

## Задача А. Конфеты Кирилла

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

У Кирилла из параллели С было  $k$  конфет, и он захотел их раздать ученикам своей параллели. Однако заметил, что конфет у него меньше чем учеников в параллели. Кирилл сел на скамейку и задумался. Просидев полчаса и доев последнюю конфету он подумал — интересно, а сколько было способов раздать все  $k$  конфет  $n$  ученикам параллели С, если конфеты нельзя делить, а каждому школьнику можно дать не более одной конфеты.

### Формат входных данных

В единственной строке записаны числа  $n, k(1 \leq k \leq n \leq 64)$ .

### Формат выходных данных

Выведите единственное число — ответ на задачу.

### Примеры

<code>combination.in</code>	<code>combination.out</code>
5 3	10

## Задача В. Необходимые разбиения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Демид, Никита, Тимофей и Олег решили отправиться вверх по Волге на лодке, от Костромы до Ярославля. Прежде чем отправиться в путь, они собирают чемоданы и берут с собой самое необходимое.

Демид сказал, что в таком трудном и опасном путешествии нельзя обойтись без упорядоченных разбиений числа  $n$  на слагаемые. Напомним, что *упорядоченным разбиением числа  $n$  на слагаемые* называется любая последовательность натуральных чисел, в сумме дающая  $n$ . Тимофей заметил, что брать с собой разбиения, в которых есть числа, большие  $r$  — пустая трата и без того небольшого места в чемоданах. Никита же заявил, что они вполне могут обойтись и без разбиений, в которых есть числа, меньшие  $l$ . Нельзя сказать, чтобы он путешествовал хоть раз без них, но знал людей, которые путешествовали. Олег же лишь одобрительно кивнул.

Помогите друзьям посчитать количество таких разбиений (то есть разбиений, в которых каждое слагаемое принадлежит отрезку  $[l; r]$ ). Так как искомое число может быть очень большим, выведите остаток от деления этого числа на  $10^9 + 7$ .

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n, l$  и  $r$  ( $1 \leq l \leq r \leq n \leq 500\,000$ ).

### Формат выходных данных

В единственную строку выведите ответ на задачу — остаток от деления искомого количества разбиений на число  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
5 2 5	3
10 3 8	8

### Замечание

В первом примере подходят разбиения:

$$5 = 5$$

$$5 = 2 + 3$$

$$5 = 3 + 2$$

Во втором примере:

$$10 = 7 + 3$$

$$10 = 3 + 7$$

$$10 = 6 + 4$$

$$10 = 4 + 6$$

$$10 = 5 + 5$$

$$10 = 3 + 3 + 4$$

$$10 = 3 + 4 + 3$$

$$10 = 4 + 3 + 3$$

## Задача С. Количество правильных скобочных последовательностей

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

Найдите число правильных скобочных последовательностей из  $n$  открывающих и  $n$  закрывающих скобок одного вида.

### Формат входных данных

Программа получает на вход одно число  $n$  ( $1 \leq n \leq 10^3$ ) — количество открывающих и закрывающих скобок в последовательности.

### Формат выходных данных

Программа должна вывести остаток от деления числа правильных скобочных последовательностей из  $n$  пар скобок на  $10^9 + 7$ .

### Примеры

<code>catalan.in</code>	<code>catalan.out</code>
3	5
4	14

## Задача D. Горилл улучшает RSA

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Однажды огромный горилл решил модернизировать известный алгоритм шифрования RSA. Он подумал, что необязательно использовать в качестве  $n$  произведение двух различных простых чисел и предложил использовать  $n$ , которое представляет собой произведение степеней  $k$  двух различных простых чисел:  $n = p^k q^k$ .

Назовём *красивым* разложение числа  $n$  на множители, если множителей хотя бы два, и каждый из них строго больше 1. Оказалось, что в случае  $n = p^k q^k$  у числа  $n$  может быть несколько различных *красивых* разложений на множители. Например,  $100 = 2^2 5^2$  имеет восемь *красивых* разложений:  $100 = 2 \cdot 50$ ,  $100 = 2 \cdot 2 \cdot 25$ ,  $100 = 2 \cdot 2 \cdot 5 \cdot 5$ ,  $100 = 2 \cdot 5 \cdot 10$ ,  $100 = 4 \cdot 25$ ,  $100 = 4 \cdot 5 \cdot 5$ ,  $100 = 5 \cdot 20$  и  $100 = 10 \cdot 10$ .

Теперь гориллу мучает вопрос: пусть  $n = p^k q^k$ , сколько существует различных *красивых* разложений  $n$  на множители?

### Формат входных данных

На вход подается одно целое число  $n$  ( $6 \leq n \leq 10^{18}$ ,  $n = p^k q^k$  для двух различных простых  $p$  и  $q$  для целого  $k > 0$ ).

### Формат выходных данных

Выведите одно число — количество *красивых* разложений  $n$  на множители.

### Примеры

стандартный ввод	стандартный вывод
6	1
100	8

## Задача Е. Симпатичные узоры 2

Имя входного файла: nice2.in  
Имя выходного файла: nice2.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

### Формат входных данных

На первой строке входного файла находятся два натуральных числа  $n$  и  $m$ .  $1 \leq n \cdot m \leq 300$ .

### Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$  по модулю  $2^{30} + 1$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

### Примеры

nice2.in	nice2.out
2 2	14
3 3	322

## Задача F. Доминошки

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

Вам дано клетчатое поле размера  $n \times m$  клеток. Некоторые клетки на этом поле отмечены как занятые. На нем необходимо разместить несколько (возможно, 0) доминошек размера  $1 \times 2$ , соблюдая следующие ограничения:

- каждая клетка может быть покрыта не более чем одной доминошкой
- никакие две соседних по стороне клетки не могут быть заняты разными доминошками

Сколькими способами можно это сделать?

### Формат входных данных

В первой строке входного файла содержится два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 6, 1 \leq m \leq 100$ ). В следующих  $n$  строках содержится описание поля. Символ  $X$  означает занятую клетку, символ  $.$  означает свободную клетку.

### Формат выходных данных

Выведите одно целое число — количество способов разместить доминошки, взятое по модулю  $10^9 + 7$ .

### Примеры

dominoes.in	dominoes.out
2 2 .. ..	5
3 3 ... .X. ...	21
5 5 ..... ..... ..X.. ..... .....	5182

## Задача G. Произведение матриц

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

Произведением матриц  $A$  и  $B$  размера  $p \times q$  и  $q \times r$ , соответственно, называется матрица  $C$  размера  $p \times r$ , элементы которой вычисляются по формуле:

$$C_{i,j} = \sum_{k=1}^q A_{i,k} \cdot B_{k,j}$$

По данным матрицам  $A$  и  $B$  найдите их произведение.

### Формат входных данных

В первой строке входного файла заданы через пробел три целых числа  $p$ ,  $q$  и  $r$  ( $1 \leq p, q, r \leq 100$ ). В следующих  $p$  строках записана матрица  $A$ ; каждая из этих строк содержит  $q$  целых чисел, разделённых пробелами. Наконец, в последних  $q$  строках записана матрица  $B$ ; каждая из этих строк содержит  $r$  целых чисел, разделённых пробелами. Элементы матриц не превосходят 100 по абсолютной величине.

### Формат выходных данных

В выходной файл выведите матрицу  $C$ :  $p$  строк, в каждой из которых  $r$  чисел через пробел.

### Примеры

<code>mmul.in</code>	<code>mmul.out</code>
2 2 2 1 0 0 1 1 0 0 1	1 0 0 1
1 3 1 1 2 3 -1 -2 -3	-14
3 2 4 0 1 1 0 0 1 2 1 0 0 1 1 2 1	1 1 2 1 2 1 0 0 1 1 2 1

## Задача Н. Степень матрицы

Имя входного файла: `mpow.in`  
Имя выходного файла: `mpow.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Задана квадратная матрица  $n \times n$ . Нужно возвести ее в степень  $m$

### Формат входных данных

В первой строке задано три целых числа  $n$ ,  $m$  и  $p$  ( $1 \leq n \leq 100$ ,  $0 \leq m \leq 10^{18}$ ,  $2 \leq p \leq 10^9$ ).  
Далее задана матрица:  $n$  строк по  $n$  целых чисел. Все числа в матрице неотрицательны и меньше  $p$ .

### Формат выходных данных

Выведите матрицу:  $n$  строк по  $n$  чисел, каждое число — остаток от деления элемента на  $p$

### Примеры

<code>mpow.in</code>	<code>mpow.out</code>
3 5 239 1 0 1 1 2 0 3 2 1	120 92 56 102 84 46 21 204 120
5 10 27 1 2 3 4 5 5 4 3 2 1 11 12 13 14 15 15 14 13 12 11 1 11 1 11 1	2 5 7 10 12 16 19 26 2 9 5 9 26 3 20 19 23 18 22 17 12 22 9 19 6



## Задача I. Обобщенные числа фибоначчи

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

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано  $n$ , найдите значение  $f_n$ , взятое по модулю  $10^9 + 7$ .

### Формат входных данных

Неотрицательные целые числа:  $a, b, c, d, e, n$ . ( $0 \leq a, b, c, d, e \leq 10^9$ ;  $1 \leq n \leq 10^{18}$ )

### Формат выходных данных

Выведите  $f_n$ , взятое по модулю  $10^9 + 7$ .

### Примеры

fibonacci.in	fibonacci.out
1 1 0 0 0 8	21
1 2 3 4 5 6	775

## Задача J. Гладкие числа

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

Назовем число гладким, если его соседние цифры по модулю различаются не более чем на 1. Для данного  $N$  определите количество гладких чисел, имеющих длину  $N$ .

### Формат входных данных

На вход программы поступает количество цифр  $N \leq 10^9$ .

### Формат выходных данных

Выведите ответ по модулю  $10^9 + 7$ .

### Примеры

<code>smooth.in</code>	<code>smooth.out</code>
1	9