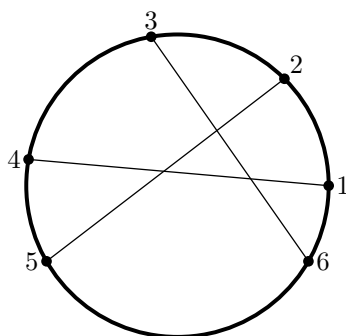


Задача Е. Хорды

Имя входного файла: chords.in
Имя выходного файла: chords.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта
Максимальная оценка за задачу: 100 баллов

На окружности отмечено $2n$ различных точек, пронумерованных от 1 до $2n$ против часовой стрелки. Петя нарисовал n хорд, i -я из которых соединяет точки с номерами a_i и b_i . При этом каждая точка является концом ровно одной хорды.



Теперь Петя заинтересовался, сколько пар хорд пересекаются. Помогите ему определить это количество.

Формат входного файла

Первая строка входного файла содержит целое число n — количество проведенных хорд ($1 \leq n \leq 100\,000$). Следующие n строк содержат по два целых числа — a_i и b_i .

Формат выходного файла

Выведите одно число — количество пар хорд, которые пересекаются.

Примеры

chords.in	chords.out
3 1 4 2 5 3 6	3
2 1 2 3 4	0
2 1 4 2 3	0

Примечание

Решения, которые работают только для $n < 2000$, будут оцениваться из 40 баллов.

Задача F. Дубы

Имя входного файла:	<code>oaks.in</code>
Имя выходного файла:	<code>oaks.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта
Максимальная оценка за задачу:	100 баллов

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 дуб.

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 входного файла

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, приведенные слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходного файла

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 3 2 4 8 5	2 2 4
5 4 5 5 5 6	0
6 1 1 3 3 2 2	-1
6 400 300 310 300 310 500	-1

Примечание

Решения, в которых рассматривается только случай, когда высоты всех дубов попарно различны, будут оцениваться из 50 баллов.

Задача G. Палиндромность

Имя входного файла:	palin.in
Имя выходного файла:	palin.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта
Максимальная оценка за задачу:	100 баллов

Напомним, что *палиндромом* называется строка, которая читается одинаково как слева направо, так и справа налево. Например, палиндромами являются строки «abba» и «madam».

Для произвольной строки s введем операцию *деления пополам*, обозначаемую $half(s)$. Значение $half(s)$ определяется следующими правилами:

- Если s не является палиндромом, то значение $half(s)$ не определено;
- Если s имеет длину 1, то значение $half(s)$ также не определено;
- Если s является палиндромом четной длины $2m$, то $half(s)$ — это строка, состоящая из первых m символов строки s ;
- Если s является палиндромом нечетной длины $2m + 1$, большей 1, то $half(s)$ — это строка, состоящая из первых $m + 1$ символов строки s .

Например, значения $half(\text{inforamatics})$ и $half(i)$ не определены, $half(\text{abba}) = \text{ab}$, $half(\text{madam}) = \text{mad}$.

Палиндромностью строки s будем называть максимальное число раз, которое можно применить к строке s операцию деления пополам, чтобы результат был определен.

Например, палиндромность строк «informatics» и «i» равна 0, так как к ним нельзя применить операцию деления пополам даже один раз. Палиндромность строк «abba» и «madam» равна 1, а палиндромность строки «totottotot» равна 3, поскольку операция деления пополам применима к ней три раза: «totottotot» → «totot» → «tot» → «to».

Рассмотрим все строки длины n , состоящие из строчных латинских букв, палиндромность которых равна p . Ваша задача — найти k -ю в алфавитном порядке среди этих строк.

Формат входного файла

В первой строке входного файла содержатся три целых числа n , p и k — длина и палиндромность строк рассматриваемого множества и номер искомой строки в этом множестве ($1 \leq n \leq 200$; $0 \leq p \leq 8$; $1 \leq k \leq 10^9$).

Гарантируется, что в множестве строк длины n с палиндромностью p содержится не менее k элементов, то есть искомая строка существует.

Формат выходного файла

В выходной файл выведите k -ю в алфавитном порядке строку из множества строк длины n с палиндромностью p .

Примеры

palin.in	palin.out
4 1 1	abba
10 3 490	totottotot
5 0 6597777	olymp