

Листок 13: 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.

А. 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

В. 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

С. 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

