三色花园（garden）

这是一道通信题。

【题目描述】

Alice 经营着一个美丽的花园，其中有 n 个供游客用餐与休息的休息厅，编号从 1 到 n。这些休息厅之间连接着一些花廊，供游客们赏花与在休息厅之间通行。花廊里种植着三种不同颜色的花朵，每座花廊里恰好种植着其中一种颜色的花。为了保证游览的秩序，对于**任意两个**不同的休息厅 u, v $(1 \le u < v \le n)$，休息厅 u 与休息厅 v 之间**恰好**连接着一个只能**单向通行**的花廊，即花廊的通行方向要么为 $u \to v$，要么为 $v \to u$。

游客在花园里赏花时，会选择一系列互不相同的休息厅，并按照顺序环绕这些休息厅一圈。具体地，游客会选择 l $(l \ge 3)$ 个休息厅 $a_1, \ldots, a_l$，然后按照 $a_1 \to a_2 \to \cdots \to a_l \to a_1$ 的顺序游览这些休息厅间的花廊。由于花廊的通行方向限制，可行的游览方式很少。定义 $a_1, \ldots, a_l$ 为一种**游览方式**当且仅当 $l \ge 3$，$a_1, \ldots, a_l$ 互不相同，且对于所有 $1 \le i \le l$，连接休息厅 a_i 与 $a_{(i \bmod l)+1}$ 的花廊的通行方向为 $a_i \to a_{(i \bmod l)+1}$。两种游览方式 $a_1, \ldots, a_l$ 与 $b_1, \ldots, b_l$ **本质不同**当且仅当两种可行游览方式经过的花廊不同，或**环绕的路径不同**，即 $l_1 \ne l_2$ 或对于任意 $0 \le x < l_1$，均存在 $0 \le y < l_1$ 满足 $a_y \ne b_{(y+x-1) \bmod l_1+1}$。

对于一种游览方式，游客将会观赏到不同颜色的花朵。若游览方式中经过的花廊包括所有三种颜色的花朵，则游客会对此感到满意，否则会对此感到不满。为了确保游客的游览体验，对于每个休息厅，经过该休息厅的游览方式中，**至多只有 k 种本质不同的游览方式会使游客感到不满**。

Bob 也想经营一个和 Alice 一样的花园，于是他找来了 Alice 的花园设计图纸，同样修建了一个有 n 个休息厅的花园。然而，Alice 丢失了花廊的设计图纸。Bob 希望他修建的花园的**花廊的布局与通行方向均与 Alice 的花园相同**，且同样对于每个休息厅，经过该休息厅的游览方式中，至多只有 k 种本质不同的游览方式会使游客感到不满。**注意：花廊种植的花朵的颜色不需要完全相同，只需要满足上述条件即可。**

为了尽快完成修建，Alice 需要向 Bob 发送尽可能少的信息，让 Bob 能够按照要求建立花廊。具体地，Alice 可以向 Bob 传输一个长度为 l 的 01 串，Bob 需要通过这个 01 串修建他的花园的花廊，即确定每两个休息厅间花廊的通行方向与种植的花朵的颜色。你需要帮助 Alice 发送尽可能少的信息，并帮助 Bob 通过这些信息修建他的花园的花廊。

【实现细节】

选手不需要，也不应该实现 `main` 函数。

选手需要确保提交的程序包含头文件 `garden.h`，即在程序开头加入以下代码：

```
1 #include "garden.h"
```

选手需要在提交的程序源文件中实现以下三个函数：

```
int init(int n, int k);
```

- n, k 分别表示休息厅的数量与使游客感到不满的游览方式的数量上限。
- 该函数需要返回一个非负整数 l，表示 Alice 向 Bob 传输的 01 串的长度。选手需要保证 $0 \le l \le 1.5 \times 10^5$。
- 对于每个测试点，该函数会在**程序第一次开始运行时**被交互库调用**恰好一次**。

```
std::string send_message(int n, int k,
  std::vector<std::vector<std::pair<bool, int>>> e);
```

- n, k 分别表示休息厅的数量与使游客感到不满的游览方式的数量上限。
- 对于 $0 \le i \le n-2$，$0 \le j \le n-i-2$，$e_{i,j}$ 表示 Alice 的花园中连接休息厅 $i+1$ 与休息厅 $i+j+2$ 的花廊的通行方向与种植的花朵的颜色。具体地，$e_{i,j}$ 是一个二元组，
  - 若 $e_{i,j}$ 的第一个元素为 `true`，则该花廊的通行方向为 $i+1 \to i+j+2$，否则为 $i+j+2 \to i+1$；
  - $e_{i,j}$ 的第二个元素是一个 $\{0,1,2\}$ 中的非负整数，表示连接休息厅 $i+1$ 与休息厅 $i+j+2$ 的花廊种植的花朵的颜色。
- 该函数需要返回一个长度为 l 的 01 字符串 s，表示 Alice 向 Bob 传输的 01 串。选手需要保证 s 的长度与 `init` 函数的返回值相同。
- 对于每个测试点，该函数会被**程序第一次运行时**被交互库调用**恰好** 10 **次**，且传入该函数中的变量 n, k 与传入 `init` 函数中的变量 n, k 相同。

```
std::vector<std::vector<std::pair<bool, int>>>
  build_flower_garden(int n, int k, std::string s);
```

- n, k, s 分别表示休息厅的数量、使游客感到不满的游览方式的数量上限与 Alice 向 Bob 传输的 01 串。
- 该函数需要返回 Bob 修建的花园中花廊的通行方向与种植的花朵的颜色，表示方法与传入 `send_message` 函数中的变量 e 一致。选手需要保证：
  - e 的长度为 $n-1$；
  - 对于 $0 \le i \le n-2$，e_i 的长度为 $n-i-1$；
  - 对于 $0 \le i \le n-2$，$0 \le j \le n-i-2$，$e_{i,j}$ 的第二个元素是一个 $\{0,1,2\}$ 中的非负整数。
- 对于每个测试点，该函数会被**程序第二次运行时**被交互库调用**恰好** 10 **次**，且传入该函数中的变量的 n, k 与程序第一次运行时传入 `init` 函数中的变量 n, k 相同，且第 c $(1 \le c \le 10)$ 次传入该函数中的变量 s 与程序第一次运行时第 c $(1 \le c \le 10)$ 次 `send_message` 函数的返回值相同。

- 选手还需要保证：
  - 对于 $1 \le c \le 10$，$0 \le i \le n-2$，$0 \le j \le n-i-2$，程序第一次运行时第 c 次传入 `send_message` 函数的 $e_{i,j}$ 的第一个元素与程序第二次运行时第 c 次 `build_flower_garden` 函数的返回值的 $e_{i,j}$ 的第一个元素相同。
  - 按照该函数返回值修建的花廊满足对于每个休息厅，经过该休息厅的游览方式中至多只有 k 种本质不同的游览方式会使游客感到不满。

注意：在任何情况下，交互库运行所需时间均不会超过 0.2 秒，所用内存为固定大小，且均不超过 64 MiB。

【测试程序方式】

试题目录下的 `stub.cpp` 是交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此选手的解法不应该依赖交互库实现。

选手可以在本题目录下使用如下命令进行测试：

```
bash run.sh garden.cpp
```

- 上述脚本将从**标准输入**读入以下格式的数据：
  - 输入的第一行包含三个正整数 t, n, k，分别表示测试数据组数，休息厅的数量与使游客感到不满的游览方式的数量上限。
  - 输入的第 $i+1$ $(1 \le i \le n-1)$ 行包含 $n-i$ 个非负整数，其中第 j $(1 \le j \le n-i)$ 个非负整数表示连接休息厅 i 与休息厅 $i+j$ 的花廊的通行方向，若为 1 则该花廊的通行方向为 $i+1 \to i+j+2$，否则为 $i+j+2 \to i+1$；
  - 输入的第 $i+n$ $(1 \le i \le n-1)$ 行包含 $n-i$ 个非负整数，其中第 j $(1 \le j \le n-i)$ 个非负整数表示连接休息厅 i 与休息厅 $i+j$ 的花廊种植的花朵的颜色。
- 上述脚本将输出以下格式的数据至**标准输出**：
  - 输出的第一行包含一个非负整数 l，表示 Alice 向 Bob 传输的 01 串的长度。
  - 输出的第二行包含一个 $[0, 1]$ 间的实数 p，表示该测试点的得分比例。

【样例 1】

见题目目录下的 ***1.in*** 与 ***1.ans***。
该样例满足子任务 1 的约束条件。

【样例 2】

见题目目录下的 ***2.in*** 与 ***2.ans***。
该样例满足子任务 2 的约束条件。

【样例 3】

见题目目录下的 ***3.in*** 与 ***3.ans***。
该样例满足子任务 3 的约束条件。

【附加文件说明】

在附加文件中：

1. `stub.cpp` 是提供的交互库参考实现。
2. `garden.h` 是头文件，选手不用关心具体内容。
3. `template_garden.cpp` 是提供的示例代码，选手可参考并实现自己的代码。
4. `bigint.cpp` 是提供的高精度模板，选手可参考并使用。
5. `bigint.pdf` 是提供的高精度模板使用说明。

【子任务】

对于所有测试数据，均有：

- $t = 10$，$n = 300$，$0 \le k \le 2$；
- 所有花廊的通行方向为 $\{0, 1\}$ 中的非负整数；
- 所有花廊种植的花朵的颜色为 $\{0, 1, 2\}$ 中的非负整数。

子任务编号	分值	$k =$
1	30	0
2		1
3	40	2

【评分方式】

注意：

- 选手不应当通过非法方式获取交互库的内部信息，如直接与标准输入、输出流进行交互。此类行为将被视为作弊；
- **最终的评测交互库与样例交互库的实现不同。**

本题首先会受到和传统题相同的限制，例如编译错误会导致整道题目得 0 分，运行时错误、超过时间限制、超过空间限制等会导致相应测试点得 0 分等。选手只能在程序中访问自己定义的变量以及交互库给出的变量，尝试访问其他地址空间将可能导致编译错误或运行错误。

若 `init` 函数的返回值或 `send_message` 函数的返回值或 `build_flower_garden` 函数的返回值不合法，则相应测试点得 0 分。

在上述条件基础上：

- 对于每个测试点，设 B 为该测试点所属子任务的参数，l 为 `init` 函数的返回值，则该测试点的得分比例如下表所示：

子任务编号	$B=$
1	$6,200$
2	$6,800$
3	$8,000$

$l\in$	得分比例
$[B+5\times10^4,1.5\times10^5]$	$(5+15\times(1.5\times10^5-l)\div(10^5-B))\,\%$
$[B+10^4,B+5\times10^4)$	$(20+(B+5\times10^4-l)\div2000)\,\%$
$[B+2\times10^3,B+10^4)$	$(40+(B+10^4-l)\div400)\,\%$
$(B,B+2\times10^3)$	$(60+(B+2\times10^3-l)\div50)\,\%$
$[0,B]$	100%

- 每个测试点的得分为得分比例与该测试点所属子任务的分值的乘积。