Задача A. 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 $0, ..., k-1$, в лексикографическом порядке.

Формат входных данных

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

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

numbers.in	numbers.out
2 3	0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2

Задача B. 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 ($n \leqslant 20$).

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 C. 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 ($1 \leqslant n \leqslant 8$).

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

permutations.in	permutations.out
3	3 2 1 3 1 2 2 3 1 2 1 3 1 3 2 1 2 3

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 ($1 \leqslant n \leqslant 100, 0 \leqslant k \leqslant n$).

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

comb1.in	comb1.out
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 E. 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 ($1 \leqslant k \leqslant n$) выведите все **убывающие** последовательности длины k состоящие из чисел $1 \ldots n$ в лексикографическом порядке.

Формат входных данных

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

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

comb2.in	comb2.out
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. 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 ($1 \leqslant n \leqslant 12$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

queens.in	queens.out
8	92

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 ($1 \leqslant N \leqslant 40$).

Формат выходных данных

В выходной файл выведите искомые разбиения по одному на строку.

Примеры

partition.in	partition.out
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.

Задача H. 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 $A = \langle a_1, a_2, \ldots, a_n \rangle$ 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 ($1 \leqslant n \leqslant 50\,000$) — 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

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