

必可赛前公益众筹赛

第六试

时间: 2024 年 10 月 12 日 08:00 ~ 12:00

题目名称	蛋神的邀请	信徒的座位	虔诚的演说	拜蛋的阴谋
题目类型	传统型	传统型	传统型	传统型
目录	invitation	believer	speech	plot
可执行文件名	invitation	believer	speech	plot
输入文件名	invitation.in	believer.in	speech.in	plot.in
输出文件名	invitation.out	believer.out	speech.out	plot.in
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	1024 MB
测试点数目	5	20	20	20
测试点是否等分	否	是	是	是

提交源程序文件名

对于 C++ 语言	invitation.cpp	believer.cpp	speech.cpp	plot.cpp
-----------	----------------	--------------	------------	----------

编译选项

对于 C++ 语言	-lm -O2 -std=c++14
-----------	--------------------

注意事项与提醒（请选手务必仔细阅读）

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立好的，且带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
4. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
5. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
6. 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**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 ulimit -s unlimited 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。

12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

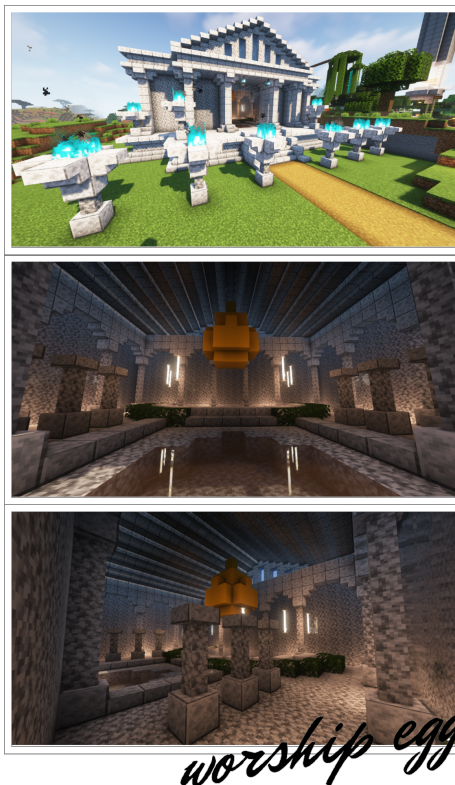
13. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

14. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

15. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**

蛋神的邀请 (invitation)

【题目描述】



发烧到无能为力|内卷到感动自己

拜蛋神庙落成了！为了庆祝拜蛋神庙的落成，宣扬伟大的飘·乱世佳人·蛋丁（下文简称蛋神），拜蛋教将在此举行一场盛大的拜蛋仪式。你作为拜蛋教的教主，将与蛋神对话，遵照他的意愿为他挑选将参与这场仪式的人选。

现在已经有 n 个人主动向你报名，他们都自称是虔诚的蛋神追随者。

蛋神的信徒拥有如 **贞洁、勤奋、慷慨、谦逊、温和、节制、宽容** 等美德，每个人都有一个最突出的美德，对于第 i 个人，我们用 a_i 记录这项美德。蛋神的信徒不仅个人信仰要足够虔诚，更要有团结博爱的精神。根据拜蛋教旨，我们认为一个集体是“ k 团结”的，当且仅当在该集体中 **所有存在的美德都出现了 k 的倍数次**。

接下来，你作为蛋神的祭司，将与蛋神进行 m 次对话，每次蛋神可能降下以下两种神谕中的一种：

- 1 i x: 蛋神通过感知，第 i 个报名人的最突出美德变为了 x ；
- 2 l r k: 表示若第 l 至第 r 个报名的人为“ k 团结”的，则发送邀请参加拜蛋仪式。

而你需要根据蛋神的神谕，决定每次是否发送邀请。作为蛋神最喜欢的的祭司，你已经决定好了每次是否要发送邀请，不过你需要抓一个下层神职人员来帮你验证。但是现在是 3202 年，你只能搞到一台计算机来帮你，所以你还需要自己编写一个程序来计算。

【输入格式】

从文件 *invitation.in* 中读入数据。

第一行输入两个整数 n, m ，表示报名人数与降下神谕的次数；

第二行 n 个整数，第 i 个表示 a_i ；

接下来 m 行，每行一个对神谕的描述，格式如题目描述所示。

【输出格式】

输出到文件 *invitation.out* 中。

对于每次第二类神谕，输出一行 YES 或 NO，表示这次神谕是否要发送邀请。

【样例 1 输入】

```
1 6 4
2 1 1 2 2 3 3
3 2 1 6 3
4 2 1 4 2
5 1 4 3
6 2 1 4 2
```

【样例 1 输出】

```
1 NO
2 YES
3 NO
```

【样例 1 解释】

初始序列为 $[1, 1, 2, 2, 3, 3]$ ；

第一次询问 $[1, 6]$ ，每种美德都出现了 2 次，均不为 3 的倍数；

第二次询问 $[1, 4]$ ，每种美德都出现了 2 次，均为 2 的倍数；

第三次修改后序列变为 $[1, 1, 2, 3, 3, 3]$ ；

第四次询问 $[1, 4]$ ，三种美德各出现了 2, 1, 1 次，其中 1 不为 2 的倍数。

【样例 2 输入】

```
1 10 8
2 1234 2 3 3 2 1 1 2 3 4
3 2 1 6 2
4 1 1 1
```

```
5 2 1 6 2
6 2 1 9 2
7 1 10 5
8 2 1 9 3
9 1 3 5
10 2 3 10 2
```

【样例 2 输出】

```
1 NO
2 YES
3 NO
4 YES
5 YES
```

【样例 3】

见选手目录下的 *invitation/invitation3.in* 与 *invitation/invitation3.ans*。
该组样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 *invitation/invitation4.in* 与 *invitation/invitation4.ans*。
该组样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 *invitation/invitation5.in* 与 *invitation/invitation5.ans*。
该组样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 *invitation/invitation6.in* 与 *invitation/invitation6.ans*。
该组样例满足子任务 5 的限制。

【数据范围】

对于 100% 的测试数据: $1 \leq n, m \leq 3 \times 10^5, 1 \leq a_i, x_i \leq 10^9, 1 \leq l_i \leq r_i \leq n, 1 \leq k_i \leq n$ 。

子任务编号	$n, m \leq$	特殊限制	分值
1	5000	无	15
2	3×10^5	$a_i, x_i \leq 100$	
3	10^5	不存在第一类神谕	25
4		无	
5	3×10^5		20

信徒的座位 (believer)

【题目描述】

你已经邀请到了蛋神的信徒，下一步就是为每个信徒安排在拜蛋仪式上的座位了。虽然这听起来很简单，但蛋神有自己的要求。

拜蛋神庙内共有 m 个座位一字排开，每个座位仅能容纳一人。

所有与会的信徒都根据其美德被分在 n 组之一，每组都有 l 个人，还记得拜蛋教团结博爱的精神吗？因此这 l 个人的座位 **必须相邻**。此外，由于每组信徒都有自己对于瞻仰蛋神雕塑的角度需求，每组信徒都有一个希望占有的座位 x_i ，表示其希望占有 $[x_i, x_i + l - 1]$ 这段区间内的座位。当他们占有这些座位时，他们将不胜犬马怖惧之情，贡献 2 单位的虔诚值。

当然，我们很可能不能满足所有信徒的要求，但如若某组信徒可以在非自己希望的座位上参加拜蛋仪式（当然座位 **也必须是连续的**），因为可以近距离感受神迹，他们也会感到沐浴清化，贡献 1 单位的虔诚值。而如果某组信徒无法获得座位，我们将单独为他们设置座位，无需在这里再考虑他们，他们也 **不会贡献虔诚值**。

作为拜蛋教的教主，你自然希望所有人的总虔诚值的和最高，那么这个最大的虔诚值是多少呢？

此外，因为你还需要引导每组信徒到指定位置就坐，你还希望能同时求出能获得最大虔诚值的座位方案。

【输入格式】

从文件 *believer.in* 中读入数据。

第一行三个正整数 m, l, n ，表示座位数，每组的人数和组数；

接下来一行包含 n 个正整数 x_1, x_2, \dots, x_n ，表示每组信徒希望占有的座位的起始编号。

【输出格式】

输出到文件 *believer.out* 中。

第一行一个整数 S ，表示最大的虔诚值。

第二行一个整数 q ，表示共有 q 组信徒将被安排在这一排内。

接下来 q 行，每行两个整数 x_i, y_i ，表示 **输入顺序** 中第 x_i 组信徒将获得从 y_i 开始连续的 l 个座位。

你需要保证输出的 $\{y_i\}$ 为单调递增数列。

如果有多种方案 **输出任意一种即可**。

【样例 1 输入】

```
1 10 3 3
2 1 2 8
```

【样例 1 输出】

```
1 5
2 3
3 1 1
4 2 4
5 3 8
```

【样例 1 解释】

样例输出中的分配方案如下图所示：

1	1	1	2	2	2		3	3	3
---	---	---	---	---	---	--	---	---	---

其中虔诚值贡献为 2 的组所在格子被阴影填充。

【样例 2】

见选手目录下的 *believer/believer2.in* 与 *believer/believer2.ans*。

该组样例满足测试点 5 ~ 10 的限制。其中样例输出只给出了最大虔诚值。

【样例 3】

见选手目录下的 *believer/believer3.in* 与 *believer/believer3.ans*。

该组样例满足测试点 11 ~ 20 的限制。其中样例输出只给出了最大虔诚值。

【数据范围】

对于全部数据, $1 \leq l \leq m \leq 10^6, 1 \leq n \leq 10^5, 1 \leq x_i \leq m - l + 1$ 。

测试点编号	$n \leq$	特殊限制
1	10^5	$l = 1$
2 ~ 4	15	无
5 ~ 10	2000	$m \leq 2000$
11 ~ 20	10^5	无

虔诚的演说 (speech)

【题目描述】

拜蛋仪式的最后,是由信徒们各自讲述蛋神的光辉事迹或者自己的拜蛋故事,以此加强对他的信仰。

当然这里的光辉故事就有很多了,包括但不限于每次下矿总能找到半组钻石、与兔国互(jié)换(lǜè)一把高配弓、在十二倍速熔炉的地毯机前沉(diào)思(xiàn)。

当然信徒也会讲述在小黑屋内为蛋神种田,或是在交易所内心甘情愿接受(dǎ)教(gǔ)化(zhé),或者因爬上了拜蛋神庙的房顶而面对蛋神手下恩(zuàn)赐(jiàn)的故事。

在活动开始前,每位信徒都有一个虔诚值 a_i 。

n 位信徒将按照某个顺序依次在祭坛前讲述自己的故事 **恰好一次**。通过讲述,他对蛋神的信仰将会越发清晰,虔诚值也将随之上升。但由于性格不同,每位讲述者的虔诚值的变化规则不同。

一类讲述者乐于帮助那些信仰还不完全坚定的信徒,这类讲述者讲述前若有 c 位听众的虔诚值 **严格小于** 自身,则自身的虔诚值将增加 c ,我们称这类信徒为第 0 类讲述者。

另一类讲述者乐于让那些本就坚定的信徒忠实信仰,这类讲述者讲述前若有 c 位听众的虔诚值 **严格大于** 自身,则自身的虔诚值将增加 c ,我们称这类信徒为第 1 类讲述者。

当然拜蛋教的团结博爱精神已经不需要赘述了,因此我们希望在完成所有讲述后所有人的 **虔诚值都是相等的**,这样才能让大家兴尽而归。

你作为拜蛋教教主,自然开始思考。有多少种顺序安排 n 位信徒的讲述,使得上述要求可以被满足。当然这个值可能很大,因此你需要输出其对蛋神最喜欢的质数 998244353 取模后的结果。

【输入格式】

从文件 `speech.in` 中读入数据。

第一行一个整数 n , 表示参加活动的人数。

第二行 n 个整数,第 i 个为 a_i , 表示在活动开始前第 i 位信徒的虔诚值。

第三行 n 个整数,第 i 个为 b_i , 表示第 i 位信徒为 b_i 类讲述者。

【输出格式】

输出到文件 `speech.out` 中。

输出一行一个整数,表示答案对 998244353 取模的结果。

【样例 1 输入】

```
1 3
2 1 2 3
3 1 0 1
```

【样例 1 输出】

```
1 3
```

【样例 1 解释】

注意到只要信徒 2 在信徒 1 之前演讲则满足条件，故有 3,2,1, 2,3,1, 2,1,3 三种顺序满足条件。

【样例 2 输入】

```
1 21
2 1 2 19 19 19 19 19 19 19 19 19 21 21 21 21 21 21 21 21 21
3 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1
```

【样例 2 输出】

```
1 49439766
```

【样例 3】

见选手目录下的 *speech/speech3.in* 与 *speech/speech3.ans*。
该组样例满足测试点 10 ~ 12 的限制。

【样例 4】

见选手目录下的 *speech/speech4.in* 与 *speech/speech4.ans*。
该组样例满足测试点 17 ~ 20 的限制。

【数据范围】

对于全部数据， $1 \leq n \leq 2 \times 10^6$, $1 \leq a_i \leq 2n$, $0 \leq b_i \leq 1$ 。

测试点编号	$n \leq$	特殊限制
1 ~ 3	10	无
4 ~ 5	2×10^6	所有 a_i 相等
6 ~ 7		所有 $b_i = 1$
8 ~ 9		所有 $b_i = 0$
10 ~ 12	2000	无
13 ~ 16	2×10^5	
17 ~ 20	2×10^6	

【提示】

本题输入文件较大，请选择合适的输入方式。

拜蛋的阴谋 (plot)

【题目描述】

拜蛋仪式圆满结束，众人都怀揣着对蛋神更崇敬的心返回了。

不过既然提到了猪灵塔，那我们就不得不来介绍一下它了。

猪灵塔由僵尸猪灵刷新点与挂机点组成，僵尸猪灵刷新点通过限制可以刷新生物空间的高度与刷新点的地板材质防止除僵尸猪灵以外的生物出现。并利用其试图踩碎海龟蛋的特性经由传送门运输到挂机点。（这里涉及到传送门连接相关的知识点，需要考虑传送门的朝向，下界与主世界坐标换算和共用传送门的距离等）而挂机点因其高度极高，可以有效防止正常的下界刷新干扰猪灵塔的效率。使得刷怪塔理论上可以拉满 70 怪物/区块/玩家的刷怪限制。

不过上面都是用来水字数的，这题我们只关心连接猪人刷新点和挂机点的脚手架。

我们可以把脚手架看做一棵 n 个结点的树，由 $n - 1$ 条等长的边连接，我们认为这个长度为 1 单位长度。

现在你想搞点艺术，于是你决定把那些孤立的，也即只有一条边与之相连的点两两连接起来。当然这样搞艺术还不够，所以连接这些点之间的边将会长达 x 单位长度。

搞完艺术后，我们再来考虑它的实用性。我们认为连接后的图的“不方便度” f_x 为其上两个点之间的最远距离，即：

$$\max_{1 \leq u < v \leq n} dis(u, v)$$

现在你有 q 个艺术计划，每个计划都将给出 x 具体的值。你需要回答在该计划下改造后的的不方便度。当然，由于这只停留在计划层面，因此每次询问不会真正将结点连接起来。

【输入格式】

从文件 `plot.in` 中读入数据。

第一行一个整数 n ，表示树的结点数；

第二行 $n - 1$ 个整数 p_2, p_3, \dots, p_n ，其中第 p_i 表示结点 i 与 p_i 之间有一条边连接；

第三行输入一个整数 q ；

第四行输入 q 个正整数，其中第 i 个表示第 i 次询问的 x_i 。

【输出格式】

输出到文件 `plot.out` 中。

输出一行 q 个整数，表示每个询问的答案。

【样例 1 输入】

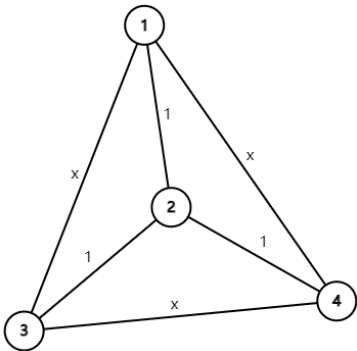
1 4

```
2 1 2 2
3 4
4 1 2 3 4
```

【样例 1 输出】

```
1 1 2 2 2
```

【样例 1 解释】



【样例 2 输入】

```
1 10
2 1 1 1 1 1 5 3 4 5
3 8
4 9 2 10 3 1 6 8 4
```

【样例 2 输出】

```
1 4 3 4 3 2 4 4 4
```

【样例 3】
见选手目录下的 *plot/plot3.in* 与 *plot/plot3.ans*。
该组样例满足测试点 4 ~ 5 的限制。

【样例 4】
见选手目录下的 *plot/plot4.in* 与 *plot/plot4.ans*。
该组样例满足测试点 9 ~ 12 的限制。

【样例 5】

见选手目录下的 *plot/plot5.in* 与 *plot/plot5.ans*。

该组样例满足测试点 19 ~ 20 的限制。

【数据范围】

对于全部数据， $1 \leq n \leq 10^6$ ， $1 \leq q \leq 10$ ， $p_i < i$ 。所有询问的 x_i 满足 $1 \leq x_i \leq n$ 且互不相同。

测试点编号	$n \leq$	特殊限制
1	10^6	$q = 1$ 且 $x_1 = n$
2 ~ 3	1000	无
4 ~ 5	5000	
6	10^6	存在一个点度数为 $n - 1$
7 ~ 8		$p_i = i - 1$
9 ~ 12		p_i 从 $[1, i - 1]$ 中均匀随机生成
13 ~ 15	10^5	无
16 ~ 18	5×10^5	
19 ~ 20	10^6	