

Задача А. Сортировка

Имя входного файла: `sort.in`
Имя выходного файла: `sort.out`
Ограничение по времени: 2.5 секунды
Ограничение по памяти: 128 мегабайт

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания используя быструю сортировку Хоара.

Формат входных данных

В первой строке входного файла содержится число N ($1 \leq N \leq 100\,000$) — количество элементов в массиве. Во второй строке находятся N целых чисел, по модулю не превосходящих 10^9 .

Формат выходных данных

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Примеры

<code>sort.in</code>	<code>sort.out</code>
10 1 8 2 1 4 7 3 2 3 6	1 1 2 2 3 3 4 6 7 8

Замечание

При решении задачи нельзя использовать встроенные функции `sort` и `sorted`.

Задача В. 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 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. 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. 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 F. 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 G. 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	((())) (()()) ()()() ()()() ()()()

Задача Н. Сортировка слиянием

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

Сортировка слиянием — один из самых известных алгоритмов сортировки. Основная функция этого алгоритма, сортирующая промежуток массива a с индексами из $[l, r)$, может быть реализована следующим образом:

1. Если промежуток $[l, r)$ уже отсортирован в неубывающем порядке (то есть, для каждого i , такого, что $l \leq i < r - 1$, $a[i] \leq a[i + 1]$), завершить вызов функции;
2. Присвоить $mid = \lfloor \frac{l+r}{2} \rfloor$;
3. Вызвать $mergesort(a, l, mid)$;
4. Вызвать $mergesort(a, mid, r)$;
5. Слить промежутки $[l, mid)$ и $[mid, r)$ воедино, после чего промежуток $[l, r)$ будет отсортирован в неубывающем порядке. Функция слияния не вызывает никаких других функций.

В этой задаче массив 0-индексирован, и для сортировки всего массива нужно вызвать $mergesort(a, 0, n)$.

Количество вызовов функции $mergesort$ очень важно, поэтому Иван решил вычислять его во время сортировки. К примеру, если $a = [1, 2, 3, 4]$, то будет сделан 1 вызов $mergesort$ — $mergesort(0, 4)$, который проверит, что массив отсортирован, и завершится. Если $a = [2, 1, 3]$, то количество вызовов равно 3: сначала вызывается $mergesort(0, 3)$, в котором присваивается $mid = 1$ и вызывается $mergesort(0, 1)$ и $mergesort(1, 3)$, не проводящие никаких рекурсивных вызовов, так как подотрезки $(0, 1)$ и $(1, 3)$ уже отсортированы.

Иван написал программу, которая считает число вызовов функции $mergesort$, но сейчас ему нужно протестировать её. Для этого необходимо найти массив a , такой, что a — перестановка первых n чисел (то есть размер a равен n , и каждое целое число из $[1, n]$ встречается в массиве ровно один раз), и число вызовов $mergesort$ при сортировке этого массива равно k .

Помогите Ивану найти нужный массив!

Формат входных данных

В единственной строке записаны два числа n и k ($1 \leq n \leq 100000, 1 \leq k \leq 200000$) — размер перестановки, которая нужна Ивану, и количество вызовов $mergesort$, необходимое для её сортировки.

Формат выходных данных

Если перестановка размера n , такая, что при её сортировке будет совершено ровно k вызовов $mergesort$, не существует, выведите -1 . Иначе выведите n целых чисел $a[0], a[1], \dots, a[n - 1]$ — элементы перестановки. Если решений несколько, выведите любое.

Примеры

стандартный ввод	стандартный вывод
3 3	3 1 2
4 1	1 2 3 4
5 6	-1

Замечание

Задача взята с сайта <http://codeforces.com>