

2014 年南海区青少年信息学奥林匹克竞赛试题

(初中组)

注意事项:

1. 本次竞赛将使用评测系统进行自动评测,以源程序的测试结果为准。
2. 要利用文件严格按题目要求的格式(参考输入输出样例)进行输入输出,否则将被判为错误而不得分。
3. 所有文件要按试卷指定的文件名命名,程序中不要使用绝对路径,否则将无法测试而被判为 0 分。
4. 比赛结束前,要将最完善的源程序集中保存到以选手考号(字母用大写)加姓名命名的文件夹中(例如: CJ372 陈才),该文件夹中不能再设子文件夹;同时还要在选手程序文件夹中建一个文本文件“<姓名>程序清单.txt”,文件内容写清上交的程序文件名。
5. 现场编程 3 小时,每测试点时间限制 1 秒、内存限制 128M,每题 50 分,总分 300 分。
6. 各题程序及数据文件名规定:

题 号	第 1 题	第 2 题	第 3 题	第 4 题	第 5 题	第 6 题
试题名称	K 好数	睡眠	字符串	猜数	序列	树
程序文件名	knumber. cpp/pas	sleep.cpp /pas	string. cpp/ pas	guess.cpp /pas	sequence.cp p/pas	tree.cpp/ pas
输入文件名	knumber. in	sleep.in	string. in	guess.in	sequence.in	tree.in
输出文件名	knumber. out	sleep.out	string. out	guess.out	sequence.ou t	tree.out

1. K 好数

(knumber.cpp/pas)

【问题描述】

当且仅当一个数的每一位都不超过 k 时,称这个数是“ k 好数”。

给定 n 和 k , 输出 1 到 n 中有多少个数是“ k 好数”。

【输入】

输入只有一行, 包含 2 个用空格隔开的整数 n 和 k 。

【输出】

输出只有一行, 包含 1 个整数, 表示 1 到 n 中“ k 好数”的个数。

【输入输出样例】

knumber.in	knumber.out
25 3	11

【输入输出样例说明】

1 到 25 中 11 个“ k 好数”分别为 1, 2, 3, 10, 11, 12, 13, 20, 21, 22, 23。

【数据范围】

对于 50% 的数据, $1 \leq n \leq 1000$ 。

对于 100% 的数据, $1 \leq n \leq 1000000$, $1 \leq k \leq 9$ 。

2.睡眠

(sleep.cpp/pas)

【问题描述】

小 G 醒来，从电子表上看到现在的时间是 s ，并且他知道自己已经睡了时间 t 。

给定时间 s 和 t ，帮助小 G 算出他开始进入睡眠的时间 p 。

注意：小 G 可能在前一天进入睡眠。

【输入】

第一行包含现在的时间 s ，形式为“hh:mm”的字符串。

第二行包含时间 t ，形式为“hh:mm”的字符串，表示小已经 G 睡了多久。

输入保证都是正确的 24 小时制的时间，也就是 $00 \leq hh \leq 23$ ， $00 \leq mm \leq 59$ 。

【输出】

输出只有一行，包含一个字符串 p ，形式为“hh:mm”，表示小 G 开始进入睡眠的时间。

【输入输出样例 1】

sleep.in	sleep.out
05:50	00:06
05:44	

【输入输出样例 2】

sleep.in	sleep.out
00:00	23:00
01:00	

【数据范围】

对于 100% 的数据， $00 \leq hh \leq 23$ ， $00 \leq mm \leq 59$ 。

3. 字符串

(string.cpp/pas)

【问题描述】

小熊有一个由小写英文字母组成的字符串 $s = s_1s_2...s_n$ 。小熊想要计算 s 中有多少子串包含字符串“bear”，也就是找出满足字符串 $x(i, j) = s_is_{i+1}...s_j$ 包含至少一个字符串“bear”的 (i, j) 对数 ($1 \leq i \leq j \leq n$)。

字符串 $x(i, j)$ 包含字符串“bear”定义为存在一个整数 k ($i \leq k \leq j-3$)，满足 $s_k = b$ ， $s_{k+1} = e$ ， $s_{k+2} = a$ ， $s_{k+3} = r$ 。

请帮助小熊解决这个问题。

【输入】

输入共 1 行，包含一个非空字符串 s 。数据保证字符串 s 中只包含小写英文字母。

【输出】

输出共 1 行，包含一个整数，表示这个问题的答案。

【输入输出样例】

string.in	string.out
bebearar	9

【输入输出样例解释】

符合条件的 9 对 (i, j) 为：(1, 6), (1, 7), (1, 8), (2, 6), (2, 7), (2, 8), (3, 6), (3, 7), (3, 8)。

【数据说明】

对于 50% 的数据， $1 \leq n \leq 200$ 。

对于 100% 的数据， $1 \leq n \leq 3000$ 。

4.猜数

(guess.cpp/pas)

【问题描述】

猜数的规则非常简单。主持人会在心中想一个数 y ($-10000 \leq y \leq 10000$)，然后参与者通过询问主持人来猜这个数。下面是 4 种可接受的询问。

第 1 种询问： y 是**严格大于** x 吗？

第 2 种询问： y 是**严格小于** x 吗？

第 3 种询问： y 是**大于或等于** x 吗？

第 4 种询问： y 是**小于或等于** x 吗？

对于每个询问，主持人都会如实地回答“是”或者“否”。

给定一系列的询问和答案，请你找到满足所有答案的**最小的** y 。如果不存在这样的 y ，输出“Impossible”（不含引号）。

【输入】

第一行包含一个整数 n ，表示询问和答案的数量。

接下来 n 行，每行包含一个询问和一个答案，形式为“sign x answer”。

“sign”有 4 种可能，分别是“>”（第 1 种询问），“<”（第 2 种询问），“>=”（第 3 种询问），“<=”（第 4 种询问）。

“answer”有 2 种可能，分别是“Y”（表示“是”），“N”（表示“否”）。

所有询问中的 x 均满足 $-10000 \leq x \leq 10000$ 。

【输出】

输出文件名为 guess.out。

如果存在满足所有答案的 y ，那么输出一个最小的 y ，否则输出“Impossible”（不含引号）。

【输入输出样例 1】

guess.in	guess.out
4 >= 1 Y < 3 N <= -3 N > 55 N	3
【输入输出样例 2】 guess.in	guess.out
2 > 100 Y < -100 Y	Impossible

【数据范围】

对于 50% 的数据， $1 \leq n \leq 1000$ 。

对于 100% 的数据， $1 \leq n \leq 100000$ 。

5. 序列

(sequence.cpp/pas)

【问题描述】

一个长度为 k 的整数序列 b_1, b_2, \dots, b_k ($1 \leq b_1 \leq b_2 \leq \dots \leq b_k \leq N$) 称为“好序列”当且仅当后一

个数是前一个数的倍数，即 b_{i+1} 是 b_i 的倍数对任意的 i ($1 \leq i \leq k-1$) 成立。

给定 N 和 k ，请算出有多少个长度为 k 的“好序列”，**答案对 1000000007 取模**。

【输入】

输入共 1 行，包含 2 个用空格隔开的整数 N 和 k 。

【输出】

输出共 1 行，包含一个整数，表示长度为 k 的“好序列”的个数对 1000000007 取模后的结果。

【输入输出样例】

sequence.in	sequence.out
3 2	5

【输入输出样例解释】

“好序列”为：[1,1], [1,2], [1,3], [2,2], [3,3]。

【数据说明】

对于 40% 的数据， $1 \leq N \leq 30$ ， $1 \leq k \leq 10$ 。

对于 100% 的数据， $1 \leq N \leq 2000$ ， $1 \leq k \leq 2000$ 。

6. 树

(tree.cpp/pas)

【问题描述】

小 L 非常喜欢树。最近，他发现了一棵有趣的树。这棵树有 n 个节点(1 到 n 编号)，节点 i 有一个初始的权值 a_i 。**这棵树的根是节点 1。**

这棵树有一个特殊的性质：当你给节点 i 的权值加 val 的时候，节点 i 的所有儿子的权值都会加 $-val$ 。注意当你给节点 i 的儿子的权值加 $-val$ 时，节点 i 的**这个儿子的所有儿子**的权值都会加 $-(-val)$ ，**以此类推**。样例说明可以很好地帮助你理解这个性质。

有 2 种操作：

操作(a). “1 x val” 表示给节点 x 的权值加 val 。

操作(b). “2 x” 输出节点 **x 当前的权值**。

为了帮助小 L 更好地理解这棵树，你必须处理 m 个操作。

【输入】

第一行包含 2 个整数 n 和 m 。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$)。

接下来的 $n-1$ 行，每行两个整数 u 和 v ($1 \leq u < v \leq n$)，表示节点 u 和节点 v 之间存在一条边。

接下来的 m 行，每行包含 2 种操作的一种。每个操作都保证 $1 \leq x \leq n$ ， $1 \leq val \leq 1000$ 。

【输出】

对于每个操作(b)，输出一个整数，表示节点 **x 当前的权值**。

【输入输出样例】

tree.in	tree.out
5 5	3
1 2 1 1 2	3
1 2	0
1 3	
2 4	
2 5	
1 2 3	

1 1 2	
2 1	
2 2	
2 4	

【输入输出样例解释】

初始各个节点的权值依次为[1, 2, 1, 1, 2]。

第一个操作给节点 2 的权值增加 3, 会给节点 2 的儿子 4、5 的权值增加-3。此时各个节点的权值变成[1, 5, 1, -2, -1]。

第二个操作给节点 1 的权值增加 2, 会给节点 1 的儿子 2、3 的权值增加-2, 然后会给节点 2 的儿子 4、5 的权值增加-(-2)。各个节点的权值变成[3, 3, -1, 0, 1]。

【数据说明】

对于 50%的数据, $1 \leq n \leq 2000$, $1 \leq m \leq 2000$ 。

对于 100%的数据, $1 \leq n \leq 100000$, $1 \leq m \leq 100000$ 。