

## Задача А. Реверс

Имя входного файла: `reverse.in`  
Имя выходного файла: `reverse.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Переверните массив чисел. Без циклов. Без списков. Без контейнеров.

### Формат входных данных

В первой строке — число  $N$ . Далее в  $N$  строках указаны  $N$  чисел ( $1 \leq N \leq 10000$ ), по одному числу на строке. Все числа по модулю не превышают  $10^5$ .

### Формат выходных данных

Выведите  $N$  чисел — перевернутый массив.

### Примеры

<code>reverse.in</code>	<code>reverse.out</code>
5 18086 53523 871 93415 50530	50530 93415 871 53523 18086

## Задача В. Степень двойки

Имя входного файла: `two.in`  
Имя выходного файла: `two.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

У инопланетян с планеты Уран по два пальца на руке, поэтому им очень нравятся степени двойки. Начинаящий дипломат Василий хочет привезти им  $N$  подарков. Чтобы всё прошло идеально, Василий хочет убедиться, что количество подарков соответствует их предпочтениям. Помогите ему в этом!

### Формат входных данных

В первой строке дано число  $N$  ( $1 \leq N \leq 10^9$ ).

### Формат выходных данных

Нужно в единственной строке выходного файла вывести строку "YES", если число является степенью двойки и "NO" в противном случае.

### Примеры

<code>two.in</code>	<code>two.out</code>
1	YES
16	YES
24	NO

### Замечание

При решении данной задачи **нельзя** использовать циклы (`for`, `while`), а также операцию возведения в степень.

## Задача С. Возведение в степень

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

### Формат входных данных

Во входном файле даны три натуральных числа  $A, B, M$  ( $1 \leq A, B \leq 10^9, 2 \leq M \leq 10^9$ ), записанные на одной строке через пробел.

### Формат выходных данных

В выходной файл выведите одно число, равное  $A^B \bmod M$  (`mod` означает взятие остатка при делении).

### Примеры

<code>stdin</code>	<code>stdout</code>
2 3 100	8

### Замечание

При решении данной задачи нельзя использовать встроенную операцию возведения в степень.

## Задача D. $k$ -ичные числа

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

По данным  $n$  и  $k$  выведите все последовательности длины  $n$ , составленные из цифр  $0, \dots, k - 1$ , в лексикографическом порядке.

### Формат входных данных

В первой строке входного файла заданы два числа  $n$  и  $k$  ( $2 \leq n, k \leq 10$ ).

### 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

стандартный ввод	стандартный вывод
2 3	0 0 0 1 0 2 1 0 1 1 1 2 2 0 2 1 2 2

## Задача Е. Без двух единиц подряд

Имя входного файла: `notwo.in`  
Имя выходного файла: `notwo.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  $n$  ( $1 \leq n \leq 20$ ).

### 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

<code>notwo.in</code>	<code>notwo.out</code>
4	0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 1 0

## Задача F. Ровно k единиц

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 G. Монетки

Имя входного файла: `coins.in`  
Имя выходного файла: `coins.out`  
Ограничение по времени: 2.2 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 Н. Правильные скобочные последовательности

Имя входного файла: `brackets.in`  
Имя выходного файла: `brackets.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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