

## Задача А. 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 натуральным  $n$  и  $k$  ( $2 \leq k \leq 10$ ) выведите все последовательности длины  $n$ , составленные из символов  $0 \dots k-1$ , в лексикографическом порядке.

### Формат входных данных

Во входном файле заданы два числа —  $n$  и  $k$ .

### 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.

### Пример

<code>numbers.in</code>	<code>numbers.out</code>
2 3	0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2

## Задача В. 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  $n$  ( $n \leq 20$ ).

### 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.

### Пример

<code>fibseq.in</code>	<code>fibseq.out</code>
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

## Задача С. 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  $k$  ( $n \geq 1, 0 \leq k \leq n$ ).

### 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.

### Пример

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 D. 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  $n$  и  $k$  ( $n \leq k$ ) выведите все возрастающие последовательности длины  $n$ , состоящие из чисел  $1 \dots k$ .

### Формат входных данных

Во входном файле два числа —  $n$  и  $k$ .

### 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.

### Пример

comb2.in	comb2.out
3 5	1 2 3 1 2 4 1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 2 3 5 2 4 5 3 4 5

### Задача Е. 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 лексикографическом порядке.

#### Формат входных данных

Во входном файле одно число —  $n$ .

#### 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.

#### Пример

<code>permutations.in</code>	<code>permutations.out</code>
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

### Задача F. Правильные скобочные последовательности

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

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

#### Формат входных данных

Во входном файле одно число —  $n$ .

#### 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.

#### Пример

<code>brackets.in</code>	<code>brackets.out</code>
3	((())) (()()) ()(()) ()()() ()()()

## Задача G. 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  $8 \times 8$  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.

### Формат входных данных

Во входном файле содержится одно число  $n$  ( $1 \leq n \leq 10$ ).

### Формат выходных данных

Выведите единственное число  $n$  - ответ на задачу.

### Пример

<code>queens.in</code>	<code>queens.out</code>
8	92