

Задача А. К-ичные числа

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

По данным натуральным n и k выведите все последовательности длины n , составленные из символов $0 \dots k - 1$, в лексикографическом порядке.

Формат входных данных

Во входном файле заданы два числа — n и k ($2 \leq k, n \leq 10$).

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>numbers.in</code>	<code>numbers.out</code>
2 3	0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2

Задача В. Без двух единиц подряд

Имя входного файла: fibseq.in
Имя выходного файла: fibseq.out
Ограничение по времени: 3 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 ($n \leq 26$).

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

fibseq.in	fibseq.out
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

Замечание

Реализовать рекурсивно.

Задача С. Сочетания-1

Имя входного файла: `comb1.in`
Имя выходного файла: `comb1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По данным натуральным n и k выведите все двоичные последовательности длины n , содержащие ровно k единиц в лексикографическом порядке.

Формат входных данных

Входной файл содержит два числа, n и k ($1 \leq n \leq 100, 0 \leq k \leq n$).

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. Гарантируется, что количество чисел в выходном файле не превосходит 200000

Примеры

<code>comb1.in</code>	<code>comb1.out</code>
4 2	0 0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0

Задача D. Сочетания-2

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

По данным натуральным k и n ($1 \leq k \leq n$) выведите все **убывающие** последовательности длины k состоящие из чисел $1 \dots n$ в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — k и n ($1 \leq k \leq n \leq 1000$).

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. Гарантируется, что количество чисел в выходном файле не превосходит 500 000.

Примеры

<code>comb2.in</code>	<code>comb2.out</code>
3 5	3 2 1 4 2 1 4 3 1 4 3 2 5 2 1 5 3 1 5 3 2 5 4 1 5 4 2 5 4 3

Задача E. 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 ($1 \leq n \leq 9$).

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 F. Правильные скобочные последовательности

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

Дано натуральное число n . Выведите каждую k -ю правильную скобочную последовательность, состоящую из n открывающих круглых скобок и n закрывающих скобок в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — n и k ($1 \leq n \leq 15$, $1 \leq k \leq 100$).

Формат выходных данных

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

Примеры

<code>brackets.in</code>	<code>brackets.out</code>
3 1	((())) (()()) (())() (()()) (())()
3 2	((()())) (()())

Задача G. Мирные ферзи

Имя входного файла: `queens.in`
Имя выходного файла: `queens.out`
Ограничение по времени: 2 секунды
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

<code>queens.in</code>	<code>queens.out</code>
8	92

Задача Н. Монетки

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

В Волшебной стране используются монетки достоинством A_1, A_2, \dots, A_M . Волшебный человечек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму N . Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

Формат входных данных

Сначала вводится целое число N ($1 \leq N \leq 10^9$), затем — целое число M ($1 \leq M \leq 16$) и далее M попарно различных целых чисел A_1, A_2, \dots, A_M ($1 \leq A_i \leq 10^9$).

Формат выходных данных

Выведите сначала K — количество монет, которое придется отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите K чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них.

Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число -1 (минус один).

Примеры

<code>coins.in</code>	<code>coins.out</code>
5 2 1 2	3 2 2 1
7 2 1 2	-1
5 2 3 4	0