1. **高精度整数相减**

**（sub.cpp）**

【问题描述】

输入2个正的整数，输出 |a-b| 的值。

输入：两行，第一行a，第二行b，a和b的长度均小于1000位。

输出：一行，|a-b| 的值。

【样例】

输入：29

48

输出：19

1. **明明的随机数**

**（random./cpp）**

【问题描述】  
     明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了N个1到1000之间的随机整数（N≤100），对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

【输入格式】  
输入有2行，第1行为1个正整数，表示所生成的随机数的个数：  
N  
第2行有N个用空格隔开的正整数，为所产生的随机数。

【输出格式】  
输出也是2行，第1行为1个正整数M，表示不相同的随机数的个数。第2行为M个用空格隔开的正整数，为从小到大排好序的不相同的随机数。

【输入样例】  
10  
20 40 32 67 40 20 89 300 400 15

【输出样例】  
8  
15 20 32 40 67 89 300 400

1. **汉诺塔问题**

（hanoi./cpp）

【题目描述】

约19世纪末，在欧州的商店中出售一种智力玩具，在一块铜板上有三根杆，最左边的杆上自上而下、由小到大顺序串着由64个圆盘构成的塔。目的是将最左边杆上的盘全部移到中间的杆上，条件是一次只能移动一个盘，且不允许大盘放在小盘的上面。

这是一个著名的问题，几乎所有的教材上都有这个问题。由于条件是一次只能移动一个盘，且不允许大盘放在小盘上面，所以64个盘的移动次数是：18,446,744,073,709,551,615

这是一个天文数字，若每一微秒可能计算(并不输出)一次移动，那么也需要几乎一百万年。我们仅能找出问题的解决方法并解决较小N值时的汉诺塔，但很难用计算机解决64层的汉诺塔。

假定圆盘从小到大编号为1, 2, ...

【输入】

输入为一个整数(小于20）后面跟三个单字符字符串。

整数为盘子的数目，后三个字符表示三个杆子的编号。

【输出】

输出每一步移动盘子的记录。一次移动一行。

每次移动的记录为例如 a->3->b 的形式，即把编号为3的盘子从a杆移至b杆。

【输入样例】

2 a b c

【输出样例】

a->1->c

a->2->b

c->1->b