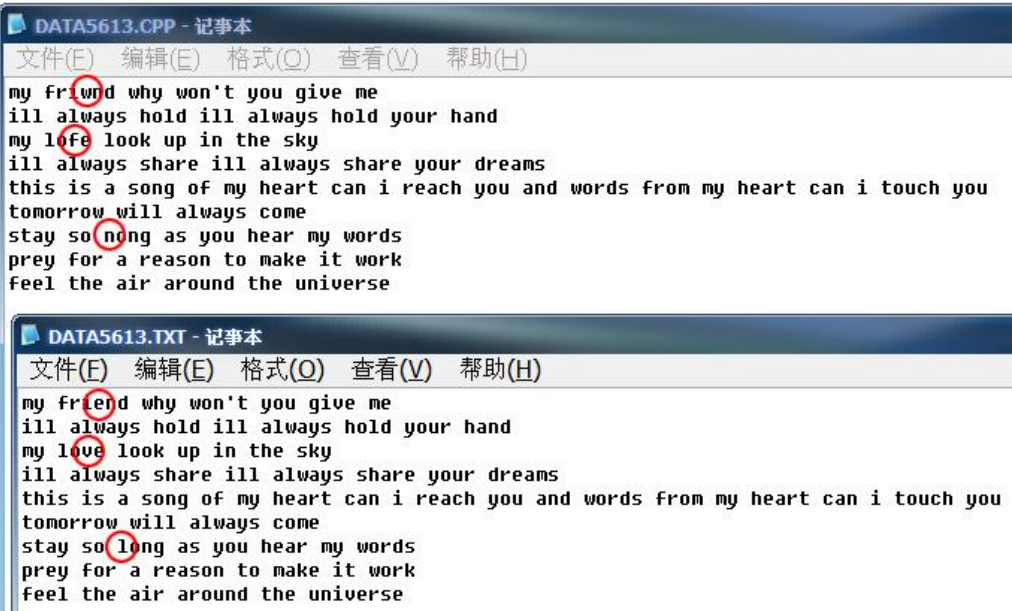
## 第 7 季：复杂数据

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 ( $0 \leq n \leq 1000010000$ )
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"aabbccdddeeeeffffff"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	aabbccdddeeeeffffff e
sample_output	eeee
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"iffordwhileelsewhilebreak"、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	iffordwhileelsewhilebreak 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	<p>定义下面结构表示复数：</p> <pre>typedef struct complex {     double r; /* 实部 */     double i; /* 虚部 */ } COMPLEX;</pre> <p>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。</p>
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	<p>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？</p>
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	<p>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。</p> <p>提交程序时无需提交 DATA5610.TXT 文件。</p>
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	<p>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（<a href="http://www.wotsit.org/">http://www.wotsit.org/</a>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。</p>

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	<p>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：</p>  <p>行号为 4 位，其后为一个空格，然后是程序行。</p>
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

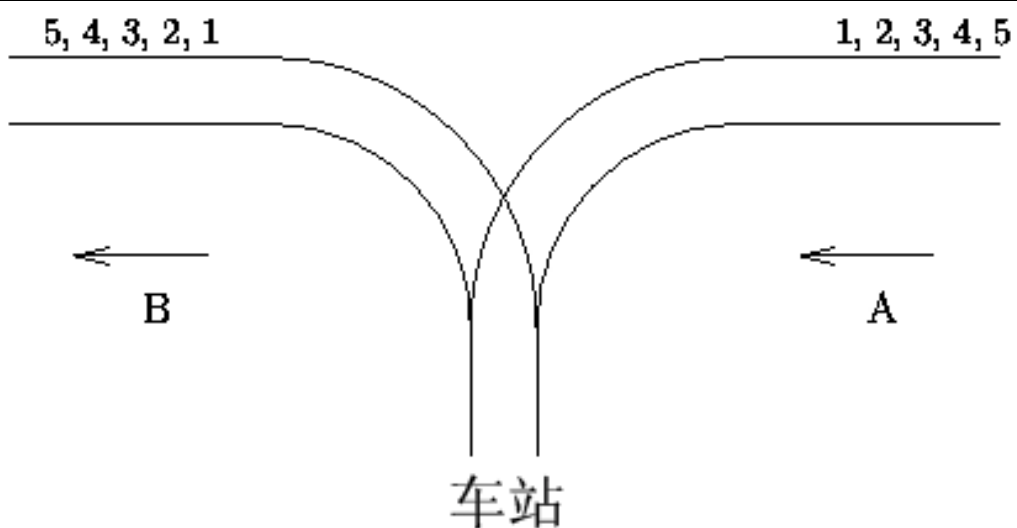
id	8000070000
description	In your job at DongDa Management Inc.(yes, it's run by a bunch of clowns), you have just finished writing a program whose output is a list of names in nondescending order by length (so that each name is at least as long as the one preceding it). However, your boss does not like 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.



[illegible]

	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	333333333333 222222222222
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	666666666666666666666666666666666666 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 $0 \leq n \leq 25$ 。
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	<p>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块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。</p> <p>具体来讲, 字符串的编码规则如下:</p> <p>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。</p> <p>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。</p>
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	<p>有一个排序算法 QuickSort。这个算法是将 <math>n</math> 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。</p> <p>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。</p>
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。

	 <p>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 <math>N</math> 节车厢 (<math>N \leq 1000</math>)，分别按顺序编号为 <math>1, 2, \dots, N</math>。负责车厢调度的工作人员需要知道能否使它以 <math>A_1, A_2, \dots, A_n</math> 的顺序从 B 方向驶出。</p> <p>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。</p>
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 ( $A_1, A_2, \dots$ )，以 0 为结束标志，用空格分隔。
output	输出 Yes 表示可以把火车 ( $1, 2, \dots, N$ ) 火车编排成所需要的顺序 ( $A_1, A_2, \dots$ )，否则用 No 表示。
sample_input	5 1 2 3 4 5 0
sample_output	Yes
id	8000070013
description	<p>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。</p> <p>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：</p> <ol style="list-style-type: none"> <li>(1) 将队列中的第一个打印工作 J 从队列中取出；</li> <li>(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；</li> <li>(3) 否则打印 J (不将 J 移到队列后端)；</li> </ol> <p>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。</p> <p>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。</p>
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 ( $1 \leq n \leq 100$ )， $m$ 是你的打

	<p>印工作的位置 (<math>0 \leq m \leq n-1</math>)。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。</p> <p>第 2 行给出 <math>n</math> 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。</p>
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	<p>游乐园准备抽奖选择一批幸运儿童 (<math>X</math> 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 <math>N</math>；从队列中由 1 到 <math>N</math> 进行报数，每次报到 <math>N</math> 时，第 <math>N</math> 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 <math>X</math> 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：</p> <p>(1) 队列 1、2、3、4、5、6、7、8、9、10，<math>N=10</math>，<math>X=2</math>，卡片次序为 3、5、4、3、2、……</p> <p>(2) 3：划掉 3、6、9，剩下 1、2、4、5、7、8、10；</p> <p>(3) 5：划掉 7，剩下 1、2、4、5、8、10；</p> <p>(4) 4：划掉 5，剩下 1、2、4、8、10；</p> <p>(5) 3：划掉 4，剩下 1、2、8、10；</p> <p>(6) 2：划掉 2、10，剩下 1、8；</p>
input	<p>第 1 行先输入一个整数 <math>N</math> (<math>1 \leq N \leq 50</math>) 给出参加抽奖的儿童人数，再输入一个整数 <math>X</math> (<math>1 \leq X \leq 9</math>) 给出有多少个幸运位置。</p> <p>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。</p>
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

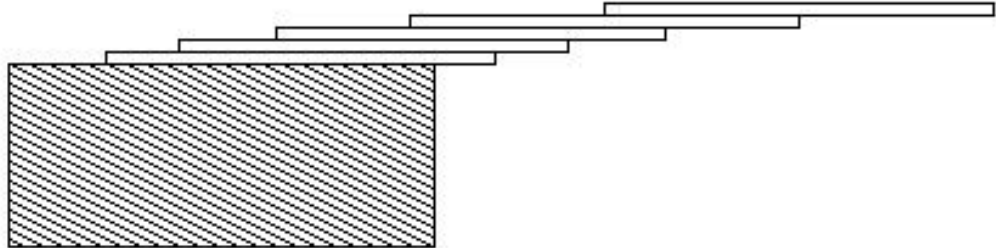
id	8000075000
description	<p>When Tom was a child , he was always thinking about some simple math problems ,such as “What it’s 1 cup of water plus 1 pile of dough ..” , “100 yuan buy 100 pig” .etc..</p> <p>One day Tom met a old man in his dream , in that dream the man whose name was “RuLai” gave Tom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

input	each line contain an integer N ( $0 \leq N \leq 1000000000000$ ).
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 $R_i$ 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 $R_i$ ( $0 < R_i < 15$ ) 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 $R_i$ ， $R_i$ 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 $w, n$ ， $80 \leq w \leq 200, 1 \leq n \leq 300$ ， $w$ 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 <i>abba</i> 和 <i>abbabba</i> 。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	<p>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？</p> <p>如图为顺时针给出</p>  <p>如图为逆时针给出</p> 
input	输入 6 个整数 $x_1, y_1, x_2, y_2, x_3, y_3$ 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	<p>设计一个程序求出 <math>A*B</math>，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：</p> <p><math>6*8=48</math></p> <p><math>4+8=12</math></p> <p><math>1+2=3</math></p> <p>输出 3 即可。</p>
input	输入给出两个非负整数 m, n ( $0 \leq m, n \leq 10000000000000$ )。
output	输出 k。
sample_input	1234567 67
sample_output	4
id	8000075011
description	<p>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 <math>(x,y), (x+1,y), \dots, (x+n-1,y)</math>。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。</p> <p>计算使所有士兵排成一行需要的最少移动步数。</p>
input	第 1 行输入士兵数 n, $1 \leq n \leq 10000$ 。接下来 n 行是士兵的初始位置，每行 2 个整数 x 和 y, $-10000 \leq x, y \leq 10000$ 。
output	输出士兵排成一行需要的最少移动步数。
sample_input	5 1 2



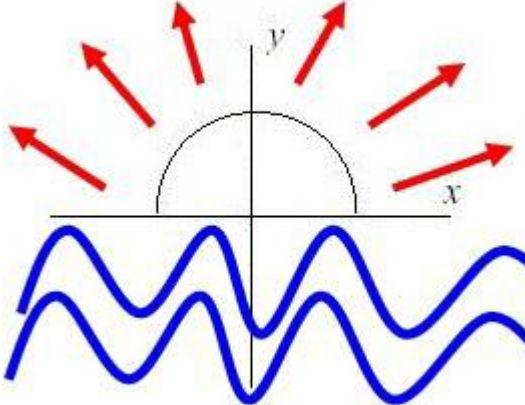
	2 2 1 3 3 -2 3 3
sample_output	8
id	8000075012
description	<p>你能使一叠在桌子上的卡片向外伸出多远？</p> <p>如果你有一个卡片，这张卡片最多可以向桌子外伸出卡片的一半长度（假设该卡片必须垂直于桌子），如果有两个卡片，就可以让上面的卡片向外伸出下面那张卡片的一半长度，而下面那张卡伸出桌子卡片的三分之一长度，所以两张卡片总的向外延伸 <math>1/2 + 1/3 = 5/6</math> 卡长度。以此类推，N 张卡片向外延伸 <math>1/2 + 1/3 + 1/4 + \dots + 1/(n+1)</math> 卡片长度，最上面的向外延伸 <math>1/2</math>，第二张向外延伸 <math>1/3</math>，第三张向外延伸 <math>1/4</math>，<math>\dots</math>，最下面一张向外延伸 <math>1/(n+1)</math>。如下图所示。</p> 
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	<p>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.</p> <p>Consider the following algorithm:</p> <ol style="list-style-type: none"> <li>1.       input n</li> <li>2.       print n</li> <li>3.       if n = 1 then STOP</li> <li>4.             if n is odd then n &lt;- 3n + 1</li> <li>5.             else n &lt;- n / 2</li> <li>6.       GOTO 2</li> </ol> <p>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</p> <p>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 <math>0 &lt; n &lt; 1,000,000</math> (and, in fact, for many more numbers than this.)</p> <p>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.</p> <p>For any two numbers i and j you are to determine the maximum cycle length over all</p>

	numbers between i and j.
input	<p>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.</p> <p>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.</p> <p>You can assume that no operation overflows a 32-bit integer.</p>
output	<p>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).</p>
sample_input	1 10
sample_output	1 10 20
id	8000075014
description	<p>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:</p> <p>Gainers 1498 Losers 902</p> <p>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 <math>1.666667 - 1.660754 = 0.005913</math>. 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.</p> <p>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 <math>\{a_1, a_2, \dots, a_n\}</math> where <math>a_1</math> is a rational number with denominator 1 that most exactly matches the true ratio of winners to losers (rounding up in case of ties), <math>a_{i+1}</math> is the rational number with least denominator that provides a more accurate approximation than <math>a_i</math>, and <math>a_n</math> 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.</p> <p>For example, if 5 stocks rose in price and 4 fell, the best approximation with denominator 1 is <math>1/1</math>; 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 <math>2/2</math> and <math>3/2</math>, but neither is an improvement on the ratio <math>1/1</math>, so neither would be considered. The best approximation with three in the denominator <math>4/3</math>, is more accurate than any seen so far, so it is one that should be reported. Finally, of course, <math>5/4</math> is exactly the ratio, and so it is the last number reported in the sequence.</p> <p>Can you automate this process and help the anchorpeople?</p>

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	<p>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.</p> <p>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.)</p> 
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	<p>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.</p> <p>For example, the number 2991 has the sum of (decimal) digits <math>2+9+9+1 = 21</math>. Since <math>2991 = 1*1728 + 8*144 + 9*12 + 3</math>, its duodecimal representation is 189312, and these digits also sum up to 21. But in hexadecimal 2991 is BAF16, and <math>11+10+15 = 36</math>, so 2991 should be rejected by your program.</p> <p>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.)</p>
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

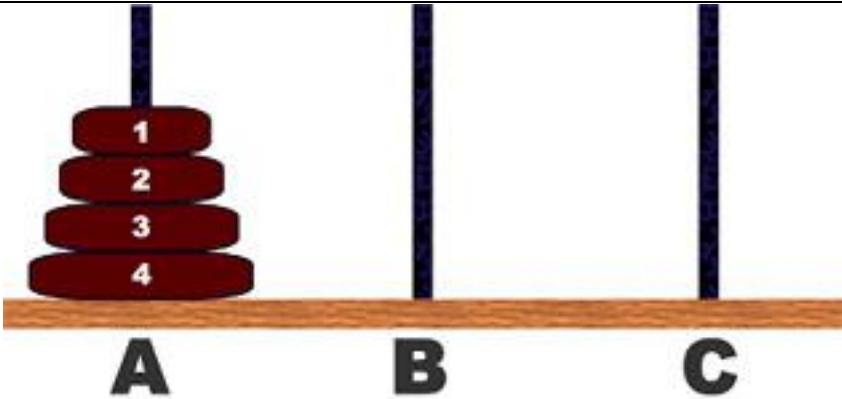
id	8000005004
description	<p>A checksum is an algorithm that scans a packet of data and returns a single number. The idea is that if the packet is changed, the checksum will also change, so checksums are often used for detecting transmission errors, validating document contents, and in many other situations where it is necessary to detect undesirable changes in data.</p> <p>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.</p> <p>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":</p> <p>ACM: <math>1*1 + 2*3 + 3*13 = 46</math></p> <p>MID CENTRAL: <math>1*13 + 2*9 + 3*4 + 4*0 + 5*3 + 6*5 + 7*14 + 8*20 + 9*18 + 10*1 + 11*12 = 650</math></p>
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.

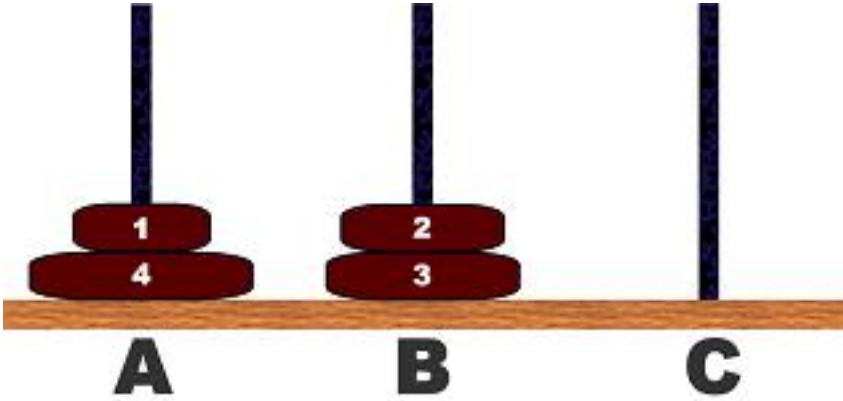
sample_input	ACM MID CENTRAL REGIONAL PROGRAMMING CONTEST ACN A C M ABC BBC #
sample_output	46 650 4690 49 75 14 15
id	8000005005
description	<p>Judging a programming contest is hard work, with demanding contestants, tedious decisions, and monotonous work. Not to mention the nutritional problems of spending 12 hours with only donuts, pizza, and soda for food. Still, it can be a lot of fun.</p> <p>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.</p> <p>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.</p> <p>Scoring</p> <p>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.</p> <p>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.</p> <p>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.</p>
input	<p>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.</p> <p>Line 1 &lt; nTeams &gt;</p> <p>Line 2 - n+1 &lt; Name &gt; &lt; p1Sub &gt; &lt; p1Time &gt; &lt; p2Sub &gt; &lt; p2Time &gt; ... &lt; p4Time &gt;</p> <p>The first element on the line is the team name, which contains no whitespace. Following</p>

	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	<p>If <math>a</math> and <math>d</math> are relatively prime positive integers, the arithmetic sequence beginning with <math>a</math> and increasing by <math>d</math>, i.e., <math>a, a + d, a + 2d, a + 3d, a + 4d, \dots</math>, 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.</p> <p>For example, the arithmetic sequence beginning with 2 and increasing by 3, i.e.,</p> <p style="text-align: center;">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, ... ,</p> <p>contains infinitely many prime numbers</p> <p style="text-align: center;">2, 5, 11, 17, 23, 29, 41, 47, 53, 59, 71, 83, 89, ... .</p> <p>Your mission, should you decide to accept it, is to write a program to find the <math>n</math>th prime number in this arithmetic sequence for given positive integers <math>a</math>, <math>d</math>, and <math>n</math>.</p>
input	<p>The input is a sequence of datasets. A dataset is a line containing three positive integers <math>a</math>, <math>d</math>, and <math>n</math> separated by a space. <math>a</math> and <math>d</math> are relatively prime. You may assume <math>a \leq 9307</math>, <math>d \leq 346</math>, and <math>n \leq 210</math>.</p> <p>The end of the input is indicated by a line containing three zeros separated by a space. It is not a dataset.</p>
output	<p>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.</p> <p>The output integer corresponding to a dataset <math>a, d, n</math> should be the <math>n</math>th prime number among those contained in the arithmetic sequence beginning with <math>a</math> and increasing by <math>d</math>.</p> <p>FYI, it is known that the result is always less than 1000000 (one million) under this input condition.</p>
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	<p>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.</p> <p>For example, if their log shows</p> <table border="1"> <thead> <tr> <th>Speed in miles perhour</th><th>Total elapsed time in hours</th></tr> </thead> <tbody> <tr> <td>20</td><td>2</td></tr> <tr> <td>30</td><td>6</td></tr> <tr> <td>10</td><td>7</td></tr> </tbody> </table> <p>this means they drove 2 hours at 20 miles per hour, then <math>6-2=4</math> hours at 30 miles per hour, then <math>7-6=1</math> hour at 10 miles per hour. The distance driven is then <math>(2)(20) + (4)(30) + (1)(10) = 40 + 120 + 10 = 170</math> miles. Note that the total elapsed time is always since the beginning of the trip, not since the previous entry in their log.</p>	Speed in miles perhour	Total elapsed time in hours	20	2	30	6	10	7
Speed in miles perhour	Total elapsed time in hours								
20	2								
30	6								
10	7								
input	The input consists of one or more data sets. Each set starts with a line containing an integer $n$ , $1 \leq n \leq 10$ , 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, $1 \leq s \leq 90$ and $1 \leq t \leq 12$ . 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	<p>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!</p> <p>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 <math>N</math> years. Please write a program that helps them find the number of male bees and the total number of all bees after <math>N</math> years.</p>								
input	Each line of input contains an integer $N$ ( $\geq 0$ ). 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	 <p>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.</p> <p>Here we have made your task little bit difficult by making the problem more flexible. Here</p>

	<p>the disks can be in any peg initially.</p>  <p>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.</p>
input	<p>The input file contains at most 100 test cases. Each test case starts with a positive integer <math>N</math> (<math>1 \leq N \leq 60</math>), which means the number of disks. You will be given the arrangements in next two lines. Each arrangement will be represented by <math>N</math> integers, which are 1, 2 or 3. If the <math>i</math>-th (<math>1 \leq i \leq N</math>) integer is 1, you should consider that <math>i</math>-th disk is on Peg-A. The end of the input is indicated by a zero.. This case should not be processed.</p>
output	<p>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.</p>
sample_input	<pre>3 1 1 1 2 2 2 3 1 2 3 3 2 1 0</pre>
sample_output	<pre>7 3</pre>
id	8000005011
description	<p>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.</p> <p>There will be a button, when it will be pushed a random number <math>N</math> will be chosen by computer. Then on screen there will be numbers from 1 to <math>N</math>. 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.</p> <p>For example if <math>N = 3</math>, 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</p>

	<p>she subtracts 2 from 2 and all the numbers become 0.</p> <p>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.</p>
input	Input consists of several lines each with N such that $1 \leq N \leq 1,000,000,000$ . 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	<p>HanBo Favorite number is 1, So he will often use some ways to put some numbers to 1, and to this endless pride.</p> <p>He would only two ways will be:</p> <p>(1) Put a number m divided by a prime number p . of course, this number must be divisible, i.e <math>m = m / p</math></p> <p>(2) The number m of one minus 1, i.e <math>m = m - 1</math></p> <p>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.</p>
input	<p>Input contains multiple sets of data (1000 data), the end of the input is indicated by 0 0.</p> <p>Each set of data begins with two integers: a, b (<math>0 &lt; a \leq b \leq 100000</math>), meaning as the meaning of problems described.</p>
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	<p>A Communist regime is trying to redistribute wealth in a village. They 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.</p>

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	<p>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.</p> <p>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.</p> <p>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!</p>
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 ( $2 \leq n \leq 1000$ , $1 \leq m \leq 1000$ ). The length of the alley along the park perimeter is exactly 10 000 feet. The 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