

原神

题目描述

你在玩一款卡牌类游戏。

你有 n 个敌人，第 i 个敌人的攻击力是 a_i 。

你手上有 m 瓶魔法药水以及 1 瓶神奇药水，第 i 瓶魔法药水的能力是使得某一个你指定的敌人的攻击力上升 b_i ，神奇药水的能力是将所有攻击力为奇数的敌人杀死。

你可以以任意顺序使用任意数量的药水，一瓶药水不能使用多次，请最小化敌方剩余敌人的攻击力之和，并输出答案。

输入格式

第一行，包含一个整数 n ，表示敌人个数。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ，表示敌人的攻击力。

第三行，包含一个整数 m ，表示手中魔法药水数量。

第四行包含 m 个整数 b_1, b_2, \dots, b_m ，表示每瓶魔法药水可以给敌人提供的攻击力。

输出格式

一行，一个整数，表示敌方剩余敌人的攻击力之和。

样例

输入样例1

| | |
|---|------------------------------|
| 1 | 10 |
| 2 | 36 2 76 18 28 60 72 80 57 42 |
| 3 | 8 |
| 4 | 42 62 97 31 15 1 17 89 |

输出样例1

| | |
|---|----|
| 1 | 48 |
|---|----|

样例解释

使用魔法药水将敌人攻击力变为 $36 + 1, 2, 76 + 97, 18, 28, 60 + 17, 72 + 31, 80 + 15, 57, 42 + 89$ ，后使用神奇药水敌人攻击力变为 $2, 18, 28$ ，答案即为 48 ，可以证明没有更优答案。

数据范围

对于50%的数据, $n, m \leq 10$, $a_i, b_i \leq 100$ 。

对于另外30%的数据, $n, m \leq 1000$, $a_i, b_i \leq 10^5$ 。

对于100%的数据, $0 \leq n, m \leq 10^5$, $1 \leq a_i, b_i \leq 10^9$ 。

方舟

题目描述

给定一个可重集 S , 对于其所有大小为偶数的子集 T (包含空集), 定义 T 的贡献为 T 中元素乘积 (若 T 为空集, 则贡献为1), 问所有大小为偶数的集合的贡献的和是多少, 也即

$$\sum_{\substack{T \subset S \\ 2 \mid |T|}} \prod_{t \in T} t \mod P$$

输入格式

一行7个整数, 表示 $n, m, k, P, s_1, s_2, s_3$ 。

你可以通过以下代码生成所需的读入:

```
1  int n, k, P, m_[N], a_[N];
2  unsigned int s1, s2, s3;
3  unsigned int next(){
4      unsigned int res;
5      s1^=s3, s3+=3055373123u;
6      res=s1, s2=(s2^s3)&31;
7      s3=(s3>>s2)|((s3<<(31^s2))<<1);
8      return res;
9  }
10 void init() {
11     int m; cin >> n >> m >> k >> P >> s1 >> s2 >> s3;
12     for(int i = 1; i <= n; i++) {
13         m_[i] = next() % m + 1;
14         a_[i] = next() % P;
15     }
16 }
```

其中 n 表示有 n 组 (m_i, a_i) , 每组 (m_i, a_i) 表示可重集中有 m_i 个 a_i 。

k 表示可重集中有 $1, \frac{1}{2}, \dots, \frac{1}{k}$ 。

保证 P 是质数且 $0 \leq a_i, k < P$ 。

输出格式

一行一个整数表示答案。

样例

样例输入1

```
1 | 1 1 2 998244353 1 2 3
```

样例输出1

```
1 | 206749456
```

样例输入2

```
1 | 3 2 3 998244353 3 2 1
```

样例输出2

```
1 | 2810123
```

样例解释

对于第一组样例，集合 S 为 $\{803329205, 1, 499122177\}$ ，其中 499122177 为 $\frac{1}{2}$ 在 $\text{mod } 998244353$ 意义下的数。答案为

$$(1 + 803329205 \times 1 + 803329205 \times 499122177 + 1 \times 499122177) \bmod 998244353 = 206749456$$

对于第二组样例，集合为 S
 $\{316175663, 423574288, 286678054, 286678054, 1, 499122177, 332748118\}$ 。

数据范围

对于5%的数据， $n \leq 15, m = 1, k \leq 5$ 。

对于另外15%的数据， $\sum m_i + k \leq 10^6$ 。

对于另外20%的数据， $n \leq 2 \times 10^5, k = 0$ 。

对于另外20%的数据， $n = 0$ 。

对于另外30%的数据， $n \leq 2 \times 10^5$ 。

对于100%的数据， $0 \leq n \leq 10^6, 1 \leq m_i \leq 10^9, 10^8 \leq P \leq 10^9, 0 \leq a_i, k < P$ 。

铁道

题目描述

有一高楼总共有 n 层，需要设计若干台电梯，为了上下楼的方便在任意两层之间都有至少一部电梯直达，同时为了效率每部电梯最多停4层，问最少需要多少部电梯才能满足上述条件，并给出具体方案。

形式化地，给定 n ，你需要给出最少的电梯数量 m 以及电梯停靠方案 $1 \leq a_i \leq b_i \leq c_i \leq d_i \leq n$ 使得 $\forall 1 \leq i < j \leq n, \exists 1 \leq k \leq m, (i, j) \in \{(a_k, b_k), (b_k, c_k), (c_k, d_k)\}$ 。

输入格式

一行一个整数代表 n 。

输出格式

第一行一个整数代表总电梯数量 m 。

之后 m 行每行4个递增整数 a_i, b_i, c_i, d_i 表示第 i 个电梯的停靠方案。

样例

样例输入1

```
1 | 3
```

样例输出1

```
1 | 2
2 | 1 2 3 3
3 | 1 3 3 3
```

样例解释

我们允许 $a_i = b_i = c_i = d_i$ ，这代表电梯停靠层数小于4层，但是**你仍然需要输出4个数字**。

在样例一中，第一个电梯停靠1, 2, 3层，第二个电梯停靠1, 3层。

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

也是样例一的合法输出，但是你只能得到90分。

数据范围

对于10%的数据， $n \leq 10$ 。

对于另外20%的数据， $n \leq 100$ 。

对于另外30%的数据， $n \leq 1000$ 。

对于100%的数据， $2 \leq n \leq 3000$

评分标准

假设最好的方案使用了 ans 台电梯，在你的方案**合法**的条件下，令你的答案为 res ，则对于每个测试点你的得分是 $100 - \min(10, res - ans) * 10$ 。如果你的方案不合法，则得分为0。

启动

题目描述

对于正整数 $n \geq 3$ ，定义 $f(n)$ 为最大的小于 $n - 1$ 的整数 k 使得 $k^2 \equiv 1 \pmod{n}$ 。例如 $f(15) = 11$ 。

共有 q 次询问，对每次询问 n 你需要求出 $f(n)$ 。

输入格式

第一行一个整数 q ，代表询问次数。

接下来 q 行，每行一个整数 n ，代表询问。

输出格式

共 q 行，每行 1 个整数，代表 $f(n)$ 。

样例

样例输入1

| | |
|---|----|
| 1 | 8 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 6 | 7 |
| 7 | 8 |
| 8 | 9 |
| 9 | 10 |

样例输出1

| | |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 5 |
| 7 | 1 |
| 8 | 1 |

样例说明

注意到 $5 * 5 = 25 \equiv 1 \pmod{8}$ ，故 $f(8) = 5$ ，可以证明没有其他更大的满足条件的整数。

数据范围

对于 5% 的数据， $n \leq 10^5$ 。

对于另外 10% 的数据， $n \leq 10^9$ 且 n 是质数。

对于另外 10% 的数据， $n \leq 10^9$ 且 n 是质数的幂。

对于另外 70% 的数据， $n \leq 10^9$ 。

对于 100% 的数据, $1 \leq q \leq 10^3, 3 \leq n \leq 10^{18}$ 。

保证数据**随机**。