

## Задача А. 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.

### Примеры

fibseq.in	fibseq.out
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

## Задача В. 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  $k$  ( $1 \leq n \leq 100, 0 \leq k \leq n$ ).

### 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. Гарантируется, что количество чисел в выходном файле не превосходит 200000

### Примеры

<code>comb1.in</code>	<code>comb1.out</code>
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 С. 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  $n$  ( $1 \leq k \leq n$ ) выведите все **убывающие** последовательности длины  $k$  состоящие из чисел  $1 \dots n$  в лексикографическом порядке.

### Формат входных данных

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

### 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

<code>comb2.in</code>	<code>comb2.out</code>
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 D. 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  $n$ . Определите сколькими способами на доске  $n \times n$  можно расставить  $n$  мирных ферзей.

### Формат входных данных

Во входном файле содержится одно число  $n$  ( $1 \leq n \leq 12$ ).

### Формат выходных данных

Выведите единственное число — ответ на задачу.

### Примеры

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

## Задача E. Монетки

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

В Волшебной стране используются монетки достоинством  $A_1, A_2, \dots, A_M$ . Волшебный человечек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму  $N$ . Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

### Формат входных данных

Сначала вводится целое число  $N$  ( $1 \leq N \leq 10^9$ ), затем — целое число  $M$  ( $1 \leq M \leq 10$ ) и далее  $M$  попарно различных целых чисел  $A_1, A_2, \dots, A_M$  ( $1 \leq A_i \leq 10^9$ ).

### Формат выходных данных

Выведите сначала  $K$  — количество монет, которое придется отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите  $K$  чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них.

Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число  $-1$  (минус один).

### Примеры

<code>coins.in</code>	<code>coins.out</code>
5 2 1 2	3 1 2 2
7 2 1 2	-1
5 2 3 4	0

## 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  $n$  открывающих круглых скобок и  $n$  закрывающих скобок в лексикографическом порядке.

### Формат входных данных

Во входном файле одно число —  $n$  ( $1 \leq n \leq 11$ ).

### Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки.

### Примеры

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

## Задача G. Генерация правильных скобочных последовательностей - 2

Имя входного файла: `brackets2.in`  
Имя выходного файла: `brackets2.out`  
Ограничение по времени: 6 секунд  
Ограничение по памяти: 64 мегабайта

По данному числу  $n$  выведите все правильные скобочные последовательности из круглых и квадратных скобок длины  $n$  в лексикографическом порядке.

### Формат входных данных

Одно целое число  $n$  ( $0 \leq n \leq 16$ ).

### Формат выходных данных

Выведите все правильные скобочные последовательности из круглых и квадратных скобок длины  $n$  в лексикографическом порядке. Каждая последовательность должна выводиться в новой строке.

### Примеры

<code>brackets2.in</code>	<code>brackets2.out</code>
4	(( ( ) ) ) [ [ ] ] ] ]