

Задача А. Все двоичные строки длины n , содержащие ровно k единиц

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

По данным числам N и K выведите все строки из нулей и единиц длины N , содержащие ровно K единиц, в лексикографическом порядке.

Формат входных данных

Заданы 2 числа: N и K ($0 \leq K \leq N$, $0 \leq N \leq 100$).

Формат выходных данных

Необходимо вывести все строки из нулей и единиц длины N , содержащие ровно K единиц, в лексикографическом порядке. Гарантируется, что размер ответа не превышает 10 мегабайт.

Примеры

<code>combnk.in</code>	<code>combnk.out</code>
4 2	0011 0101 0110 1001 1010 1100

Задача В. Номер по правильной скобочной последовательности

Имя входного файла: `brackets2num.in`
Имя выходного файла: `brackets2num.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана правильная скобочная последовательность, содержащая N открывающих скобок. Найдите её номер в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, учитывая, что «(» < «)». Последовательности нумеруются с 0.

Формат входных данных

Во входном файле задана строка, являющаяся правильной скобочной последовательностью, $1 \leq N \leq 20$.

Формат выходных данных

Выведите единственное число — номер последовательности.

Примеры

<code>brackets2num.in</code>	<code>brackets2num.out</code>
<code>((()))()</code>	3

Задача С. Кувшинки

Имя входного файла: `parenth2.in`
 Имя выходного файла: `parenth2.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

Правильной скобочной последовательностью из двух типов скобок из $2n$ скобок называется такая последовательность круглых и квадратных скобок, которая может встречаться в некотором арифметическом выражении. Например, $() [] ()$ и $((()))$ являются правильными скобочными последовательностями из двух типов скобок, а $((()))]$ и $[])] [$ — нет.

Все правильные скобочные последовательности из двух типов скобок можно упорядочить в лексикографическом порядке, считая, что порядок скобок соответствует их кодам символов: $'(' < '[' < ')' < ']'$. Скажем, при $n = 2$ список упорядоченных правильных скобочных последовательностей из двух типов скобок будет выглядеть так: $((()))$, $()()$, $() []$, $([])$, $[()]$, $[][]$, $[] ()$, $[] []$.

В этой задаче требуется найти правильную скобочную последовательность из двух типов скобок по лексикографическому номеру (нумерация ведется с нуля).

Формат входных данных

Два числа n и x ($1 \leq n \leq 20$), x задаёт номер существующей правильной скобочной последовательности из двух типов скобок.

Формат выходных данных

Выведите строку из $2n$ круглых и квадратных скобок, задающих требуемую правильную скобочную последовательность из двух типов скобок.

Примеры

	<code>parenth2.in</code>	<code>parenth2.out</code>
2 1		$()()$
2 4		$[()]$

Задача D. Предыдущая правильная скобочная последовательность

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

По правильной скобочной последовательности вывести последовательность, предшествующую ей в лексикографическом порядке.

Формат входных данных

На вход подается одна строка, состоящая из $2N$ ($N \leq 10000$) символов - исходная правильная скобочная последовательность.

Формат выходных данных

Вывести в единственной строке искомую последовательность. Если исходная последовательность - первая в лексикографическом порядке, то следует вывести последнюю в лексикографическом порядке правильную скобочную последовательность такой же длины.

Примеры

<code>prev.in</code>	<code>prev.out</code>
<code>()()</code>	<code>((()))</code>
<code>()(())</code>	<code>((()))()</code>

Задача Е. 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 входных данных

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 выходных данных

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 F. Размещение с повторениями по номеру

Имя входного файла: `arrange2.in`
Имя выходного файла: `arrange2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Размещением с повторениями называется способ выбрать из n элементов k , при этом способы, отличающиеся перестановкой элементов, считаются различными, а один и тот же элемент можно выбирать более одного раза. Например, существует всего 9 размещений с повторениями из 3 элементов по 2.

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

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

Формат входных данных

Три числа n , k и x ($1 \leq k \leq n \leq 15$), x задаёт номер существующего размещения с повторениями.

Формат выходных данных

Выведите k чисел, задающих требуемое размещение с повторениями.

Примеры

<code>arrange2.in</code>	<code>arrange2.out</code>
3 2 1	1 2

Задача G. Номер по сочетанию с повторениями

Имя входного файла: `combr2.in`
Имя выходного файла: `combr2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

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 лексикографический номер по сочетанию с повторениями (нумерация ведётся с нуля).

Формат входных данных

В первой строке даны два числа n , k ($1 \leq k \leq n \leq 30$). Во второй строке даны k чисел, задающих требуемое сочетание с повторениями.

Формат выходных данных

Выведите номер сочетания с повторениями.

Примеры

<code>combr2.in</code>	<code>combr2.out</code>
3 2 1 2	1

Задача Н. Номер по перестановке букв

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

Перестановкой букв строки S называются все различные строки, получающиеся из неё перестановкой букв.

Естественно, что все перестановки букв можно упорядочить лексикографически как строки. Например, для строки `abba` список упорядоченных перестановок букв будет выглядеть так: `aabb`, `abab`, `abba`, `baab`, `baba`, `bbaa`.

В этой задаче требуется найти лексикографический номер строки во множестве всех перестановок её букв (нумерация ведется с нуля).

Формат входных данных

В первой строке входного файла дана строка S , состоящая не более чем из 20 маленьких букв латинского алфавита.

Формат выходных данных

В выходной файл выведите номер строки во множестве перестановок её букв.

Примеры

<code>stringr.in</code>	<code>stringr.out</code>
<code>abab</code>	1
<code>baba</code>	4