全国信息学奥林匹克联赛（NOIP2020）模拟赛

考试时间：4 小时

题目名称	并肩作战	旅游	战成平手	礼物
题目类型	传统型	传统型	传统型	传统型
目录	fight	travel	tie	gift
可执行文件名	fight	travel	tie	gift
输入文件名	fight.in	travel.in	tie.in	gift.in
输出文件名	fight.out	travel.out	tie.out	gift.out
每个测试点时限	4 秒	1 秒	1 秒	3 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	20	20	8	6
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	fight.cpp	travel.cpp	tie.cpp	gift.cpp

编译选项

对于 C++ 语言	-O2 -lm -m32

注意事项

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
4. 评测时采用的机器配置为：Intel i5 − 4200M (4) @ 3.100GHz，内存 8GB。上述时限以此配置为准。

5. 只提供 Linux 格式附加样例文件。
6. **选手应在选手文件夹内建立各个题目对应的子文件夹，各题的源程序应放在相应的子文件夹里。**
7. 特别提醒：评测在当前最新版本的 **Arch Linux** 下进行，但是使用的编译器版本为 **GCC 4.8.5**，并开启**-m32**。

并肩作战（fight）

【题目背景】

小 d 和小 w 在玩一款网游。

【题目描述】

现在他们要参加 q 次战斗，每次小 d 和小 w 需要买装备去战斗。

由于职业不同，所以小 d 和小 w 能买的装备不一样。

小 d 不买装备战斗力为 a_0。一共有 n 个装备，买第 i 个装备后战斗力为 a_i。

小 w 不买装备战斗力为 b_0。一共有 m 个装备，买第 i 个装备后战斗力为 b_i。

小 d 和小 w 能买的第 i 个装备价格都为 i。

现在每次战斗前小 d 和小 w 分别可以不买装备或买恰好一个装备，装备只能在这一次中使用。

两个人的总战斗力是两个人的战斗力乘积。

对于每次战斗，小 d 和小 w 预算在 $[l,r]$ 之间，也就是要求总花费在这个区间中，求所有方案中两个人的总战斗力和，其中两个人的总战斗力是两个人分别战斗力的乘积。

当然由于这个数很大，你只需要输出它对 1145141923($=2\times3\times31\times409\times15053+1$，一个质数) 取模的值。

【输入格式】

从文件 ***fight.in*** 中读入数据。

由于输入量可能很大，所以本题改为在程序中生成 a_i,b_i。

第一行五个数 $n,m,q,seed,w$，表示小 d 能买的装备个数，小 w 能买的装备个数，战争的次数，和随机数相关的种子，还有 a_i,b_i 的值域，最终数据保证 $0\le a_i,b_i<w$。

接着 q 行，每行两个数 l,r 表示小 d 小 w 对这次战斗的预算。

选手目录中的 `fight/fight_sample.cpp` 包含了读入的部分，选手可以对其进行修改来完成此题。

【输出格式】

输出到文件 ***fight.out*** 中。

共 q 行，每行一个数 s，表示所有方案的总战斗力和对 1145141923 取模的结果。

【样例 1 输入】

```
2 2 2 114514 5
1 2
0 4
```

【样例 1 输出】

```
21
40
```

【样例 1 解释】

$a=[3,0,2],b=[3,3,2]$。

对于第一组询问，$(0,1)(1,0)(0,2)(1,1)(2,0)$ 均为可行的选法，答案为 $3\times3+0\times3+3\times2+0\times3+3\times2=21$。

对于第二组询问，所有方案均可行，答案为 $3\times3+3\times3+3\times2+0\times3+0\times3+0\times2+2\times3+2\times3+2\times2=40$

【样例 2】

见选手目录下的 ***fight/fight2.in*** 与 ***fight/fight2.ans***。

【数据范围】

对于前 30% 的数据，保证 $1\le n,m\le500$。

对于前 45% 的数据，保证 $1\le n,m\le10^5$。

对于另外 15% 的数据，保证 $1\le n,m\le10^6,1\le w\le20$。

对于前 60% 的数据，满足 $1\le q\le100$。

对于后 40% 的数据，满足 $1\le n,m\le6\times10^6,1\le q\le25$。

对于 100% 的数据，满足 $1\le w\le1145141923,0\le seed<2^{64},0\le l\le r\le n+m$。

【提示】

注意模数。

旅游（travel）

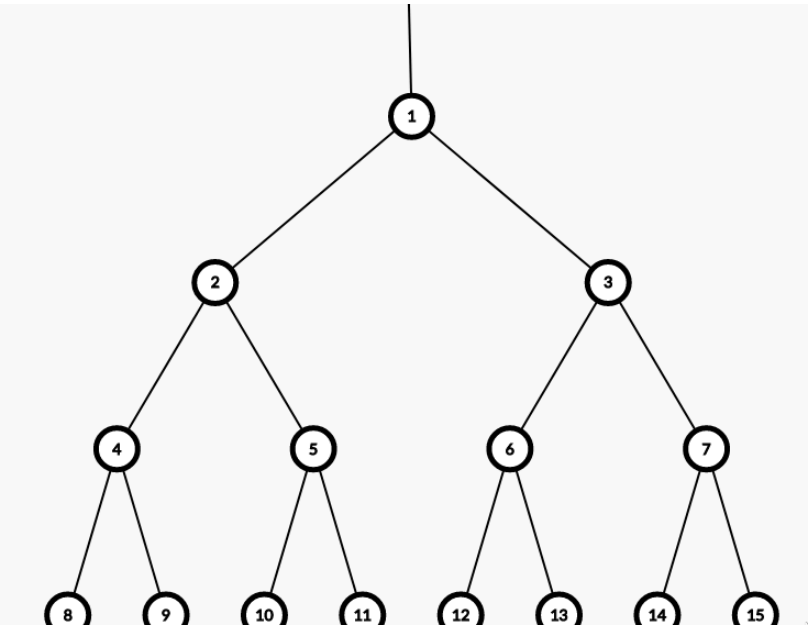
【题目背景】

小 d 和小 w 相约一起到 B 城旅游。

【题目描述】

B 城有很多条河流，把城市分成了很多岛屿，具体的，如果将河流的交叉点记为从 1 开始的正整数，则 $(i, 2i), (i, 2i+1)$ 之间都有一条河流，同时还有一条从无限远到 1 的河流，显然如果两个点之间能不经过河流到达就在同一个岛屿中。

如图：



现在小 d 乘坐的船在 x 点抛锚了，小 w 乘坐的船在 y 点也抛锚了。

小 d 现在要去找到小 w。虽然小 d 和小 w 都不会游泳没法渡河，但是每条河流上都有一座桥，不过每次通过桥需要当地货币 1 元的花费。

由于小 d 移动速度非常快，所以不需要考虑用时，小 d 希望走一条花费最少的路径。

注意：x 点到相邻的岛屿，y 点到相邻岛屿可以直接通过船，不需要过桥。

【输入格式】

从文件 ***travel.in*** 中读入数据。

第一行一个数 T，表示一共有 T 组数据。
接下来 T 行每行两个数 a,b，表示一组询问。

【输出格式】

输出到文件 ***travel.out*** 中。
T 行，每行一个数 s 表示最少的花费。

【样例 1 输入】

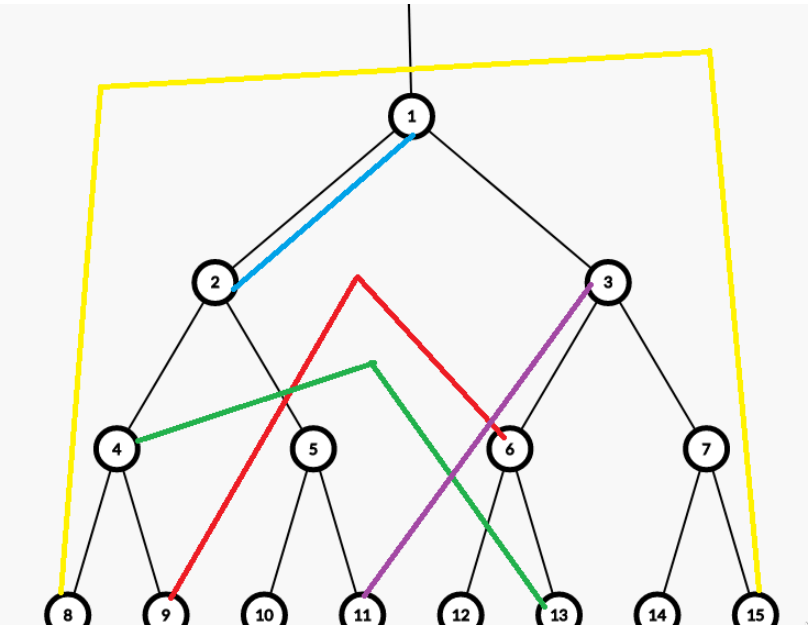
```
10
9 6
13 4
1 2
3 11
8 15
425 145
80150 313
123457 654322
672525845223757475 339224993515259184
1 1000000000000000001
```

【样例 1 输出】

```
1
2
0
0
1
6
3
9
46
16
```

【样例 1 解释】

下面是前 5 组数据的样例解释，其中每组数据的方案用不同颜色的线段表示：



【数据范围】

对于 20% 的数据，$a, b \leq 7$。
对于 50% 的数据，$a, b \leq 1000$。
对于 70% 的数据，$a, b \leq 10^6$。
对于 100% 的数据，$1 \leq T \leq 10^4, 1 \leq a, b \leq 10^{18}$。

战成平手（tie）

【题目背景】

小 d 和小 w 在玩游戏，他们玩的是一个非常经典的游戏：取石子游戏。

【题目描述】

现在有 n 堆魔法石，每个人每次可以给其中一堆石子中拿走任意正整数个，不能操作的获胜。
但是由于魔法石有魔法，如果两堆石子个数相同就会同时消失。
由于小 d 非常绅士，所以由小 w 先手。
现在由你来记录游戏结果。
当然众所周知，小 d 和小 w 都绝顶聪明，所以会采取最优策略。
由于某些原因，无论结果如何你都需要记录战成平手 (tie)，但你还是想记录下真实的结果，所以在小 w 获胜的时候你需要记录 `tie`，小 d 获胜的时候你需要记录 `TIE`。

【输入格式】

从文件 ***tie.in*** 中读入数据。
第一行一个数 T，表示小 d 和小 w 进行的游戏次数，接下来 $2T$ 行，每两行表示一次游戏。
对于每次游戏，第一行一个数 n，表示有 n 堆石头。
第二行 n 个数，表示 n 堆石头的个数。

【输出格式】

输出到文件 ***tie.out*** 中。
共 T 行，第 i 行输出一个字符串表示第 i 次游戏后你记录的结果。

【样例 1 输入】

```
3
3
1 47 7
2
2 3
3
12 20 24
```

【样例 1 输出】

```
tie
TIE
TIE
```

【样例 1 解释】

对于第一组数据，小 w 只需要把 47 那一堆中取走 40 个，就能留给小 d 仅有一个石头的局面，这样就胜利了。

对于第二组数据，考虑小 w 可能的操作：

将其中一堆全部拿走：小 d 可以把另一堆拿到剩一个。

将其中一堆留下一个：小 d 可以把另一堆全部拿走。

将 3 变成 2：同时消失，小 d 直接获胜。

所以小 d 必胜。

【数据范围】

对于 100% 的数据，$T \leq 1000, 1 \leq n \leq 50, 1 \leq a_i \leq 10^9, a_i$ 两两不同。

每个测试点的具体限制见下表 (留空表示无特殊限制)：

测试点编号	分数	n	a_i
1	2	$= 1$	
2	8	≤ 2	
3	10	≤ 3	
4	10		≤ 5
5	10		≤ 10
6	20		≤ 100
7	10	≤ 10	
8	30		

礼物（gift）

【题目背景】

小 d 准备给小 w 送礼物。

【题目描述】

商店里有 n 个礼物，第 i 个的种类为 a_i。

现在小 d 要给小 w 送 $\frac{n(n+1)}{2}$ 次礼物，每次先选定一个**不同**的区间 (显然一共只有 $\frac{n(n+1)}{2}$ 个不同区间)，然后从这个区间选出一些，按照特定顺序送给小 w。

在一次送礼物中，如果第 i 个送给小 w 的礼物是第 b_i 种，则小 d 和小 w 之间会增加 $\sum i \cdot b_i$ 点羁绊值。

小 d 不希望一次送的礼物中出现同样种类的礼物 (但不同次的可以)，在此基础上小 d 希望最大化羁绊值。

求送完 $\frac{n(n+1)}{2}$ 次礼物之后小 d 和小 w 之间羁绊值增加了多少，这个数很大，请输出对 (10^9+7) 取模后的结果。

【输入格式】

从文件 ***gift.in*** 中读入数据。

第一行一个数 n，表示礼物的个数。

第二行 n 个数 $a_1 \cdots a_n$，表示每个礼物的种类。

【输出格式】

输出到文件 ***gift.out*** 中。

一行一个整数表示答案。

【样例 1 输入】

```
3
1 3 3
```

【样例 1 输出】

```
24
```

【样例 1 解释】

对于 $[1,1],[2,2],[3,3]$ 只能选这一个礼物，分别增加 $1,3,3$ 点羁绊值。
对于 $[1,2]$ 方案是 $\{1,3\}$，羁绊值是 $1+3\times 2=7$
对于 $[1,3]$ 方案是 $\{3\}$，羁绊值是 3。
对于 $[1,3]$ 方案是 $\{1,3\}$，羁绊值是 7。
总共羁绊值增加了 $1+3+3+7+3+7=24$。

【样例 2 输入】

```
8
4 3 4 4 7 1 2 1
```

【样例 2 输出】

```
861
```

【样例 3..8】

见选手目录下的 ***gift/gift3..8.in*** 与 ***gift/gift3..8.ans***。
第 $i+2$ 个数据包满足第 i 个 Subtask 的限制。

【数据范围】

由于某些原因，本题使用捆绑测试。
对于 100% 的数据：$n\leq 5\times 10^4, 1\leq a_i\leq n$。
每个子任务的具体限制见下表 (留空表示无特殊限制)：

子任务编号	分数	n	a_i	特殊限制
1	25	$\leq 10^2$		
2	20	$\leq 10^3$		
3	1		$=1$	
4	14			a_i 是 $1\cdots n$ 的一个排列
5	25	$\leq 2\times 10^4$		
6	15			