

Задача А. Футболки

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

На зарядку сегодня утром пришло N ЛКШат, они построились в ряд. Разумеется, ребята ходят в разноцветных футболках. Никита Павлович, наш многоуважаемый физрук, заметил, что можно попросить некоторых ребят присесть, и тогда для ребят, которые останутся стоять, будет выполнено следующее: последовательность цветов их футболок при перечислении слева направо будет такой же как и последовательность при перечислении справа налево, то есть будет *палиндромом*.

Например, если на зарядку пришли Ксюша в зеленой футболке, Слава в желтой, Артём в красной и Вася в зеленой, то можно попросить присесть Славу, тогда последовательность цветов будет «зеленый, красный, зеленый» как слева направо, так и справа налево. Аналогично можно попросить присесть Артёма (последовательность будет «зеленый, желтый, зеленый»), Славу и Артёма одновременно или любой набор из троих ребят. Таким образом, всего есть 7 способов добиться того, чтобы последовательность цветов была палиндромом.

Помогите Никите Павловичу найти количество способов попросить некоторых ЛКШат присесть, чтобы последовательность цветов футболок оставшихся стоять была палиндромом. Поскольку это число может быть очень большим, выведите его по модулю 10^9 .

Формат входных данных

Первая строка входного файла содержит число N — количество ЛКШат, пришедших на зарядку ($1 \leq N \leq 2000$). Вторая строка содержит N целых чисел, каждое из которых задает цвет футболки ЛКШонка и изменяется в пределах от 1 до 10^9 . Разные цвета задаются разными числами, а одинаковые — одинаковыми.

Формат выходных данных

Выведите в выходной файл одно число — искомое количество способов по модулю 10^9 .

Примеры

<code>tshirts.in</code>	<code>tshirts.out</code>
4 1 2 3 1	7

Задача В. НВП

Имя входного файла: `lis.in`
Имя выходного файла: `lis.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (ka_i + b) \bmod m$. Найдите её наибольшую возрастающую подпоследовательность.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k, b, m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4$, $0 \leq k < m$, $0 \leq b < m$, $0 \leq a_1 < m$).

Формат выходных данных

Требуется вывести длину НВП.

Примеры

<code>lis.in</code>	<code>lis.out</code>
5 41 2 1 100	3

Замечание

В данном примере последовательность состоит из 5 элементов: $a_1 = 41$, $a_{i+1} = (2a_i + 1) \bmod 100$, то есть последовательность имеет вид 41, 83, 67, 35, 71.

Задача С. Дубы

Имя входного файла: `oaks.in`
Имя выходного файла: `oaks.out`
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 256 мегабайт

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Примеры

<code>oaks.in</code>	<code>oaks.out</code>
5	2
3 2 4 8 5	2
	4

Задача D. Транзисторы над Пекином

Имя входного файла: `transistor.in`
Имя выходного файла: `transistor.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме N этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа f транзистор обязательно разобьется при падении с этажа $f + 1$. Дополнительно поставлено условие, что разрешено проведение не более чем K испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

Формат входных данных

В первой и единственной строке входного файла содержатся два целых положительных числа N и K не превосходящих 2000.

Формат выходных данных

Выведите единственное число - ответ на поставленную задачу. Если для данных N и K возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите -1 .

Примеры

<code>transistor.in</code>	<code>transistor.out</code>
4 2	-1
4 3	2

Задача Е. Покраска забора

Имя входного файла: `paint.in`
Имя выходного файла: `paint.out`
Ограничение по времени: 0.7 секунда
Ограничение по памяти: 256 мегабайт

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

paint.in	paint.out
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A