

NOIP2025 模拟赛

第一试

时间：2025 年 7 月 29 日 08:00 ~ 12:30

题目名称	统计	序列	划分	操作
题目类型	传统型	传统型	传统型	传统型
目录	count	seq	divide	op
可执行文件名	count	seq	divide	op
输入文件名	count.in	seq.in	divide.in	op.in
输出文件名	count.out	seq.out	divide.out	op.out
每个测试点时限	1.0 秒	1.5 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	10	20	25	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	count.cpp	seq.cpp	divide.cpp	op.cpp
-----------	-----------	---------	------------	--------

编译选项

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

注意事项（请仔细阅读）

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++ 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，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 选手提交的程序源文件必须不大于 100KB。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 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。
7. 直接复制 PDF 题面中的多行样例，数据将带有行号，并且某些字符可能无法正常显示，建议选手直接使用对应目录下的样例文件进行测试。

统计 (count)

【题目描述】

给定两个正整数 x, p ，求出有多少个正整数 y 满足 $\gcd(x, y)^p = \text{lcm}(x, y)$ ，答案对 998244353 取模（换句话说，你需要输出答案除以 998244353 后的余数）。可以证明满足条件的正整数 y 的数量是有限的。

其中， $\gcd(x, y)$ 表示满足同时为 x, y 的因数的所有数字中，最大的那个数字，例如 $\gcd(6, 8) = 2$ ， $\gcd(8, 12) = 4$ 。 $\text{lcm}(x, y)$ 表示满足同时为 x, y 的倍数的所有数字中，最小的那个数字，例如 $\text{lcm}(6, 8) = 24$ ， $\text{lcm}(8, 12) = 24$ 。

【输入格式】

从文件 `count.in` 中读入数据。

第一行包含两个整数 x, p 。

【输出格式】

输出到文件 `count.out` 中。

输出一行一个整数表示有多少个满足条件的正整数 y ，对 998244353 取模。

【样例 1 输入】

```
1 4 2
```

【样例 1 输出】

```
1 2
```

【样例 1 解释】

$y = 2$ 时 $\gcd(4, 2)^2 = 4 = \text{lcm}(4, 2)$ 。

$y = 16$ 时 $\gcd(4, 16)^2 = 16 = \text{lcm}(4, 16)$ 。

可以证明只有这两个数满足条件。

【样例 2 输入】

```
1 7 3
```

【样例 2 输出】

11

【样例 3 输入】

1576 3

【样例 3 输出】

12

【样例 4 输入】

1183514777 10

【样例 4 输出】

11

【数据范围】

对于所有数据，保证 $2 \leq x \leq 10^9$ ， $1 \leq p \leq 10$ 。

测试点编号	特殊性质
1	$p = 1$
2 ~ 3	$p = 2$
4	x 为质数
5 ~ 6	$x \leq 50$ 且 $p \leq 3$
7 ~ 8	$x \leq 3 \times 10^3$
9 ~ 10	无

序列 (seq)

【题目描述】

给定一个长度为 n 的序列 a 。从中选择 m 个数，将选择的元素按原顺序排成一个首尾相邻的序列 b ，此时，我们定义 b 的权值为所有相邻两个数之和的最大值（形式化地，权值为 $\max_{i=1}^m \{b_i + b_{(i \bmod m)+1}\}$ ）。求出 b 的权值最小能为多少。

【输入格式】

从文件 `seq.in` 中读入数据。
第一行包含两个整数 n, m 。
第二行包含 n 个整数 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 `seq.out` 中。
输出一行一个整数表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 9
```

【样例 1 解释】

保留 $[1, 7, 2, 5, 3, 5]$ 即可，权值为 $\max\{1 + 7, 7 + 2, 2 + 5, 5 + 3, 3 + 5, 5 + 1\} = 9$ 。

【样例 2】

见选手目录下的 `seq/seq2.in` 与 `seq/seq2.ans`。

【样例 3】

见选手目录下的 `seq/seq3.in` 与 `seq/seq3.ans`。

【样例 4】

见选手目录下的 *seq/seq4.in* 与 *seq/seq4.ans*。

【样例 5】

见选手目录下的 *seq/seq5.in* 与 *seq/seq5.ans*。

【样例 6】

见选手目录下的 *seq/seq6.in* 与 *seq/seq6.ans*。

【样例 7】

见选手目录下的 *seq/seq7.in* 与 *seq/seq7.ans*。

【数据范围】

对于所有数据，保证 $2 \leq m \leq n \leq 2 \times 10^5$ ， $0 \leq a_i \leq 10^8$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 2	20	无
3 ~ 4	60	
5 ~ 6	3×10^2	
7 ~ 10	3×10^3	
11 ~ 12	2×10^5	$a_i \leq 1$
13 ~ 14		$m = 3$
15 ~ 16		$m = n - 2$
17 ~ 20		无

划分 (divide)

【题目描述】

对于一些数，我们定义这些数的 mex 为最小的不在这些数中出现的自然数。例如 $[0, 4, 1, 1, 2]$ 的 mex 为 3， $[2, 1]$ 的 mex 为 0。

给定一个长为 n 的序列 a ，将其划分为若干连续的段，如果每一段对应的数的 mex 相等，我们称这种划分是好的。请你求出有多少种划分方案是好的，对 998244353 取模。

【输入格式】

从文件 *divide.in* 中读入数据。

第一行包含一个整数 n 。

第二行包含 n 个整数 a_1, a_2, \dots, a_n 。

【输出格式】

输出到文件 *divide.out* 中。

输出一行一个整数表示有多少种划分方案是好的，对 998244353 取模。

【样例 1 输入】

```
1 5
2 0 1 1 0 1
```

【样例 1 输出】

```
1 3
```

【样例 1 解释】

好的划分方案有：

- $[0, 1, 1, 0, 1]$
- $[0, 1], [1, 0, 1]$
- $[0, 1, 1], [0, 1]$

【样例 2】

见选手目录下的 *divide/divide2.in* 与 *divide/divide2.ans*。

【样例 3】

见选手目录下的 *divide/divide3.in* 与 *divide/divide3.ans*。

【样例 4】

见选手目录下的 *divide/divide4.in* 与 *divide/divide4.ans*。

【样例 5】

见选手目录下的 *divide/divide5.in* 与 *divide/divide5.ans*。

【样例 6】

见选手目录下的 *divide/divide6.in* 与 *divide/divide6.ans*。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 3 \times 10^5$ ， $0 \leq a_i \leq n$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 2	18	无
3 ~ 4	3×10^2	
5 ~ 10	5×10^3	
11 ~ 14	3×10^5	A
15 ~ 18		B
19 ~ 25		无

特殊性质 A：保证 $a_i \leq 20$ 。
特殊性质 B：保证 a 中每个元素都在 $[0, n]$ 中随机生成。

操作 (op)

【题目描述】

给定一个整数 p 和 n 个操作，操作有如下两种：

1. 给定 x ，将 w 修改为 x 。
2. 给定 x ，将 w 修改为 $(w + x) \bmod p$ 。

其中 w 是一个初始为 0 的变量。

你可以以任意顺序执行上面的 n 个操作，得到最终的 w 。你要求出在 $0 \sim p - 1$ 中，有多少个数是无论以什么顺序执行操作都无法得到的。

【输入格式】

从文件 `op.in` 中读入数据。

本题为多组数据，输入数据第一行包含一个整数 T ，表示数据组数。

对于每组数据：

第一行包含两个整数 p, n 。

接下来 n 行，每行包含两个整数 op_i, x_i 。其中 $op_i = 0$ 表示第一种操作， $op_i = 1$ 表示第二种操作， x_i 表示操作中给定的数。

【输出格式】

输出到文件 `op.out` 中。

对于每组数据，输出一行一个整数表示答案。

【样例 1 输入】

```
1 1
2 6 3
3 1 2
4 0 3
5 1 1
```

【样例 1 输出】

```
1 2
```


【样例 1 解释】

1, 2 无法被生成。

【样例 2】

见选手目录下的 *op/op2.in* 与 *op/op2.ans*。

【样例 3】

见选手目录下的 *op/op3.in* 与 *op/op3.ans*。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 2$ ， $1 \leq n \leq 10^6$ ， $2 \leq p \leq 10^6$ ， $\sum n \leq 1.1 \times 10^6$ ， $op_i \in \{0, 1\}$ ， $0 \leq x_i < p$ 。

测试点编号	$n \leq$	$p \leq$	特殊性质
1 ~ 2	10	10^6	无
3 ~ 4	10^4		A
5 ~ 6	10^3	10^4	无
7 ~ 10	10^5	10^5	
11 ~ 20	10^6	10^6	

特殊性质 A：保证最多存在 12 个第二种操作。