

Задача А. Следующая перестановка

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

Найдите следующую перестановку. Лексикографически первая перестановка является следующей для обратной.

Формат входного файла

В первой строке входного файла записано число N ($1 \leq N \leq 100\,000$) — количество элементов в перестановке. Во второй строке записана перестановка из N чисел.

Формат выходного файла

В выходной файл вывести N чисел — искомую перестановку.

Примеры

<code>nextperm.in</code>	<code>nextperm.out</code>
3 3 2 1	1 2 3
2 1 2	2 1

Задача В. Следующее сочетание

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

Дано множество целых чисел от 1 до N . Рассмотрим подмножество этого множества, состоящее из K элементов, в возрастающем порядке.

Выведите следующее в лексикографическом порядке подмножество из K элементов.

Формат входного файла

В первой строке входного файла содержатся целые положительные числа N и K ($1 \leq K \leq N \leq 50$). Во второй строке содержится K целых чисел от 1 до N в возрастающем порядке — подмножество из K элементов.

Формат выходного файла

Выведите следующее в лексикографическом порядке после данного подмножество из K элементов. Если следующего подмножества нет, выведите 0.

Примеры

<code>nextcomb.in</code>	<code>nextcomb.out</code>
6 4 1 4 5 6	2 3 4 5
6 2 5 6	0

Задача С. Номер по перестановке

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

Дана перестановка из N чисел от 1 до N . Требуется найти её номер в лексикографическом порядке.

Формат входного файла

Во входном файле сначала записано число N ($1 \leq N \leq 12$). В следующей строке записана сама перестановка — N чисел, разделённых пробелами.

Формат выходного файла

В выходной файл нужно вывести единственное число — номер перестановки в лексикографическом порядке.

Примеры

<code>perm.in</code>	<code>perm.out</code>
3 2 1 3	3

Задача D. Перестановка по номеру

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

Демидурги Шамбамбукли и Мазукта уже достигли невероятного мастерства в искусстве сотворения миров. Но им, как никому другому, известно, что нет предела совершенству. Неудивительно, что время от времени они собираются в на скорую руку сотворённом кафе за чашечкой кофе, чтобы вместе постигать вершины своего искусства. По своему небольшому опыту они знают, что самые интересные и сложные закономерности проще всего обнаружить на самых простых примерах.

В этот раз они решили посмотреть, что выйдет из миров, в основу которых положены самые обычные перестановки. Чтобы не упустить ни один из возможных вариантов, демидурги решили использовать перестановки для сотворения миров по очереди, в лексикографическом порядке.

Шамбамбукли и Мазукта уже успешно создали $K - 1$ мир, но тут их настиг творческий кризис. Теперь только вы можете помочь им создать очередной мир и вдохновить их этим поступком на завершение эксперимента. По счастливому совпадению, этот мир как раз должен оказаться Хрымбелем, который демиурги в прошлый раз так и не смогли сотворить.

Формат входного файла

В первой строке входного файла записано число N ($1 \leq N \leq 12$) — количество элементов в перестановке, которая должна быть положена в основу Хрымбеля. Во второй строке число K ($1 \leq K \leq N!$) — номер перестановки.

Формат выходного файла

В выходной файл вывести N чисел через пробел — первооснову Хрымбеля.

Примеры

bynumber.in	bynumber.out
3 1	1 2 3

Задача Е. Сочетание с повторениями по номеру

Имя входного файла: comb2.in
Имя выходного файла: comb2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Сочетанием с повторениями называется способ выбрать из n элементов k , при этом способы, отличающиеся перестановкой элементов, считаются одинаковыми, но один элемент можно выбирать более одного раза. Например, существует всего 6 сочетаний с повторениями из 3 элементов по 2.

В этой задаче мы будем рассматривать все k -элементные сочетания с повторениями множества из n чисел от 1 до n . Естественно, что все эти сочетания можно упорядочить лексикографически как вектора, если считать порядок чисел в сочетании неубывающим. Скажем, при $n = 3$ и $k = 2$ список упорядоченных сочетаний с повторениями будет выглядеть так: (1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3).

В этой задаче требуется найти сочетание с повторениями по лексикографическому номеру (нумерация ведётся с нуля).

Формат входного файла

Три числа n , k и x ($1 \leq k \leq n \leq 30$), x задаёт номер существующего сочетания с повторениями.

Формат выходного файла

Выведите k чисел, задающих требуемое сочетание с повторениями.

Примеры

comb2.in	comb2.out
3 2 1	1 2