D . 蛋糕通信 / Cake communication

时间限制： 2.0 秒

本题为**通信题**。

请务必在每次输出后刷新标准输出缓冲区。各编程语言的刷新方式如下：

- 对于 C/C++，请使用 `fflush(stdout)` 或 `cout.flush()`；
- 对于 Java，请使用 `System.out.flush()`；
- 对于 Python，请使用 `sys.stdout.flush()`。

【题目描述】

小 L 与小 S 是一对关系要好的朋友。

有一天，小 L 看到了一个正整数 n，她需要**秘密**地将 n 的值传递给小 S。

小 L 有 26 块蛋糕，每块蛋糕上分别写着一个小写英文字母，且不同蛋糕上标有的字母不同。此外，她还有 26 个外表完全相同的盒子。

小 L 可以选择若干块蛋糕放进一些盒子内，然后将它们寄给小 S，其中：

- 由于盒子的外表完全相同，因此盒子之间是**无序的**。
- 一个盒子内的蛋糕之间也是**无序的**。
- 小 L 邮寄的每个盒子内应至少有 1 块蛋糕。

请帮助小 L 与小 S 制定传递策略，使得小 S 可以根据小 L 邮寄的盒子与蛋糕推断出小 L 看到的正整数 n。

【实现细节】

你的程序将会**独立运行两次**，其中第一次将扮演小 L，需要根据看到的正整数 n 向小 S 传递若干个盒子。每个盒子内的蛋糕由一个包含小写字母的字符串表示；第二次将扮演小 S，需要根据收到的盒子与蛋糕推断出小 L 看到的正整数 n。交互器将在你的程序之间传递信息。

在第一次运行中，你的程序会首先收到正整数 n，然后你的程序需要输出若干个不含重复字符的字符串 $s_1, s_2, \ldots, s_m$，从而向第二次运行传递信息。

在第二次运行中，你的程序会从交互器收到来自第一次运行的信息，即被打乱顺序后的字符串 $s'_{p_1}, s'_{p_2}, \ldots, s'_{p_m}$，其中每个字符串内部字符的顺序也会被打乱。你需要根据这些信息推断出其对应正整数 n 的内容。

【输入输出格式】

每个测试点中包含多组测试数据。输入的第一行包含两个正整数 T, r ($1 \le T \le 1000, r \in \{1, 2\}$)，表示数据组数与阶段编号。对于每组测试数据：

- 若 $r = 1$，则输入格式如下：

  - 第一行输入一个正整数 n $(1 \le n \le 10^{16})$，表示待传递的正整数 n。

  你需要按如下格式输出：

  - 首先输出一行一个正整数 m $(1 \le m \le 26)$，代表需要的盒子数量。
  - 接下来 m 行，第 i 行输出一个仅包含小写字母的字符串 s_i，代表每个盒子内包装的蛋糕。由于每种字母的蛋糕只有一个，因此你需要保证每个小写字母**在所有** m **个字符串中总共**出现至多 1 次。

  对于每一组测试数据，你必须输出所有字符串并刷新标准输出缓冲区后，才能继续输入下一组测试数据。

- 若 $r = 2$，则输入格式如下：

  - 第一行输入一个正整数 m，表示盒子个数。
  - 接下来 m 行，第 i 行包含一个字符串 s'_{p_i}，代表每个盒子内装有的蛋糕。所有字符串的顺序与每个字符串内的字符顺序相较第一次运行**可能不同**。

  你需要输出一行一个正整数 n，代表小 L 看到的正整数。
  对于每一组测试数据，你必须输出正整数 n 并刷新标准输出缓冲区后，才能继续输入下一组测试数据，并且单个测试点内测试数据输入的先后顺序相较第一次运行时**可能不同**。

【样例】

第一次运行时，输入如下：

```
4 1
26
315
20170922
1192018010501
```

小 L 可以按如下输出邮寄蛋糕：

```
1
cake
1
nekomura
2
just
konami
3
sut
```

```
e
ia
```

根据上述输出，第二次运行时，一种可能的输入如下：

```
4 2
2
just
monika
1
acek
1
ramukone
3
e
ust
ia
```

通过心灵感应，小 S 推断出四组测试数据的 n，按如下格式输出：

```
20170922
26
315
1192018010501
```

注意小 L 不能按如下方式邮寄蛋糕：

```
1
pudding
```

字母 d 的出现次数大于 1，不符合条件。

【程序测试方式】

下发的脚本文件 cake_testing_tool.py 可以协助你进行本地测试，并打印详细的交互过程。测试时，将脚本文件与你编译生成的可执行文件置于同一文件夹下，然后在终端内运行以下命令：

```
python3 cake_testing_tool.py [--quiet] <data_file> <program>
   <arguments>
```

其中：

- -q, --quiet 为可选参数，使用该参数后，测试工具不会打印详细的交互过程。

- `data_file` 为你提供给测试脚本的输入数据文件路径。该文件需满足以下格式：
  - 第一行包含两个非负整数 $T, seed$，分别代表测试数据组数，与用于打乱测试数据的随机数种子。若指定 $seed = 0$，则代表不进行打乱。
  - 每组数据的格式与前文描述的“第一次运行”的输入格式完全相同。
- `program` 为你编译生成的可执行文件路径。
- `arguments` 为传递给该可执行文件的额外命令行参数。

关于测试脚本的更多实现细节，请直接查阅其源代码。

注意：

1. 该测试工具与实际评测时的交互库实现**不完全相同**，本地测试结果不具有终断性，仅作调试阶段参考。
2. 测试工具**不会监控**程序的时间与空间消耗，无法用于测试是否符合题目的时空限制。