

第十八届“快手杯” 北京航空航天大学程序设计竞赛 决赛

2023 年 12 月 17 日



BCPC 2023 The 18th Beihang University
Collegiate Programming Contest

题目概况

题号	题目名	时间限制	空间限制
A	rnm, 退钱!	1 s	2048 MB
B	奥术之尘	1 s	2048 MB
C	更小但是异或之后至少有 x 个 1	1 s	2048 MB
D	生活在树上	1 s	2048 MB
E	宝石合成	1 s	2048 MB
F	二分查找	1 s	2048 MB
G	What if ...?	1 s	2048 MB
H	Rolling Star	2 s	2048 MB
I	请问您今天要来点取模吗?	2 s	2048 MB



Problem A. rnm, 退钱!

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 256 megabytes

暴雪带着他的全家桶跑路了!

在跑路以后, 大冤种网易选择主动承担起国服玩家的部分损失, 具体来说, 规则如下:

对于游戏内未花完的战网点, 将按照原本的价格退款。

已知战网点和 RMB 的比例是 1 : 1。

现在小 A 找到了他的账单记录, 里面完整的记录着他的每一笔消费, 包括使用 RMB 充值和在游戏内花费战网点的具体情况。

他想请你帮他算一算, 他到底能退多少钱?

输入格式

第一行包含一个整数 n ($1 \leq n \leq 1000$), 表示记录次数。

之后的 n 行, 每行一个整数 x_i ($-10^9 \leq x_i \leq 10^9$), 如果 $x_i > 0$, 则表示这一条记录你用 RMB 购入了 x_i 的战网点。如果 $x_i < 0$, 则表示你在游戏内使用了 $-x_i$ 的战网点。如果 $x_i = 0$, 则表示这一条记录和退款无关。

可以保证, 任意时刻, 如果 $x_i < 0$, 他此时一定有这么多战网点供你消费。

输出格式

一个整数, 表示他能退回的钱数。

样例

standard input	standard output
6	328
328	
-328	
488	
-488	
0	
328	

提示

他购买了一次 328 战网点并用于预购; 又购买了 488 战网点并用于豪华预购。这两笔钱由于已经在游戏内用掉了, 所以无法退钱。

中间有一次免费领取旧冒险的操作。该操作与退款无关。

但是最后一笔 328 还没来得及用来预购就得知了暴雪跑路的消息, 所以可以退款。

Problem B. 奥术之尘

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

随着网易与暴雪宣布分手，暴雪全家桶的生命正式进入倒计时。

你此时发现，随着充值入口的关闭，还想玩特定卡组的人就只能通过购买“卡组号”的方式来游玩。一般来说，一个账号内包含的卡组越多，这个账号价格就越高。

而市场上还有一些账号被称为“金币号”，这些账号里面包含了大量金币，你之前低价购买了 n 个金币号用来开卡包玩，开完以后你将里面所有的卡转化成了“奥术之尘”。

你发现了商机以后，你决心将这些“金币号”转变为“卡组号”卖出去。

现在有 m 种主流卡组，每个卡组的合成都需要一定的“奥术之尘”；对于每一个账号，给定你拥有的“奥术之尘”数量，求你能最多合出多少个卡组？

在本题中，我们假设多个卡组不会共享同一张卡，也就是合任意的卡组的需求为这些卡组奥术之尘的和。

输入格式

第一行，一个整数 n ($1 \leq n \leq 10^5$)，表示你的金币号数量。

第二行， n 个整数 A_i ($0 \leq A_i \leq 10^9$)，表示每个金币号内，“奥术之尘”的数量。

第三行，一个整数 m ($1 \leq m \leq 10^5$)，表示目前主流卡组的数量。

第四行， m 个整数 B_i ($0 \leq B_i \leq 10^9$)，表示每个主流卡组所需要的“奥术之尘”。

输出格式

一行，总共 n 个整数，表示对于每个金币号，你能合成的卡组数量。

样例

standard input	standard output
3 12000 15000 16000 6 0 48000 2240 3800 9360 3120	4 4 4

提示

对于第一套赠送卡组，所有号都可以免费领取。

对于第二套卡组，所有号都不考虑合成。

第一个号可以合成 $0 + 2240 + 3120 + 3800$ 四套卡组。

第二个号可以合成 $0 + 2240 + 9360 + 3120$ 四套卡组。

第三个号可以合成 $0 + 3800 + 9360 + 2240$ 四套卡组。

Problem C. 更小但是异或之后至少有 x 个 1

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

由于给预赛题编题面已经耗光了 Nerovix 的灵感，他决定让这道题的题面简单一点。

给你一个以二进制形式表示的大整数 a 和一个非负整数 x ，求最大的 b 满足 $b \leq a$ 且 $\text{popcount}(a \oplus b) \geq x$ 。其中， $\text{popcount}(n)$ 表示自然数 n 在二进制表示下 1 的个数， $\oplus$ 表示按位异或。

输入格式

第一行包含一个以二进制形式表示的大整数 a ($0 \leq a < 2^{2 \times 10^6}$ ，不含前导 0)。

第二行包含一个非负整数 x ($0 \leq x < 2 \times 10^6$)。

输出格式

如果有解，输出一行包含一个以二进制形式表示的大整数 b （不含前导 0）；否则，输出 -1 。

样例

standard input	standard output
10101001 7	1110110
10101001 0	10101001
10010 6	-1

Problem D. 生活在树上

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

决赛题目以 Stella 的一句“一切简单题目都已经被出完了”为嚆矢。

给定 n, d , 试求一棵 n 个点的树与树上的一个点 k , 使得 k 到树上各点的距离之和恰为 d 。

树上两点 u, v 之间的距离定义为以 u, v 为端点的最短路径的边数。

如果存在多组满足条件的答案, 输出任意一种均可。

输入格式

一行, 两个整数 n, d ($1 \leq n \leq 2 \times 10^5$, $0 \leq d \leq 2 \times 10^{10}$), 含义见上。

输出格式

如果存在满足条件的答案, 第一行输出 **YES**。接下来 $n-1$ 行, 每行包含两个整数 u, v ($1 \leq u, v \leq n$), 表示树上编号为 u, v 的两点之间的边。最后一行包含一个整数 k ($1 \leq k \leq n$), 为树上指定点的编号。

如果不存在满足条件的答案, 输出 **NO**。

样例

standard input	standard output
4 5	YES 1 2 2 3 2 4 1
3 5	NO

Problem E. 宝石合成

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

问号同学有一个长度为 n 的宝石序列，其中第 i 颗宝石的等级为 a_i 。他每次可以把连续的两个及以上相同等级的宝石合成为一个高一级的宝石。他想知道能不能将这个宝石序列合成为一个宝石。

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 10^5$)，表示数据组数。

对于每组数据：

第一行两个整数 n ($1 \leq n \leq 10^5$)，表示宝石序列的长度。

第二行，共 n 个整数 $a_1, \dots, a_n$ ，表示第 i 颗宝石的等级为 a_i ($1 \leq a_i \leq n$)。

保证所有数据的 n 的和不超过 5×10^5 。

输出格式

对于每组数据：

输出一行 Yes 或 No，表示能不能将这个宝石序列合成为一个宝石。

样例

standard input	standard output
3	Yes
5	Yes
3 2 1 1 3	No
4	
1 1 1 2	
3	
1 1 3	

Problem F. 二分查找

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

问号同学今天刚学习了二分查找算法，他写了一段二分查找代码（代码中的除法为下取整）。

```
int find(int x) {  
    int l = 1, r = n;  
    while (l < r) {  
        int m = (l + r) / 2;  
        if (a[m] < x) l = m + 1;  
        else r = m;  
    }  
    return l;  
}
```

之后，他对一个长度为 n 的排列数组 a （下标从 1 开始）执行了 m 次二分查找，第 i 次二分查找的值为 x_i ，返回结果为 y_i 。注意长度为 n 的排列中 1 到 n 均恰好出现了一次。

但是由于小行星撞击地球，问号同学的 a 数组找不到了，只留下了 m 组 x_i, y_i 的记录。

聪明的你能否帮助问号同学找回 a 数组呢？

输入格式

本题包含多组数据。

第一行包含一个整数 T ($1 \leq T \leq 1024$)，表示数据组数。

对于每组数据：

第一行两个整数 n, m ($n = 2^k, 1 \leq k \leq 16, 1 \leq m \leq n$)，表示序列长和二分查找记录的个数。

接下来 m 行，每行两个整数 x_i, y_i ($1 \leq x_i, y_i \leq n$)，表示二分查找 x_i 的结果为 y_i 。

保证所有数据的 n 的和不超过 2^{18} 。

输出格式

对于每组数据：

如果存在可行解，输出 n 个整数 $a_1, \dots, a_n$ ，表示原来的排列数组 a 。

当有多种可行解时，输出任意一种即可。

如果不存在可行解，输出一行 -1 。

样例

standard input	standard output
3	1 2
2 2	4 3 2 1
1 1	-1
2 2	
4 3	
1 1	
2 1	
3 1	
4 2	
2 3	
4 1	

Problem G. What if ...?

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 256 megabytes

Oshwiciqwq 正在学习编程。他学到了 if 语句，并写下了一个有 m 个条件 $m+1$ 个分支的 if 语句。这个 if 语句在一个循环变量为 i ，范围为 $[1, n]$ 的 for 语句中。

if 语句的第 j 个条件形如 $i \text{ op}_j x_j$ ， op_j 可能为 $=, <, >$ 。

此外，程序中还有一个数组 A ，下标范围为 $[1, m+1]$ 。在第 j 个分支中，只有一条语句：

```
A[j] = A[j] + 1
```

以 C 语言为例，整个 if 语句形如：

```
for (int i = 1; i <= n; i++) {  
    if (i == x[1]) {  
        A[1] = A[1] + 1;  
    } else if (i < x[2]) {  
        A[2] = A[2] + 1;  
    } else {  
        A[3] = A[3] + 1;  
    }  
}
```

他设置了适当的 n ，准备运行这段程序。可是运行的时候世界线发生了变动， x 数组里的所有数变成了 $[1, n]$ 内的随机整数！

他对此很好奇，想知道 A 数组里所有数的期望值，你能告诉他答案吗？

输入格式

第一行两个正整数 n, m ($1 \leq n \leq 10^9, n \neq 998\,244\,353, 1 \leq m \leq 10^3$)，分别表示循环的上界和条件的个数。

第二行有 m 个字符串，第 i 个字符串 op_i 表示第 i 个条件的运算符。保证 op_i 为 $=, <, >$ 中的一个。（注意，虽然在一些语言中“等于”逻辑运算符是 `==`，但是本题的输入为 `=`）

输出格式

输出一行 $m+1$ 个整数，第 i 个数表示 A_i 的期望值，对 998 244 353 取模。

令 $M = 998\,244\,353$ ，可以证明所求期望可写作既约分数 $\frac{p}{q}$ 的形式，其中 p, q 为整数且 $q \not\equiv 0 \pmod{M}$ 。输出的整数应与 $p \cdot q^{-1} \pmod{M}$ 相等，换言之，输出一个整数 x 满足 $0 \leq x < M$ 且 $x \cdot q \equiv p \pmod{M}$ 。

样例

standard input	standard output
10 2 < =	499122181 648858830 848507705

提示

样例输出的三个数分别对应的分数为： $\frac{9}{2}$, $\frac{11}{20}$, $\frac{99}{20}$

Problem H. Rolling Star

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
空间限制: 512 megabytes

Stella 是一位刚进入 BUAA 的萌新小朋友。

Stella 好不容易才学会计算课程成绩的方法。具体来说，一门课程有其学分 c ，某位已修该门课程的同学会获得成绩 s 。其中学分 c 一般为正整数，成绩 s 为不超过 100 的非负整数，并且成绩 s 有其对应的 GPA，计算公式为：

$$g(s) = 4 - \frac{3(100 - s)^2}{1600} \quad (60 \leq s \leq 100)$$

假设一位同学已修 n 门课程，这些课程的学分分别为 $c_1, c_2, \dots, c_n$ ，对应每门课程的成绩分别为 $s_1, s_2, \dots, s_n$ ，则这位同学的已修学分数为 $c = \sum_{i=1}^n c_i$ ，加权平均分为：

$$\bar{s} = \frac{\sum_{i=1}^n c_i s_i}{\sum_{i=1}^n c_i}$$

加权 GPA 为：

$$\bar{g} = \frac{\sum_{i=1}^n c_i g(s_i)}{\sum_{i=1}^n c_i}$$

Stella 的同学 Rolling Star 是 BUAA 卷王界冉冉升起的新星。Stella 很想知道 Rolling Star 有多卷，于是 Stella 打听到了 Rolling Star 的已修学分数 c 、加权平均分 $\bar{s}$ 和加权 GPA $\bar{g}$ 。

Stella 想根据得到的三个数值求出一组 Rolling Star 可能的课程成绩表以供瞻仰。课程成绩表为一组学分和成绩的列表，形如 $(c_1, s_1), (c_2, s_2), \dots, (c_n, s_n)$ 。为了简化 Stella 的计算，课程成绩表中可以有任意门课程，但每门课程的学分 c_i 必须为正整数，并且由于 Rolling Star 是不会挂科的卷王，每门课程的成绩 s_i 不低于 60。

由于可能存在的精度误差，课程成绩表计算得到的加权平均分和加权 GPA 与给出的值相差不超过 10^{-9} 即认为二者相等。一组课程成绩表符合条件，当且仅当计算得到的已修学分数、加权平均分和加权 GPA 均与 Stella 给出的值相等。

作为萌新，Stella 并没有能力求出符合条件的课程成绩表，甚至不能确定打听到的数据是否正确，于是 Stella 找到了你帮忙。你需要判断是否存在符合条件的课程成绩表，如果存在则求出符合条件的任意一组课程成绩表。

输入格式

第一行，一个整数 c ($1 \leq c \leq 250$)，表示 Rolling Star 的已修学分数。

第二行，两个包含 10 位小数的实数 $\bar{s}, \bar{g}$ ($60 \leq \bar{s} \leq 100$, $1 \leq \bar{g} \leq 4$)，表示 Rolling Star 的加权平均分与加权 GPA。

输出格式

若存在符合条件的课程成绩表，则第一行输出 **YES**。第二行包含一个整数 n ，表示课程成绩表包含的课程数。接下来 n 行，第 i 行包含两个整数 c_i, s_i ($1 \leq c_i \leq c$, $60 \leq s_i \leq 100$)，表示第 i 门已修课程的学分和成绩。

若不存在符合条件的课程成绩表，则输出 NO。

样例

standard input	standard output
9 98.6666666667 3.9941666667	YES 4 2 97 2 98 2 99 3 100
4 98.2500000000 3.9950000000	NO

Problem I. 请问您今天要来点取模吗？

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
空间限制: 256 megabytes

Serval 是加帕里幼儿园的新生。

Serval 觉得取模是一个很有趣的运算，所以……请问您今天要来点取模吗？

Serval 给出了一个正整数序列 $A_1, A_2, \dots, A_n$ 。对于一个非负整数 x ，Serval 定义 $f(x) = (((x \bmod A_1) \bmod A_2) \cdots) \bmod A_n$ 。

Serval 有两个非负整数 L, R ，你只需要简单地求出 $\sum_{k=L}^R f(k)$ 对 1 000 000 007 取模的结果即可。

输入格式

第一行，三个整数 n, L, R ($1 \leq n \leq 10^5, 0 \leq L \leq R \leq 10^{18}$)，分别为序列长度以及和式中 k 的上下界。

第二行， n 个正整数 $A_1, A_2, \dots, A_n$ ($1 \leq A_i \leq 10^{18}$)，代表给出的序列。

输出格式

共一行，一个整数， $\sum_{k=L}^R f(k)$ 对 1 000 000 007 取模的结果。

样例

standard input	standard output
3 141592653 589793238	224100293
2 718281828 459045235	