
谜之阶乘 (factorial)

【题目描述】

namespace_std 有一个正整数 n 。

他觉得这个数的性质非常好，因为这个数可以表示成若干对 $a_i!$ 和 $b_i!$ 的商，即 $\frac{a_i!}{b_i!}$ 。

很不幸，namespace_std 遗忘了 a_i 和 b_i ，因此他可能需要你计算出所有合法的正整数对 (a_i, b_i) 。

namespace_std 不仅遗忘了 a_i 和 b_i ，他对 n 的记忆也不太清晰。所以他会向你询问若干个 n 来确定自己的记忆是否存在差错。

【输入格式】

从文件 *factorial.in* 中读入数据。

输入的第一行是询问组数 T ，保证 $1 \leq T \leq 10^4$ 。

接下来 T 行，每行一个正整数 n ($1 \leq n \leq 10^{18}$)，表示本组询问中的 n 。

【输出格式】

输出到 *factorial.out* 中。

对于每组询问，分别输出答案。

输出的第一行是一个正整数 c ，表示合法的 (a_i, b_i) 对的个数。特别地，如果这样的 (a_i, b_i) 对有无限个，你需要输出一行 **-1** 并忽略以下内容。

接下来 c 行，每行输出两个正整数 a_i 和 b_i ，表示你找到的一组解。

为保证答案唯一，请按照 a_i 从小至大的顺序输出。不难发现当 a_i 固定时不可能存在超过一个满足要求的 b_i 。

【输入样例 1】

见选手目录下的 *factorial/factorial1.in*。

```
3
1
6
120
```

【输出样例 1】

见选手目录下的 *factorial/factorial1.ans*。

```
-1
2
3 1
6 5
3
5 1
6 3
120 119
```

【样例 2】

见选手目录下的 *factorial/factorial2.in* 和 *factorial/factorial2.ans*。

【数据规模与约定】

对于 5% 的数据， $1 \leq T, n \leq 10$ 。

对于 10% 的数据， $1 \leq T, n \leq 100$ 。

对于 20% 的数据， $1 \leq T, n \leq 10000$ 。

对于 40% 的数据， $1 \leq n \leq 10^9$ 。

对于 60% 的数据， $1 \leq n \leq 10^{12}$ 。

对于另外 20% 的数据， $T = 1$ 。

对于 100% 的数据， $1 \leq T \leq 10^4$ ， $1 \leq n \leq 10^{18}$ 。