

Задача А. 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 В. Без двух единиц подряд

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

По данному натуральному числу n выведите все двоичные последовательности длины n , не содержащие двух единиц подряд, в лексикографическом порядке.

Формат входных данных

Одно натуральное число n ($1 \leq n \leq 20$).

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>notwo.in</code>	<code>notwo.out</code>
4	0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0

Задача С. Правильные скобочные последовательности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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 D. Разбиения на слагаемые

Имя входного файла: `partition.in`
Имя выходного файла: `partition.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 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

partition.in	partition.out
4	1 1 1 1 2 1 1 2 2 3 1 4

Задача Е. Мирные ферзи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Известно, что на шахматной доске размером 8×8 можно расставить 8 ферзей не бьющих друг друга, причем сделать это можно 92 способами.

Дано натуральное n . Определите сколькими способами на доске $n \times n$ можно расставить n мирных ферзей.

Формат входных данных

Во входном файле содержится одно число n ($1 \leq n \leq 12$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
8	92

Задача F. 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

stdin	stdout
3 1 2 3	3 2 1

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

stdin	stdout
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. 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 J. ПСП по номеру

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

Помогите Андрею написать программу, генерирующую k -ю в лексикографическом порядке правильную скобочную последовательность, состоящую из $2n$ скобок.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leq n \leq 20$, $0 \leq p \leq 2 \cdot 10^9$). Скобочные последовательности нумеруются с нуля.

Формат выходных данных

В первой строке выходного файла выведите $2n$ символов без пробелов — p -ю правильную скобочную последовательность длины $2n$.

Если для данного n не существует p -я правильная скобочная последовательность, выведите в первой строке "N/A".

Примеры

<code>parens.in</code>	<code>parens.out</code>
3 0	((()))
3 1	((()()))