Задача A. Палиндромное кодирование

Имя входного файла:	`palin.in`
Имя выходного файла:	`palin.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В одной секретной лаборатории разработали новый сверхсекретный способ архивации данных. Этот метод настолько уникален, что позволяет все данные сжимать примерно в 2 раза. Заключается он в следующем. Шифруемый файл представляется в виде последовательности из нулей и единиц. Потом из этой последовательности выбирается непустая подпоследовательность, которая является палиндромом. Затем из оставшейся части выбирается непустая последовательность, которая является палиндромом и т.д. Таким образом, из исходной последовательности остаётся только несколько палиндромов. Но ведь от палиндромов теперь можно хранить только одну половину!

Сразу видно, насколько хорош этот метод. Однако есть и небольшие трудности — чем на большее число палиндромов приходится разбивать строку при архивации, тем сложнее потом разархивировать. Поэтому требуется программа, которая разобьёт последовательность на минимальное количество палиндромов.

Однако комиссия по проверке не согласилась принимать такой алгоритм, сказав, что программа архивации работает нестабильно и по-разному архивирует один и тот же файл. Для унификации был придуман следующий метод: изо всех способов разбить на минимальное число палиндромов выбирается такой, в котором первый палиндром лексикографически наименьший. Изо всех таких — тот, в котором второй наименьший. И так далее.

Ваша задача — написать программу для архивации данных описанным выше способом.

Формат входного файла

В первой и единственной строке входного файла содержится непустая строка из нулей и единиц длиной не более 3 000 000 символов.

Формат выходного файла

В первой строке выведите число N — минимальное число палиндромов. В следующих N строках выведите сами палиндромы, по одному на строке.

Пример

palin.in	palin.out
1001	1 1001

Примечание

Если ваша программа пройдёт тесты со строками, длины которых не больше 3000, то она получит не меньше 50% баллов.

Задача B. Б-деревья

Имя входного файла:	`btrees.in`
Имя выходного файла:	`btrees.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

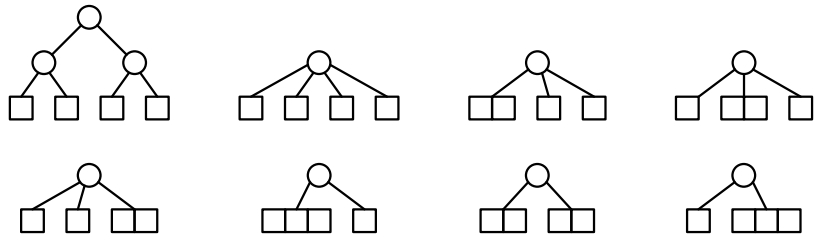
Если б дерево умело говорить...

Б-дерево — структура для хранения данных во вторичной памяти (например, на жёстком диске). Б-дерево обладает несколькими свойствами:

1. У каждой вершины, которая не является листом, количество детей не меньше t и не больше $2 \cdot t$, где t — параметр, который называется *степенью ветвления*.
2. В каждом листе хранится несколько ключей — от $t - 1$ до $2 \cdot t - 1$ штук.
3. Расстояние от всех листьев до корня одинаково.

Обратите внимание, что корень может являться листом.

Два Б-дерева считаются различными, если они различны как графы с помеченными вершинами, или если вершина с одинаковыми пометками содержит различное количество ключей. Например, существует всего 8 различных Б-деревьев с четырьмя ключами и со степенью ветвления 2:



Посчитайте количество различных Б-деревьев с n ключами в листьях и степенью ветвления t.

Формат входного файла

В первой строке находятся два натуральных числа n и t — количество ключей в листьях и степень ветвления, соответственно ($1 \leqslant n \leqslant 500$, $2 \leqslant t \leqslant 10^9$).

Формат выходного файла

В первой строке выведите единственное число без ведущих нулей — количество Б-деревьев с n ключами в листьях и степенью ветвления t.

Примеры

btrees.in	btrees.out
4 2	8
20 2	17220826

Задача C. Динамический массив

Имя входного файла:	`dynarray.in`
Имя выходного файла:	`dynarray.out`
Ограничение по времени:	15 секунд
Ограничение по памяти:	256 мегабайт

Требуется реализовать структуру данных, которая представляет из себя динамический массив, поддерживающий следующие операции:

1. Присвоить числу, которое стоит на позиции u, значение p.
2. Вставить число p после позиции u (но перед позицией $u + 1$, если она существует).
3. Сообщить, сколько чисел, не превосходящих p, стоит на позициях с u по v включительно.

Элементы массива нумеруются с единицы.

Формат входного файла

В первой строке находятся два натуральных числа n и m — начальное количество элементов в массиве и количество операций ($1 \leqslant n \leqslant 200\,000$, $1 \leqslant m \leqslant 200\,000$). Во второй строке находится n целых чисел a_1, a_2, ..., a_n — элементы исходного массива ($0 \leqslant a_i \leqslant 10^6$). Следующие m строк содержат описания операций в формате "`1 u p`" ($u \geqslant 1$, u не превосходит текущего размера массива, $0 \leqslant p \leqslant 10^6$), "`2 u p`" ($u \geqslant 0$, u не превосходит текущего размера массива, $0 \leqslant p \leqslant 10^6$) или "`3 u v p`" ($1 \leqslant u \leqslant v$, v не превосходит текущего размера массива, $0 \leqslant p \leqslant 10^6$).

Формат выходного файла

В выходном файле должны находиться ответы на запросы — по одному в строке.

Пример

dynarray.in	dynarray.out
5 5 1 3 5 4 2 3 2 4 4 1 3 3 3 2 4 4 2 2 10 3 2 4 4	2 3 2