Задача A. Файловое дерево

Имя входного файла:	`k.in`
Имя выходного файла:	`k.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У студента Альберта завтра очень ответственный экзамен. Посмотрев на часы, он понял, что не успевает выучить N билетов. И тогда он решился пойти на крайние меры — сделать шпаргалку. Опасаясь за целостность своих карманов (посмотрите на ограничения для N и угадайте, в каком вузе учится Альберт!), он отказался от идеи написать ответы на бумаге. Вместо этого он решил разместить ответы на своём телефоне, который работает под управлением операционной системы CheatOS. Он уже подготовил N файлов: `FILE_1`, `FILE_2`, ..., `FILE_`N, и теперь ему необходимо разместить их в каталоге, который он в целях конспирации назвал `tmp`.

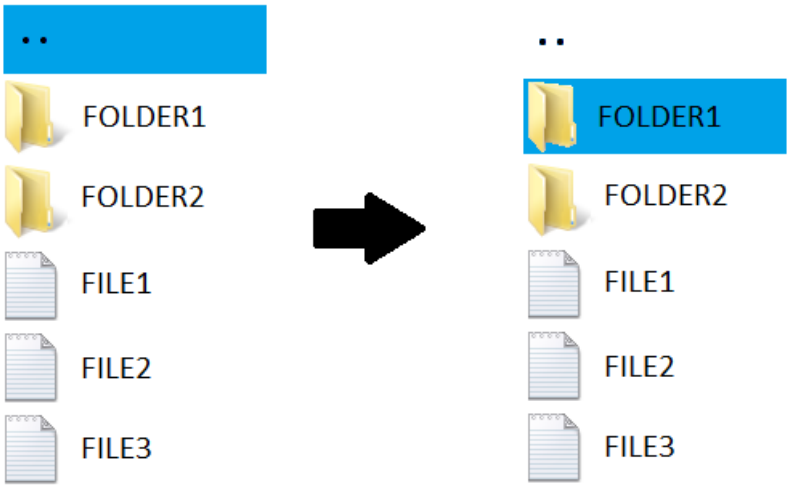
Задача состоит в том, чтобы создать некоторое дерево подкаталогов в каталоге `tmp` и разместить в нём файлы так, чтобы доступ к *самому далёкому* файлу осуществлялся бы за минимально возможное количество действий: Альберт не хочет быть пойманным на списывании из-за промедления в поиске нужного файла.

Более формально, при работе с файловой системой операционной системы CheatOS на экране телефона в каждый момент отображается содержимое текущего каталога в виде списка элементов, расположенных в следующем порядке: сперва символ «`..`» перехода в родительскую папку, затем список всех размещённых непосредственно в этом каталоге подкаталогов и файлов. Файлы и каталоги, содержащиеся в подкаталогах текущего каталога, не отображаются.

В каждый момент на один из элементов отображаемого списка (файл или папку) указывает курсор.

Нас будут интересовать два вида действий, которые потребуются Альберту на экзамене:

1. Войти в дочерний каталог, на который указывает курсор. При этом этот каталог становится текущим и всё его содержимое отображается на экране телефона так, как описано выше. Сразу после этого действия курсор указывает на символ «`..`».
2. Сдвинуть курсор на следующий по списку элемент (см. рисунок).



Изначально текущий каталог — `tmp`, курсор указывает на символ «`..`».

Расстоянием до файла будем называть минимальное суммарное число действий обоих видов, необходимое для того, чтобы курсор на экране стал указывать на этот файл.

Альберт просит вас помочь ему построить дерево каталогов и разместить в них все N файлов так, чтобы расстояние до самого далёкого файла было минимально возможным.

Времени до экзамена остается совсем немного, создавать каталоги на телефоне — трудоёмкое занятие, поэтому среди всевозможных деревьев каталогов с минимальным расстоянием он просит вас найти дерево, содержащее как можно меньше каталогов.

В файловой системе допускаются каталоги, не содержащие файлов, каталоги, не содержащие дочерних каталогов, а также пустые каталоги. Каталоги должны иметь названия `FOLDER_1`, `FOLDER_2`, ..., `FOLDER_`M.

Каждый из файлов, подготовленных Альбертом, должен быть размещён в построенном дереве каталогов ровно один раз.

В случае, если удовлетворяющих условию деревьев несколько, выведите любое.

Формат входных данных

На вход программе подаётся одно целое число N ($1 \leqslant N \leqslant 10^5$).

Формат выходных данных

В первой строке выведите через пробел два числа: расстояние до самого далёкого файла и минимальное количество каталогов M, которое необходимо создать, чтобы добиться такого ответа.

В следующей строке выведите содержимое каталога `tmp`, а в следующих M строках выведите содержимое каталогов `FOLDER_1`, `FOLDER_2`, ..., `FOLDER_`M соответственно. Содержимое каталога выводится в следующем формате: сперва выводится общее число файлов и подкаталогов, содержащихся непосредственно в данном каталоге, а затем список файлов и папок в том порядке, в котором они располагаются в каталоге. В этом списке файл под номером K обозначается `FILE_`K, а папка под номером K обозначается `FOLDER_`K. Символ «`..`» выводить не требуется.

Для лучшего понимания формата вывода рекомендуется изучить примеры.

Примеры

k.in	k.out
3	3 0 3 FILE_1 FILE_2 FILE_3
5	4 1 4 FOLDER_1 FILE_1 FILE_2 FILE_3 2 FILE_4 FILE_5