

Задача А. Количество инверсий

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

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$.

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 80\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A — целых неотрицательных чисел, не превосходящих 10^6 .

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

Примеры

<code>inverse.in</code>	<code>inverse.out</code>
5 6 11 18 28 31	0
8 999994 999989 999982 999972 999969 999961 999954 999950	28

Задача В. k -ичные числа

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

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

Формат входных данных

В первой строке входного файла заданы два числа n и k ($2 \leq n, k \leq 10$).

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 С. 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

<code>fibseq.in</code>	<code>fibseq.out</code>
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

Задача D. 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 Е. 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 F. 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 G. 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

Задача Н. Генерация правильных скобочных последовательностей - 2

Имя входного файла: `brackets2.in`
Имя выходного файла: `brackets2.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайта

По данному числу n выведите все правильные скобочные последовательности из круглых и квадратных скобок длины n . Придерживайтесь следующего порядка скобок: "`()`" (см. тест из условия)

Формат входных данных

Одно целое число n ($0 \leq n \leq 16$).

Формат выходных данных

Выведите все правильные скобочные последовательности из круглых и квадратных скобок длины n в лексикографическом порядке. Каждая последовательность должна выводиться в новой строке.

Примеры

<code>brackets2.in</code>	<code>brackets2.out</code>
4	<code>((()))</code> <code>([[]])</code> <code>()()</code> <code>()[]</code> <code>[(<code>()</code>)]</code> <code>[[]]</code> <code>[]()</code> <code>[] []</code>

Задача I. 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

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

Задача J. Правильные скобочные последовательности

Имя входного файла: `brackets.in`
Имя выходного файла: `brackets.out`
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

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