

## Листок 12: Порождение комбинаторных объектов. Итеративные алгоритмы.

Во всех задачах этого листка решения должны быть нерекурсивными. Время работы (если не указано иное) должно быть пропорционально количеству выводимых на экран чисел.

В задачах 1 – 10 не допускается использование вспомогательных массивов (единственный массив используется для хранения результата).

### 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. Все последовательности длины $N$ из чисел $1 \dots K$

Напечатать все последовательности длины  $N$  из чисел  $1 \dots K$ .

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$ .

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. Все возрастающие

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

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

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

Input	Output
3	1 2
2	1 3
	2 3

### D. Все убывающие

Перечислить все убывающие последовательности длины  $K$  из чисел  $1 \dots N$ .

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

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

Input	Output
3	3 2
2	3 1
	2 1

Е. *Все подмножества*

Напечатать все подмножества множества  $\{1 \dots N\}$ .

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

Программа должна вывести  $2^N - 1$  строк, в каждой из которых перечислены элементы одного из подмножеств (пустое подмножество печатать не надо). Подмножества разной мощности должны быть упорядочены по размеру, подмножества равной мощности должны быть перечислены в лексикографическом порядке.

Элементы подмножеств должны быть разделены одним пробелом.

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

Ф. *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

Г. *Следующая анаграмма*

Для данного слова (последовательности строчных латинских букв) выведите следующее за ним (в лексикографическом порядке) слово, которое может быть получено из данного перестановкой букв (анаграмму). Если данное слово уже является последним среди всех своих анаграмм, то необходимо вывести первую возможную (в лексикографическом порядке) анаграмму.

*Указание 1:* для хранения последовательности символов используйте массив символов (массив можно изменять в отличие от строки).

*Указание 2:* поскольку количество слов во входных данных заранее неизвестно, то читать их требуется из файла.

Задана последовательность слов, по одному слову в строке. Длина одного слова не превышает 50 символов.

Необходимо вывести результат для каждого полученного на вход слова.

Input	Output
aab	aba
aba	baa
baa	aab
aaa	aaa

Н\* *Все вложения*

Перечислить все вложения (функции, переводящие разные элементы в разные) множества  $\{1 \dots K\}$  в  $\{1 \dots N\}$ . Порождение очередного элемента должно требовать не более  $C * K$  действий.

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

Программа должна вывести все возможные последовательности длины  $K$ , состоящие из элементов множества  $N$ . Порядок следования последовательностей отличается от лексикографического (см. пример). Элементы указанных последовательностей должны быть разделены одним пробелом.

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

I. *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 в лексикографическом порядке.

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

J. *Двоичные последовательности, 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

K. *Двоичные последовательности, ровно  $K$  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

L. *Двоичные последовательности, 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

M. *Двоичные последовательности, 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

N. *Разбиения натурального числа на слагаемые: невозрастание, 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. Элементы последовательностей должны быть разделены одним пробелом.

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

O. *Разбиения натурального числа на слагаемые: невозрастание, обратный лексикографический 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

P. *Разбиения натурального числа на слагаемые: неубывание, 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

Q. *Разбиения натурального числа на слагаемые: неубывание, обратный лексикографический 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

R. *Правильные скобочные последовательности длины  $2N$*

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	((())) (()()) (())() ()(() ()()()

S. *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	((()()) (()())() (())(()) (())()() ()()() ()()() ()()() ()()() ()()() ()()()

Задачи T–W сдаются не в тестирующую систему, а преподавателю.

T. *Код Грея*

Перечислить все последовательности длины  $N$  из чисел  $1 \dots K$  в таком порядке, чтобы каждая следующая отличалась от предыдущей в единственной позиции, причем не более, чем на 1. Разрешается использовать вспомогательный массив длины  $N$ .

В первой строке вводится натуральное число  $N$ , во второй — натуральное число  $K$ .

Программа должна вывести все указанные последовательности по одной на строке. Элементы последовательностей должны быть разделены пробелами.

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

U\* *Перестановки: соседние отличаются одной транспозицией*

Напечатать все перестановки чисел  $1 \dots N$  так, чтобы каждая следующая получалась из предыдущей перестановкой (транспозицией) двух соседних чисел.

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

Программа должна вывести все перестановки, удовлетворяющие условию.

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

V. *Все расстановки скобок в неассоциативном произведении*

Перечислить все расстановки скобок в произведении  $N$  сомножителей. Порядок сомножителей не меняется, скобки должны полностью определять порядок действий.

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

Программа должна вывести все способы расстановки скобок в приведённом ниже формате.

Input	Output
4	$x*(x*(x*x))$
	$x*((x*x)*x)$
	$(x*(x*x))*x$
	$(x*x)*(x*x)$
	$((x*x)*x)*x$

W.  *$N$  непересекающихся хорд*

На окружности задано  $2N$  точек, пронумерованных от 1 до  $2N$ . Перечислить все способы провести  $N$  непересекающихся хорд с вершинами в этих точках.

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

Программа должна вывести в лексикографическом порядке все наборы хорд, удовлетворяющие условию. Каждая хорда в наборе должна быть записана в виде пары вершин, упорядоченных по возрастанию, разделённых знаком '-'.  
Пример: 1-2 3-4 5-6

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