

Задача А. Правильные скобочные последовательности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано натуральное число n . Выведите все правильные скобочные последовательности, состоящие из n открывающих круглых скобок и n закрывающих скобок в лексикографическом порядке.

Формат входных данных

Во входном файле одно число — n ($1 \leq n \leq 11$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки.

Примеры

стандартный ввод	стандартный вывод
3	((())) (()()) ()()() ()(()) ()()()

Задача В. Разбиения на слагаемые

Имя входного файла: `partition.in`
Имя выходного файла: `partition.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 4 мегабайта

Перечислите все разбиения целого положительного числа N на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

- Слагаемые в разбиениях идут в невозрастающем порядке.
- Разбиения перечисляются в лексикографическом порядке.

Формат входных данных

Во входном файле находится единственное число N ($1 \leq N \leq 45$).

Формат выходных данных

В выходной файл выведите искомые разбиения по одному на строку.

Примеры

<code>partition.in</code>	<code>partition.out</code>
4	1 1 1 1 2 1 1 2 2 3 1 4

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. Гарантируется, что x меньше количества возможных правильных скобочных последовательностей, удовлетворяющих условию задачи.

Формат выходных данных

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

Примеры

parenth2.in	parenth2.out
2 1	$() ()$
2 4	$[(())]$

Задача D. Лексикографический порядок

Имя входного файла: `lexsort.in`
Имя выходного файла: `lexsort.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 64 мегабайта

Будем считать, что одно натуральное число лексикографически меньше другого, если таковы их записи в десятичной системе счисления. Вам необходимо найти k -е по порядку число в лексикографически отсортированном множестве натуральных чисел от 1 до n включительно.

Формат входных данных

Первая строка входного файла содержит два числа n и k ($1 \leq k \leq n \leq 10^9$).

Формат выходных данных

Выведите единственное число — k -й в лексикографическом порядке элемент множества натуральных чисел от 1 до n .

Примеры

<code>lexsort.in</code>	<code>lexsort.out</code>
10 2	10

Задача Е. 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 входных данных

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 выходных данных

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 F. Конфеты Кирилла

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

У Кирилла из параллели С было k конфет, и он захотел их раздать ученикам своей параллели. Однако заметил, что конфет у него меньше чем учеников в параллели. Кирилл сел на скамейку и задумался. Просидев полчаса и доев последнюю конфету он подумал — интересно, а сколько было способов раздать все k конфет n ученикам параллели С, если конфеты нельзя делить, а каждому школьнику можно дать не более одной конфеты.

Формат входных данных

В единственной строке записаны числа n, k ($1 \leq k \leq n \leq 64$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

<code>combination.in</code>	<code>combination.out</code>
5 3	10

Задача G. Перестановка по номеру

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Выведите перестановку по её номеру.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 12$) — количество элементов в перестановке. Во второй строке записано число K ($0 \leq K < N!$) — номер перестановки в нумерации с нуля.

Формат выходных данных

В выходной файл выведите N чисел через пробел — искомую перестановку.

Примеры

<code>stdin</code>	<code>stdout</code>
3 0	1 2 3

Задача Н. Номер по перестановке

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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 входных данных

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

Формат выходных данных

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

Примеры

<code>stdin</code>	<code>stdout</code>
3 2 1 3	3

Замечание

В решении нельзя пользоваться функцией `next_permutation`.

Задача I. Перестановки

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

Дано натуральное число n . Выведите всевозможные перестановки чисел от 1 до n в **обратном** лексикографическом порядке.

Формат входных данных

Во входном файле одно число — n ($1 \leq n \leq 8$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Примеры

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

Задача J. Предыдущая перестановка

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Найдите предыдущую в лексикографическом порядке перестановку. Перестановка вида $N, N - 1, \dots, 3, 2, 1$ является предыдущей для $1, 2, 3, \dots, N - 1, N$

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 10^5$) количество элементов в перестановке. Во второй строке записана перестановка.

Формат выходных данных

В выходной файл вывести N чисел — искомую перестановку.

Примеры

<code>stdin</code>	<code>stdout</code>
3 1 2 3	3 2 1