

## Задача А. 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 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

## Задача В. Симпатичные узоры наносят ответный удар

Имя входного файла: nice3.in  
Имя выходного файла: nice3.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  $m$ .  $1 \leq n \leq 10^{100}$ ,  $1 \leq m \leq 5$ ,  $1 \leq p \leq 10000$ .

### Формат выходных данных

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

### Примеры

nice3.in	nice3.out
2 2 20	14
3 3 7	0

## Задача С. Транзисторы над Пекином возвращаются

Имя входного файла: transistor2.in  
Имя выходного файла: transistor2.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме  $N$  этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа  $f$  транзистор обязательно разобьется при падении с этажа  $f + 1$ . Дополнительно поставлено условие, что разрешено проведение не более чем  $K$  испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

### Формат входных данных

В первой и единственной строке входного файла содержатся два целых числа  $N$  и  $K$  ( $1 \leq N \leq 10^{18}$ ,  $0 \leq K \leq 10^{18}$ ).

### Формат выходных данных

Выведите единственное число — ответ на поставленную задачу. Если для данных  $N$  и  $K$  возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите  $-1$ .

### Примеры

transistor2.in	transistor2.out
4 2	-1
4 3	2

## Задача D. Рюкзак-2

Имя входного файла: `knapsack.in`  
Имя выходного файла: `knapsack.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 8 мегабайт

Найдите максимальную цену слитков золота, которые можно унести в рюкзаке вместительностью  $S$ , если есть  $N$  золотых слитков с заданными весами и ценами.

### Формат входных данных

В первой строке входных данных записаны два числа —  $S$  и  $N$  ( $1 \leq S \leq 10\,000$ ,  $1 \leq N \leq 300$ ).

В двух следующих строках записано по  $N$  неотрицательных целых чисел в каждой — веса и стоимости слитков соответственно. Каждое из этих чисел не превосходит 100 000.

### Формат выходных данных

Определите набор предметов максимальной стоимости, помещающийся в данный рюкзак. В первой строке выведите стоимость предметов в набранном наборе, во второй — количество предметов в наборе. В следующей строке выведите через пробел номера этих предметов.

### Примеры

<code>knapsack.in</code>	<code>knapsack.out</code>
10 3	123
1 4 8	2
72 7 51	1 3

## Задача E. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`  
Имя выходного файла: `lcs.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 7 мегабайта

Найдите наибольшую общую подпоследовательность двух строк.

### Формат входных данных

В первой и второй строке входных данных записаны строки из строчных латинских букв длины не более  $5 \cdot 10^3$ . Строки не пусты.

### Формат выходных данных

Выведите единственную строку – наибольшую общую подпоследовательность двух данных строк.

### Примеры

<code>lcs.in</code>	<code>lcs.out</code>
abc bacd	bc
abacabadabacaba dbdccbdb	dbcb

## Задача F. Банкомат

Имя входного файла: `atm.in`  
Имя выходного файла: `atm.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 6 мегабайт

В некотором государстве в обращении находятся банкноты определенных номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.

### Формат входных данных

Первая строка входных данных содержит натуральное число  $N$  не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит  $N$  различных натуральных чисел  $x_1, x_2, \dots, x_N$ , не превосходящих  $10^6$  — номиналы банкнот. Третья строка содержит натуральное число  $S$ , не превосходящее  $10^6$  — сумму, которую необходимо выдать.

### Формат выходных данных

В первую строку выходного файла выведите минимальное число слагаемых (или -1, если такого представления не существует). Во вторую строку выведите это представление в любом порядке.

### Примеры

<code>atm.in</code>	<code>atm.out</code>
5 1 3 7 12 32 40	3 1 7 32