

## Порождение комбинаторных объектов при помощи рекурсивных функций.

Во всех задачах этого листка решения должны быть реализованы при помощи рекурсивной функции (или функций). Решение должно иметь вид:

```
def f(<параметры>):  
    <тело рекурсивной функции>
```

<чтение входных данных>

<инициализация массива или иных данных, если требуется>

f(<параметры>)

В основной программе может быть один или несколько вызовов (м.б. в цикле) рекурсивной функции. Все решения можно написать **без** использования глобальных переменных.

Обязательные задачи: А — G, J, P.

Задачи на "5": обязательные, а также H, I, K, L, M, N, O, Q.

### Определения

*Лексикографический порядок:*  $(x_1, \dots, x_n) < (y_1, \dots, y_n) \Leftrightarrow \exists k \geq 1 : x_k < y_k, x_i = y_i, \forall i < k$

При сравнении последовательностей разной длины действует это же определение, но сначала короткая дополняется нужным количеством минимальных элементов.

*Обратный лексикографический порядок:* меньше та последовательность, которая больше в лексикографическом порядке.

### A. Последовательности длины $N$ из чисел $1 \dots K$

По данным натуральным  $N$  и  $K$  выведите все последовательности длины  $N$  из чисел  $1 \dots K$ .

Программа должна вывести все последовательности в лексикографическом порядке. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
2	1 1
3	1 2
	1 3
	2 1
	2 2
	2 3
	3 1
	3 2
	3 3

### B. Элемент не больше своего номера

Напечатать все последовательности положительных целых чисел длины  $N$ , у которых  $i$ -ый член не превосходит  $i$ .

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все последовательности в лексикографическом порядке. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
3	1 1 1
	1 1 2
	1 1 3
	1 2 1
	1 2 2
	1 2 3

### C. Возрастающие последовательности

Перечислить все возрастающие последовательности длины  $N$  из чисел  $1 \dots K$  в лексикографическом порядке.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \geq N$ ).

Программа должна вывести все последовательности в лексикографическом порядке. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
2	1 2
3	1 3
	2 3

D. *Убывающие последовательности*

Перечислить все убывающие последовательности длины  $N$  из чисел  $1 \dots K$  в обратном лексикографическом порядке.

В первой строке вводится натуральное число  $K$ , во второй — натуральное число  $N$  ( $K \geq N$ ).

Программа должна вывести все последовательности в обратном лексикографическом порядке.

Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
4	4 3 2
3	4 3 1
	4 2 1
	3 2 1

E. *Двоичные последовательности без двух единиц подряд*

По данному натуральному  $N$  выведите все двоичные последовательности длины  $N$ , не содержащие двух единиц подряд.

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные последовательности. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
3	0 0 0
	0 0 1
	0 1 0
	1 0 0
	1 0 1

F. *Двоичные последовательности длины  $N$  содержащие не более  $K$  единиц*

По данным натуральным  $N$  и  $K$  ( $0 \leq K \leq N, N \geq 1$ ) выведите все двоичные последовательности длины  $N$ , содержащие не более  $K$  единиц.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ).

Программа должна вывести все указанные последовательности в лексикографическом порядке.

Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
3	0 0 0
1	0 0 1
	0 1 0
	1 0 0

G. *Двоичные последовательности длины  $N$  содержащие  $K$  единиц*

По данным натуральным  $N$  и  $K$  ( $0 \leq K \leq N, N \geq 1$ ) выведите все двоичные последовательности длины  $N$ , содержащие ровно  $K$  единиц.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ).

Программа должна вывести все указанные последовательности в лексикографическом порядке.

Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
3	0 1 1
2	1 0 1
	1 1 0

H. *Двоичные последовательности длины  $N$  содержащие не более  $K$  единиц без двух единиц подряд*

По данным натуральным  $N$  и  $K$  ( $0 \leq K \leq N, N \geq 1$ ) выведите все двоичные последовательности длины  $N$ , содержащие не более  $K$  единиц без двух единиц подряд.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ).

Программа должна вывести все указанные последовательности в лексикографическом порядке.

Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
3	0 0 0
2	0 0 1
	0 1 0
	1 0 0
	1 0 1

I. *Двоичные последовательности длины  $N$  содержащие ровно  $K$  единиц без двух единиц подряд*  
 По данным натуральным  $N$  и  $K$  ( $0 \leq K \leq N, N \geq 1$ ) выведите все двоичные последовательности длины  $N$ , содержащие ровно  $K$  единиц без двух единиц подряд.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ).  
 Программа должна вывести все указанные последовательности в лексикографическом порядке.  
 Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
4	0 1 0 1
2	1 0 0 1
	1 0 1 0

J. *Разбиение на невозрастающие слагаемые, лексикографический порядок*

Перечислить в лексикографическом порядке все невозрастающие разбиения целого положительного числа  $N$  на целые положительные слагаемые (разбиения, отличающиеся лишь порядком слагаемых, считаются одинаковыми).

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные последовательности в лексикографическом порядке.  
 Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
4	1 1 1 1
	2 1 1
	2 2
	3 1
	4

K. *Разбиение на невозрастающие слагаемые, обратный порядок*

Перечислить в обратном лексикографическом порядке все невозрастающие разбиения целого положительного числа  $N$  на целые положительные слагаемые (разбиения, отличающиеся лишь порядком слагаемых, считаются одинаковыми).

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные последовательности. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
5	5
	4 1
	3 2
	3 1 1
	2 2 1
	2 1 1 1
	1 1 1 1 1

L. *Разбиение на неубывающие слагаемые, лексикографический порядок*

Перечислить в лексикографическом порядке все неубывающие разбиения целого положительного числа  $N$  на целые положительные слагаемые (разбиения, отличающиеся лишь порядком слагаемых, считаются одинаковыми).

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные последовательности. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
4	1 1 1 1
	1 1 2
	1 3
	2 2
	4

M. *Разбиение на неубывающие слагаемые, обратный порядок*

Перечислить в обратном лексикографическом порядке все неубывающие разбиения целого положительного числа  $N$  на целые положительные слагаемые (разбиения, отличающиеся лишь порядком слагаемых, считаются одинаковыми).

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные последовательности. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
5	5
	2 3
	1 4
	1 2 2
	1 1 3
	1 1 1 2
	1 1 1 1 1

N. *Разбиение на  $K$  невозрастающих слагаемых*

Даны натуральные числа  $N$  и  $K$  ( $1 \leq K \leq N$ ). Выведите всевозможные разбиения числа  $N$  на  $K$  слагаемых, упорядоченных в порядке невозрастания. Сами разбиения необходимо выводить в лексикографическом порядке.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ). Программа должна вывести все указанные последовательности в лексикографическом порядке. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
8	3 3 2
3	4 2 2
	4 3 1
	5 2 1
	6 1 1

O. *Разбиение на  $K$  неубывающих слагаемых*

Даны натуральные числа  $N$  и  $K$  ( $1 \leq K \leq N$ ). Выведите всевозможные разбиения числа  $N$  на  $K$  слагаемых, упорядоченных в порядке неубывания. Сами разбиения необходимо выводить в лексикографическом порядке.

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ). Программа должна вывести все указанные последовательности в лексикографическом порядке. Элементы последовательностей должны быть разделены одним пробелом.

Input	Output
8	1 1 6
3	1 2 5
	1 3 4
	2 2 4
	2 3 3

P. *Правильные скобочные последовательности*

Дано натуральное число  $N$ . Выведите все правильные скобочные последовательности, состоящие из  $N$  открывающихся круглых скобок и  $N$  закрывающихся скобок в лексикографическом порядке.

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все указанные скобочные последовательности в лексикографическом порядке (открывающая скобка меньше, чем закрывающая), не разделяя скобки пробелами.

Input	Output
3	((()))
	(()())
	((())())
	()(())
	()()()

Q. *Правильные скобочные последовательности длины  $2N$  вложенности не более  $K$*

Даны натуральные числа  $N$  и  $K$ . Выведите в лексикографическом порядке все правильные скобочные последовательности, состоящие из  $N$  открывающихся круглых скобок и  $N$  закрывающихся скобок так, что максимальная вложенность не превосходит  $K$ .

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$  ( $K \leq N$ ). Программа должна вывести все указанные скобочные последовательности в лексикографическом порядке (открывающая скобка меньше, чем закрывающая), не разделяя скобки пробелами.

Input	Output
4	((()()))
2	((()())())
	((())()())
	((())()())
	()(())()
	()(())()
	()(())()
	()(())()
	()(())()

R. *Все перестановки*

Напечатать все перестановки чисел  $1 \dots N$  (то есть последовательности длины  $n$ , в которые каждое из этих чисел входит по одному разу).

На вход программе подаётся единственная строка, содержащая натуральное число  $N$ .

Программа должна вывести все перестановки чисел  $1 \dots N$ . Элементы перестановок должны быть разделены одним пробелом.

Input	Output
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

S. *Правильные скобочные последовательности — 2*

Дано натуральное число  $N$ . Выведите в лексикографическом порядке все правильные скобочные последовательности, состоящие из  $N$  открывающихся скобок и  $N$  закрывающихся скобок. Используются круглые и квадратные скобки, их порядок следующий:

$( < [ < [ <$

Input	Output
2	(( )) ( ) ( [ ] ) ([ ] ) [ ( ] ) [ [ ] ] [ ] ( ) [ ] [ ]

T\* *Правильные скобочные последовательности - 3*

Дано натуральное число  $N$ . Выведите  $K$  первых в лексикографическом порядке правильных скобочных последовательности, состоящих из  $N$  открывающихся скобок и  $N$  закрывающихся скобок в лексикографическом порядке. Используются круглые, квадратные и фигурные скобки. Лексикографический порядок для скобок задаётся в тестовых данных.

Программа получает на вход число  $N$  — количество открывающихся скобок в исходной последовательности и число  $K$  — количество правильных скобочных последовательностей, которые необходимо вывести ( $NK \leq 10^6$ ).

Затем идет строка из 6 символов, являющаяся некоторой перестановкой строки  $() [] \{\}$ , задающая лексикографический порядок.

Гарантируется, что существует  $k$  правильных последовательностей указанной длины.

Input	Output
3 10 { } ( ) (	[[[]]] [[{}]] [[[]]] [[{}]] [[[]]] [[[]]] [[[]]] [[[]]] [[[]]] [[[]]] [[[]]] [[{}]]

U. *Расстановки ферзей*

Известно, что на шахматной доске размером  $8 \times 8$  можно расставить 8 ферзей не бьющих друг друга, причем сделать это можно 92 способами.

Дано натуральное  $N \leq 10$ . Определите сколькими способами на доске  $N \times N$  можно расставить  $N$  мирных ферзей.

Input	Output
8	92

V. *Расстановки ферзей — 2*

Решите предыдущую задачу, если расстановки ферзей, которые можно получить друг из друга поворотами и отражениями доски считать за одно.

Input	Output
8	12