

Задача А. Строка Фибоначчи

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

Строка Фибоначчи — это строка из нулей и единиц, в которой не встречается двух идущих подряд единиц.

Даны числа n и k . Нужно вывести строку Фибоначчи, которая состоит из n цифр и является k -ой в лексикографическом порядке из таких строк.

Формат входных данных

В первой строке входного файла записаны целые числа n и k через пробел ($0 \leq n \leq 44$, $0 \leq k \leq 2 \cdot 10^9$). Гарантируется, что k -ая строка из n символов существует. Строки нумеруются с нуля.

Формат выходных данных

Выведите лексикографически k -ую строку Фибоначчи длины n .

Примеры

fibstr.in	fibstr.out
3 0	000
3 1	001
3 2	010
3 3	100
3 4	101

Задача В. Номер по перестановке

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

Дана перестановка из N чисел от 1 до N . Требуется найти ее 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, разделенных пробелами.

Формат выходных данных

В выходной файл нужно вывести единственное число — номер перестановки в лексикографическом порядке.

Примеры

perm.in	perm.out
3 2 1 3	3

Задача С. Перестановка по номеру

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

Дан номер перестановки в лексикографическом порядке. Найдите перестановку с таким номером.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 12$) — количество элементов в перестановке. Во второй строке число K — номер перестановки ($1 \leq K \leq N!$).

Формат выходных данных

В выходной файл выведите N чисел через пробел — искомую перестановку.

Примеры

bynumber.in	bynumber.out
3 1	1 2 3

Задача D. Следующая перестановка

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

Найдите следующую перестановку. Лексикографически первая перестановка является следующей для обратной.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 100000$) — количество элементов в перестановке. Во второй строке записана перестановка из N чисел.

Формат выходных данных

В выходной файл нужно вывести N чисел — искомую перестановку.

Примеры

nextperm.in	nextperm.out
3 3 2 1	1 2 3
2 1 2	2 1

Задача Е. Лексикографический порядок

Имя входного файла: `lexsort.in`
Имя выходного файла: `lexsort.out`
Ограничение по времени: 0.3 секунды
Ограничение по памяти: 16 мегабайт

Будем считать, что одно натуральное число лексикографически меньше другого, если таковы их записи в десятичной системе счисления. Вам необходимо найти k -е по порядку число в лексикографически отсортированном множестве натуральных чисел от 1 до n включительно.

Формат входных данных

Первая строка входного файла содержит два числа n и k ($1 \leq k \leq n \leq 10^9$).

Формат выходных данных

Выведите единственное число — k -й в лексикографическом порядке элемент множества натуральных чисел от 1 до n .

Пример

lexsort.in	lexsort.out
10 2	10

Задача F. Следующее разбиение

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

Рассмотрим все разбиения целого положительного числа N на K целых положительных слагаемых. Запишем каждое разбиение как последовательность чисел от больших слагаемых к меньшим. Отсортируем разбиения в обратном лексикографическом порядке. Найдите следующее разбиение N на K слагаемых в заданном порядке или определите, что его не существует.

Формат входных данных

В первой строке дано целое число K ($K \leq 10^5$, $K \leq N$, $1 \leq N \leq 10^9$). Во второй строке даны K целых чисел A_i — слагаемые разбиения ($1 \leq A_i \leq N$).

Формат выходных данных

Если следующее разбиение не существует, выведите -1 . Иначе в первой строке выведите K . Во второй строке выведите K чисел — слагаемые следующего в обратном лексикографическом порядке разбиения.

Примеры

next-partition.in	next-partition.out
6 4 2 2 2 1 1	6 3 3 3 1 1 1
3 2 2 2	-1

Пояснение к примерам

В первом примере $N = 12$ и $K = 6$. Первое разбиение в обратном лексикографическом порядке — $12 = 7 + 1 + 1 + 1 + 1 + 1$, второе — $12 = 6 + 2 + 1 + 1 + 1 + 1$, 7-е — это данное разбиение $12 = 4 + 2 + 2 + 2 + 1 + 1$, следующее — $12 = 3 + 3 + 3 + 1 + 1 + 1$ и последнее 11-е разбиение — $12 = 2 + 2 + 2 + 2 + 2 + 2$.

Задача G. Следующее разбиение в RLE записи

Имя входного файла: `next-partition-rle.in`
Имя выходного файла: `next-partition-rle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим все разбиения целого положительного числа N на K целых положительных слагаемых. Запишем каждое разбиение как последовательность чисел от больших слагаемых к меньшим. Отсортируем разбиения в обратном лексикографическом порядке. Найдите следующее разбиение N на K слагаемых в заданном порядке или определите, что его не существует.

Задача усложняется тем, что разбиение задано в RLE-записи (RLE — это сокращение от **R**un **L**ength **E**ncoding). Разобьём последовательность разбиения на куски максимальной длины, состоящие из одинаковых чисел. В этой записи каждый такой кусок заменяется на пару чисел Q_i и A_i — размер i -го куска и число, из которого он состоит.

Например, одно из разбиений числа 12 на 6 слагаемых выглядит так: $12 = 4 + 2 + 2 + 2 + 1 + 1$. В RLE-записи ему соответствует последовательность пар чисел $(1, 4)$, $(3, 2)$, $(2, 1)$.

Заметим, что порядок на разбиениях задается обычной (а не RLE) формой записи.

Формат входных данных

В первой строке дано целое число L — длина RLE-записи разбиения некоторого

числа N на K слагаемых ($1 \leq L \leq 10^5$, $1 \leq N \leq 10^{18}$, $K \leq N$). В каждой из следующих строк дано два числа Q_i и A_i — количество чисел A_i в RLE-записи и само число A_i ($1 \leq Q_i, A_i \leq N$).

Формат выходных данных

Если следующее разбиение не существует, выведите -1 . Иначе в первой строке выведите M — длину RLE-записи следующего разбиения в обратном лексикографическом порядке. В каждой из следующих M строк выведите два числа R_i и B_i — количество чисел B_i в RLE-записи и само число B_i .

Примеры

next-partition-rle.in	next-partition-rle.out
3	2
1 4	3 3
3 2	3 1
2 1	
1	-1
3 2	

Пояснение к примерам

В первом примере $N = 12$ и $K = 6$. Первое разбиение в обратном лексикографическом порядке — $12 = 7 + 1 + 1 + 1 + 1 + 1$, второе — $12 = 6 + 2 + 1 + 1 + 1 + 1$, 7-е — это данное разбиение $12 = 4 + 2 + 2 + 2 + 1 + 1$, следующее — $12 = 3 + 3 + 3 + 1 + 1 + 1$ и последнее 11-е разбиение — $12 = 2 + 2 + 2 + 2 + 2 + 2$.

Задача Н. Числа фиксированной суммы

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

Сколько существует K -значных чисел с суммой цифр равной S ? Числа берутся в десятичной системе счисления. Ведущие нули допустимы.

Формат входных данных

K ($1 \leq K \leq 1000$) и S ($0 \leq S \leq 9 \cdot K$).

Формат выходных данных

Количество таких чисел, посчитанное по модулю $10^9 + 7$.

Пример

fixedsum.in	fixedsum.out
3 10	63