

NOIP 模拟赛

题目名称	实力	素质	钥匙	七星
题目类型	传统型	传统型	传统型	传统型
可执行文件名	capability	quality	key	star
输入文件名	capability.in	quality.in	key.in	star.in
输出文件名	capability.out	quality.out	key.out	star.out
时间限制	1.0 秒	2.0 秒	1.0 秒	2.5 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	10	20	20
测试点等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	capability.cpp	quality.cpp	key.cpp	star.cpp
-----------	----------------	-------------	---------	----------

编译选项

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

1. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
2. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
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. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
6. 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。
7. 考试过程中若对题目有疑问，请联系出题人。
8. 题目很简单，AK 了请勿大声喧哗。

实力 (capability)

【题目描述】

出题人想考考你的构造实力。

给定一个正整数 n ，请你构造一个点数不超过 500 的简单无向图使得三元环数量为 n 。若有多组合法方案，输出任意一种即可。

【输入格式】

一行一个正整数 n 。

【输出格式】

第一行一个正整数 x ，表示图的点数。

接下来 $x - 1$ 行，其中第 i 行有 $x - i$ 个数，第 i 行第 j 个数表示点 i 和点 $i + j$ 是否有边。有边记为 0，无边记为 1。

【样例 1 输入】

```
1 3
```

【样例 1 输出】

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

【提示】

为了方便选手测试，在附件中的 capability 目录下我们下发了 checker.cpp 文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：g++ checker.cpp -o checker -std=c++11。

checker 的使用方式为：checker <inputfile> <outputfile>，参数依次表示输入文件与你的输出文件。

由于显而易见的原因，本题没有大样例。

【子任务】

对于全部的数据，满足 $1 \leq n \leq 2 \times 10^6$ 。

测试点编号	$\sum n \leq$	特殊性质
1	10	无
2	20	无
3	30	无
4 ~ 5	100	无
6	200	无
7	400	无
8 ~ 9	1000	无
10	3000	无
11 ~ 12	10^4	无
13	3×10^4	无
14	10^5	无
15	3×10^5	无
16	10^6	A
17	10^6	B
18	10^6	无
19	1.5×10^6	无
20	2×10^6	无

特殊性质 A：保证存在 x 满足 $n = x^3$ 。
特殊性质 B：保证存在一个完全图满足条件。

素质 (quality)

【题目描述】

大 D 喜欢有素质的序列。

定义一个长度为 n 的序列 a 是有素质的，当且仅当不存在满足 $\max\{a_1, a_2, \dots, a_i\} = \min\{a_{i+1}, \dots, a_n\}$ 的 i 。

给定 n, m, p ，求长度为 n ，值域在 $[1, m]$ 的有素质的序列数量对 p 取模的结果。

【输入格式】

一行三个数 n, m, p ，含义见题目描述。

【输出格式】

一行一个数表示答案。

【样例 1 输入】

```
1 2 2 1000000007
```

【样例 1 输出】

```
1 2
```

【样例 2 输入】

```
1 3 4 1000000009
```

【样例 2 输出】

```
1 36
```

【样例 3 输入】

```
1 228 112263 998244353
```

【样例 3 输出】

1 379700769

【子任务】

对于全部的数据，满足 $1 \leq n \leq 400, 1 \leq m \leq 10^8, 10^8 \leq p \leq 1.01 \times 10^9$ ， p 为质数。

测试点编号	$n \leq$	$m \leq$
1	5	5
2 ~ 3	15	15
4 ~ 6	100	100
7 ~ 8	400	400
9 ~ 10	400	10^8

钥匙 (key)

【题目描述】

对于一个长度为 n 的数组 a ，你可以进行如下操作：

创建一个长度为 n 的数组 c ，初始全为 0。然后进行任意多次以下操作：选择一个 $x \in [1, n]$ ，令 $a_x := (a_x - 2^{c_x}) \times 2$ ，然后 $c_x := c_x + 1$ 。

如果你可以使用该操作使 a 数组严格递增且操作完都是正整数，那么这样一个 a 数组是好的。

你希望找到一个长度为 n 的好的 a 数组使得 $\sum_{i=1}^n a_i$ 最小，在 $\sum_{i=1}^n a_i$ 相同时，找到字典序最小的 a 数组。

由于 n 非常大，你只需要输出 $\sum_{i=1}^n a_i$ 和给定 q 个位置的值 a_{b_1}, \dots, a_{b_q} 。

【输入格式】

第一行两个正整数 n, q 。

第二行 q 个正整数 b_1, \dots, b_q ，表示询问的 q 个位置。

【输出格式】

第一行一个数 $\sum_{i=1}^n a_i$ 。

接下来 q 行每行一个数，分别表示 a_{b_1}, \dots, a_{b_q} 。

【样例 1 输入】

```
1 10 10
2 1 2 3 4 5 6 7 8 9 10
```

【样例 1 输出】

```
1 38
2 1
3 2
4 3
5 3
6 5
7 4
8 4
```

9 5
10 5
11 6

【样例 2】

见选手目录下 *key2.in* 与 *key2.ans*, 该样例满足测试点 6 ~ 8 的性质。

【子任务】

对于全部的数据, $1 \leq n \leq 10^9, 1 \leq q \leq 5 \times 10^5, 1 \leq b_i \leq n$ 。

测试点编号	$n \leq$	$q \leq$
1 ~ 2	8	8
3 ~ 5	1000	0
6 ~ 8	1000	1000
9 ~ 11	10^9	0
12 ~ 13	10^9	1000
14 ~ 16	10^9	5×10^4
17 ~ 20	10^9	5×10^5

七星 (star)

【题目描述】

给定一棵 n 个点的树，点带权，初始所有点点权为 0。

给定 q 次操作，每次操作形如以下四种之一：

1. 给定 l, r, k ，将所有编号在 $[l, r]$ 的点的点权加上 k 。
2. 给定 d, k ，将所有编号为 d 的约数的点的点权加上 k 。
3. 给定 d, k ，将所有编号为 d 的倍数的点的点权加上 k 。
4. 给定 u, v ，查询 u 到 v 的路径上所有点的点权和。

你需要对于每个 4 操作，输出对应的答案。

【输入格式】

第一行两个数 n, q ，含义见题目描述。

接下来 $n - 1$ 行，每行两个数 u, v ，表示树上的一条边。

接下来 q 行，每行一个操作，保证形式为以上四种之一。

【输出格式】

对于每个 4 操作，输出一行一个数表示答案。

【样例 1 输入】

```
1 5 5
2 1 2
3 1 3
4 3 4
5 3 5
6 1 3 5 1
7 2 4 2
8 3 2 3
9 4 2 5
10 4 1 4
```

【样例 1 输出】


```
1 9
2 9
```

【样例 2】

见选手目录下 *star2.in* 与 *star2.ans*，该样例满足 Subtask 1 的性质。

【样例 3】

见选手目录下 *star3.in* 与 *star3.ans*，该样例满足 Subtask 3 的性质。

【样例 4 】

见选手目录下 *star4.in* 与 *star4.ans*，该样例满足 Subtask 4 的性质。

【样例 5】

见选手目录下 *star5.in* 与 *star5.ans*，该样例满足 Subtask 5 的性质。

【子任务】

本题使用捆绑测试。

对于全部的数据， $1 \leq n, q \leq 2 \times 10^5, 1 \leq u, v, l, r, d \leq n, 0 \leq |k| \leq 10^6$ 。

子任务编号	$n, q \leq$	特殊性质	分值
1	5000	无	8
2	10^5	BCD	8
3	10^5	D	12
4	10^5	AC	12
5	10^5	BC	12
6	10^5	A	12
7	10^5	无	16
8	1.5×10^5	无	12
9	2×10^5	无	8

- 特殊性质 A：保证不存在第一类命令。
- 特殊性质 B：保证不存在第二类命令。
- 特殊性质 C：保证不存在第三类命令。
- 特殊性质 D：保证第四类命令总满足 $u = v$ 。