海亮信息学奥赛集训 数学专场

2019.2

一、题目概况

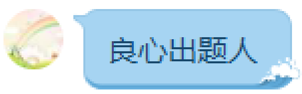
题目名称	亚瑟王	取石子游戏	约数个数和	礼物
题目英文名称	simple	stone	formula	gift
输入文件名	simple.in	stone.in	formula.in	gift.in
输出文件名	simple.out	stone.out	formula.out	gift.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	20	20	10
每个测试点分值	10	5	5	10
结果比较方式	全文比较 忽略行末空格文末回车			实数比较
题目类型	传统型	传统型	传统型	传统型
运行内存上限	256M	256M	256M	256M

二、提交源程序文件名

对于 C 语言	simple.c	stone.c	formula.c	gift.c
对于 C++语言	simple.cpp	stone.cpp	formula.cpp	gift.cpp
对于 pascal 语言	simple.pas	stone.pas	formula.pas	gift.pas

三、注意事项

1. 文件名（程序名和输入输出名）必须用英文小写。
2. C/C++中函数 main 的返回值类型必须为 int，程序正常结束返回值必须是 0。
3. 根据 std 在评测机的运行时间适当放宽时间限制。
4. 祝考试顺利



1. 亚瑟王

(simple.c/cpp/pas)

【问题描述】

亚瑟王掷一枚硬币，概率p正面向上，概率$(1-p)$反面向上。

现在亚瑟王要掷k次正面向上，他想知道期望要投掷多少枚硬币才能掷到k次正面向上。

小 B 觉得这题太简单了，加了一个问题：如果亚瑟王第i次投掷硬币花费为$(2i-1)$，那么期望花费多少才能掷到k次正面向上。不用担心，你只需要对一些特定的测试点回答这个问题即可。

亚瑟王不喜欢小数，因此他会告诉你$p=\frac{a}{b}$，你只需要告诉他答案对(10^9+7)取模即可。

【输入格式】

输入文件名为 simple.in。

第一行，一个数 T，表示**数据类型**。当$T=1$时你只需要回答亚瑟王的问题，当$T=2$的时候你只需要回答小 B 的问题。

第二行，三个数，a,b,k，含义如题所示。

【输出格式】

输出文件名为 simple.out。

仅一行，一个数，要求回答的问题的答案，对(10^9+7)取模。

【输入输出样例 1】

simple.in	simple.out
1 1 2 2	4

【输入输出样例 2】

simple.in	simple.out
2 1 2 2	20

【数据规模与约定】

测试点	数据范围
1-2	$T=1,1\le k\le 10$
3-4	$T=1,1\le k\le 10^3$
5-6	$T=1,1\le k\le 10^6$
7-8	$T=1,1\le k\le 10^9$
9	$T=2,1\le k\le 10$
10	$T=2,1\le k\le 10^9$

对于所有的数据，保证$1\le k,a,b\le 10^9$。

2. 取石子游戏

(stone.c/cpp/pas)

【问题描述】

小 A 拿来了一堆 n 个石子，邀请小 B 一起玩取石子游戏，规则是每人每次可以取不超过 3 个石子，谁先取完谁获胜。

小 B 看了一眼石子的个数，在飞快的心算过后，明白了小 A 存心要坑他，于是小 B 提出了一个新的游戏规则：

每个人每步可以将一堆石子分成 k 堆($k \geq 2$)，由于小 B 非常喜欢等差数列，尤其喜欢公差为 1 的等差数列，故要求这 k 堆石子排成一个公差为 1 的等差数列。换句话说，如果分出来的每堆石子有$a_1, a_2, \dots, a_k$个（不妨$a_1 \leq a_2 \leq \cdots \leq a_k$），那么需要满足：

$$a_2 - a_1 = a_3 - a_2 = \cdots = a_k - a_{k-1} = 1$$

无法操作的玩家则失败。

小 B 为了显示出自己的友善，让小 A 先取石子。小 A 觉得小 B 也别有用心，但是他自己并无法一眼看出这个状态是否有必胜方案，于是他找了你，想知道他是否有必胜策略。

小 A 发现，得知是否有必胜策略还不够，他还需要知道先手第一步该如何操作。由于把石头搬来搬去很累，所以小 A 想知道，若有必胜策略，先手第一步最少分成几堆可以必胜？

【输入格式】

输入文件名为 stone.in。

仅一行，一个数 n，表示初始的一堆石子总数。

【输出格式】

输出文件名为 stone.out。

仅一行，一个数，如果小 A 必败，则输出-1；否则输出所有的小 A 的必胜策略中，第一步最少需要分成几堆。

【输入输出样例】

stone.in	stone.out
3	2

【输入输出样例说明】

小 A 分成两堆后，两堆石子个数分别为 1、2 个，此时后手无法操作，小 A 获胜。

【数据规模与约定】

测试点	数据范围
1-2	$1 \leq n \leq 10$
3-6	$1 \leq n \leq 100$
7-10	$1 \leq n \leq 1000$
11-14	$1 \leq n \leq 10^4$
15-16	$1 \leq n \leq 10^5$
17-20	$1 \leq n \leq 1.2 \times 10^5$

对于所有的数据，保证$1 \leq n \leq 1.2 \times 10^5$。

3. 约数个数和

(formula.c/cpp/pas)

【问题描述】

小 B 做完约数个数和那题后，有一天，突发奇想，出了一道这样的题：

已知a,b,c，求：

$$\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(ijk) \mod 2^{30}$$

其中，$d(x)$表示x的约数个数。

小 A 发现他不会，就把题目扔给了你。他为了方便你的计算，算出了$2^{30}=1073741824$，你只需要告诉他上式的值即可。

【输入格式】

输入文件名为 formula.in。

仅一行，三个数a,b,c，含义如题所示。

【输出格式】

输出文件名为 formula.out。

一行，一个正整数，表示答案。

【输入输出样例】

formula.in	formula.out
2 2 2	20

【数据规模与约定】

测试点	数据范围
1-6	$1 \le a,b,c \le 10$
7-12	$1 \le a,b,c \le 100$
13-20	$1 \le a,b,c \le 2\times10^{3}$

4. 礼物

(gift.c/cpp/pas)

【问题描述】

小 A 打算送小 B 一些礼物。

小 A 来到了随机商店，这个商店有 m 种礼物，且第 i 种礼物有k_i个。**同种礼物的价值各不相同不同，但是同种礼物的名字相同。**这个商店的购买物品的方式也非常特别，需要你提供一个礼物清单，然后商店会根据你的礼物清单来按照一定规则挑选物品。

小 A 想买 n 个礼物，他列了一个礼物清单，只包括**这些礼物的名字**，如果第 i 种礼物小 A 要买t_i个，那么他就要写t_i个第 i 种礼物的名字在礼物清单上，显然应该满足$t_i \le k_i$。

商店的挑选规则如下：假设清单上写了 p 种礼物，分别是第$a_1, a_2, \dots, a_p$种，第a_i种礼物需要t_i个，那么就从第a_i种礼物的全部k_{a_i}个种**随机挑选**t_i个礼物给顾客。

由于小 A 是欧洲人抽中了免单资格，因此他想让他的所有礼物的价值和最大。因此他会制定一个**有可能得到礼物的最大价值总和**的方案，如果有 s 种可以达到最大价值的方案，他会**随机选择**一种。求：小 A 得到最大价值的概率。

【输入格式】

输入文件名为 gift.in。

第一行，两个数，n 和 m，含义如题目所示。

接下来 m 行，每行表示一种礼物。每行第一个数为k_i，表示第 i 种礼物的个数。接下来有k_i个数，表示每个礼物的价值。

【输出格式】

输出文件名为 gift.out。

一行，一个浮点数，表示答案，你的答案与标准答案之差不超过10^{-9}时正确。

【输入输出样例 1】

gift1.in	gift1.out
3 1 3 10 20 30	1.000000000000

【输入输出样例 2】

gift1.in	gift2.out
3 2 1 40 4 10 20 30 40	0.166666666667

【输入输出样例 1 说明】

显然小 A 可以选择所有礼物，获得最大价值。

【输入输出样例 2 说明】

小 A 期望获得的最大价值为 40+40+30，因此他清单上需要选择第 1 种礼物 1 个，第 2 种礼物 2 个。这样他获得最大价值的概率为$1/6 = 0.166 \dots \dots$。

【数据规模与约定】

测试点	数据范围
1-3	$1 \le n, m \le 10, \sum_{i=1}^{m} k_i \le 100$
4-6	$1 \le n, m \le 100, \sum_{i=1}^{m} k_i \le 1000$

7-10	$1 \le n, m \le 10^3, \sum_{i=1}^{m} k_i \le 3000$

对于所有数据，$1 \le$ 价值 $\le 10^9, \sum_{i=1}^{m} k_i \le 1000$， 不存在价值和名称均相同的两个物品。