Задача A. RSA. Взлом RSA

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

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа p и q, вычислить $n = pq$ и сгенерировать два числа e и d такие, что $ed \equiv 1 \pmod{(p-1)(q-1)}$ (заметим, что $(p-1)(q-1) = \varphi(n)$). Числа n и e составляют открытый ключ и являются общеизвестными. Число d является секретным ключом, также необходимо хранить в тайне и разложение числа n на простые множители, так как это позволяет вычислить секретный ключ d.

Сообщениями в системе RSA являются числа из $\mathbb{Z}_n$. Пусть M — исходное сообщение. Для его шифрования вычисляется значение $C = M^e \bmod n$ (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение C передается по каналу связи. Для его расшифровки необходимо вычислить значение $M = C^d \bmod n$, а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение C и знаете только открытый ключ: числа n и e. “Взломайте” RSA — расшифруйте сообщение на основе только этих данных.

Формат входных данных

Программа получает на вход три натуральных числа: n, e, C, $n \leqslant 10^9$, $e \leqslant 10^9$, $C < n$. Числа n и e являются частью какой-то реальной схемы RSA, т.е. n является произведением двух простых и e взаимно просто с $\varphi(n)$. Число C является результатом шифрования некоторого сообщения M.

Формат выходных данных

Выведите одно число M $(0 \leqslant M < n)$, которое было зашифровано такой криптосхемой.

Примеры

rsa.in	rsa.out
143 113 41	123
9173503 3 4051753	111111

Задача B. Xor-шифрование

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

Каждый тест этой задачи — английский текст, зашифрованный с помощью xor-шифрования.

Перед шифрованием выбран случайный байт b, после чего каждый байт исходного текста с кодом c был заменён на символ с кодом $c \wedge b$. Гарантируется, что в исходном тексте встречались только печатные символы с кодами от 32 до 128 и переводы строк, а размер текста был не менее 1 Кб.

Восстановите исходный текст.

Формат входных данных

Во входном файле содержится последовательность байтов, полученная после xor-шифрования. Входной файл не является текстовым и может содержать байты с произвольными кодами.

Формат выходных данных

Выведите исходных английский текст, который был зашифрован.

Примеры

xor.in	xor.out

Задача C. Xor-шифрование — 2

Имя входного файла:	`xor2.in`
Имя выходного файла:	`xor2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Каждый тест этой задачи — английский текст, зашифрованный с помощью xor-шифрования.

Перед шифрованием выбрана последовательность не более чем из пяти (и не менее чем из одного) случайных байтов b_0, b_1, ..., b_{k-1} ($k \leqslant 5$). После этого каждый байт исходного текста на позиции i (позиции нумеруются с нуля) и с кодом c был заменён на символ с кодом $c \wedge b_{i\%k}$. Гарантируется, что в исходном тексте встречались только печатные символы с кодами от 32 до 128 и переводы строк, а размер текста был не менее 1 Кб.

Восстановите исходный текст.

Формат входных данных

Во входном файле содержится последовательность байтов, полученная после xor-шифрования. Входной файл не является текстовым и может содержать байты с произвольными кодами.

Формат выходных данных

Выведите исходных английский текст, который был зашифрован.

Примеры

xor2.in	xor2.out

Задача D. Impossible to Solve

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

Программисты скажут, что эта задача нерешаема. Однако лингвисты и археологи решают даже более сложные криптографические задачи каждый день, поэтому вероятно сочтут эту очень простой.

Рассмотрим некоторую книгу, написанную на английском языке. Обозначим её текст за S. Обработаем строку S следующим образом: заменим в ней символы, коды которых не входят диапазон 32..127 на пробел. Назовём полученную строку X.

Выберем n от 2 до 20, а затем выберем случайную перестановку чисел от 1 до n и назовём её π. Разобьём X на блоки длины n, получим X_1, X_2, ..., X_t (последний блок X_t, возможно, содержит меньше n символов). После этого переупорядочим символы в каждом блоке используя перестановку π (последний блок X_t переупорядочивается, только если в нём ровно n символов). Например, если X_i = "*Contest*" и $\pi =< 2, 3, 1, 7, 4, 5, 6 >$, то результатом переупорядочивания станет "onCttes". Склеим полученные блоки и обозначим за Y полученный текст.

Выберем m от 2 до 20, а затем выберем случайную последовательность K длины m. Каждый элемент этой последовательности — это целое число от 0 до 127. Разобьём Y на блоки длины m, получим Y_1, Y_2, ..., Y_r (последний блок Y_r, возможно, содержит меньше m символов). Выполним операцию `xor` для каждого блока и последовательности K (ASCII-код j-го символа блока Y_i хорится с j-м числом в последовательности K, а затем берётся символ с полученным ASCII-кодом). Для последнего блока Y_r только первые $|Y_r|$ символов из последовательности K используется. Например, если Y_i = "*Contest*", а $K =< 1, 2, 1, 2, 1, 3, 0 >$, то результатом будет "Bmovdpt". Склеим полученные блоки и обозначим за Z полученный текст.

Вам дан Z. Восстановите X.

Формат входных данных

Входной файл к этой задаче бинарный, его размер — от 140 до 650 Кб. Он был создан путём применения описанной последовательности действий к некоторому классическому английскому тексту, взятому из проекта Gutenberg (https://www.gutenberg.org/)

Формат выходных данных

Выведите текст X.

Примеры

impossible.in	impossible.out

Задача E. Взлом Диффи-Хеллмана

Имя входного файла:	`dh.in`
Имя выходного файла:	`dh.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Алиса и Боб хотят вычислить общий ключ с помощью алгоритма Диффи-Хеллмана. Для этого они сначала публично выбрали числа g и простое p. Затем Алиса придумала секретное число a ($1 \leqslant a \leqslant 1000$), а Боб — секретное число b ($1 \leqslant b \leqslant 1000$). Алиса вычислила $A = g^a \mod p$, а Боб вычислил $B = g^b \mod p$, этими числами они публично обменялись.

Общий ключ K Алиса вычисляет как $B^a \mod p$, а Боб как $A^b \mod p$.

Вычислите ключ K, не зная секретных чисел Алисы и Боба.

Формат входных данных

В первой строке через пробел записаны два целых числа: g ($2 \leqslant g \leqslant 10\,000$) и p ($3 \leqslant p \leqslant 10\,000$). Гарантируется, что p — простое число.

Во второй строке записаны вычисленные A и B ($0 \leqslant A, B < p$).

Формат выходных данных

Выведите одно целое число — общий ключ K.

Примеры

dh.in	dh.out
10 1357 419 34	33