

Задача А. Без двух единиц подряд

Имя входного файла: `fibseq.in`
Имя выходного файла: `fibseq.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 ($n \leq 20$).

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

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

Задача В. 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 ($1 \leq M \leq 10$) 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

coins.in	coins.out
5 2 1 2	3 1 2 2
7 2 1 2	-1
5 2 3 4	0

Задача С. Мирные ферзи

Имя входного файла: `queen2.in`
Имя выходного файла: `queen2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Известно, что на шахматной доске размером 8×8 можно расставить 8 ферзей так, чтобы они не били друг друга. Попробуйте расставить n ферзей на шахматной доске размером $n \times n$ так, чтобы они не били друг друга. Ваша программа должна определить количество способов это сделать.

Формат входных данных

Программа получает на вход натуральное число n , не превосходящее 12.

Формат выходных данных

Программа должна вывести единственное число: количество расстановок мирных ферзей на доске $n \times n$.

Примеры

<code>queen2.in</code>	<code>queen2.out</code>
2	0
4	2

Задача D. 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 Е. 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 F. Разбиения на слагаемые

Имя входного файла: `partition.in`
Имя выходного файла: `partition.out`
Ограничение по времени: 2 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 ($1 \leq N \leq 40$).

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

Замечание

На питоне, чтобы задача проходила по времени, надо использовать метод `write` вместо функции `print`.