Реализация протокола RSA, часть 1. Теория чисел.

A. *Алгоритм Евклида*

Даны a, b ($0 \leqslant a, b \leqslant 10^{1000}$), вычислить НОД$(a, b)$.

Input	Output
12345 1239	3
234567634062957357737538483291768262603 4 234576625184956843677428152391786052009265	1000000007321

B. *Расширенный алгоритм Евклида*

Даны целые числа a, b, c, по модулю не превосходящие 10^{50}. Вывести любое целочисленное решение уравнения

$$ax + by = c$$

Если решения нет, вывести `NO SOLUTION`.

Указание: Вместо данного уравнения решим уравнение $ax + by =$ НОД(a, b), а потом домножим найденные значения x и y на $c \mathbin{/\!/}$ НОД(a, b). Для решения рассмотрите систему уравнений

$$\begin{cases} a \cdot 1 + b \cdot 0 = a \\ a \cdot 0 + b \cdot 1 = b \end{cases}$$

Затем для правых частей уравнений вычислять НОД(a, b), используя тождество

$$a\%b = a - b * (a \mathbin{/\!/} b)$$

а для левых частей выполнять соответствующие преобразования над коэффициентами при a и b. Например, найдём решение уравнения $11x + 5y = 3$. Для этого решим уравнение $11x + 5y = 1$ (т.к. НОД$(5, 11) = 1$)

$$\begin{cases} 11 \cdot 1 + 5 \cdot 0 = 11 \\ 11 \cdot 0 + 5 \cdot 1 = 5 \end{cases}$$

Вычтем из первого уравнения второе, домноженное на $11 \mathbin{/\!/} 5 = 2$ и поменяем их местами.

$$\begin{cases} 11 \cdot 0 + 5 \cdot 1 = 5 \\ 11 \cdot 1 + 5 \cdot (-2) = 1 \end{cases}$$

Вычтем из первого уравнения второе, домноженное на $5 \mathbin{/\!/} 1 = 5$ и поменяем их местами.

$$\begin{cases} 11 \cdot 1 + 5 \cdot (-2) = 1 \\ 11 \cdot (-5) + 5 \cdot 11 = 0 \end{cases}$$

В правой части второго уравнения стоит 0, значит в правой части первого НОД$(a, b) = 1$, а перед 11 и 5 стоят нужные нам коэффициенты — решение уравнния $11x + 5y = 1$. Домножив их на $3 \mathbin{/\!/}$ НОД$(5, 11) = 3$, получим $x = 3, y = -6$.

Input	Output
17 -24 1	-7 -5
12345 1239 1	NO SOLUTION
2364563534 1000900008 28	-713667094 1685993778

C. *Обратный элемент по модулю*

Даны числа a, m ($1 < m \leqslant 10^{50}, 1 \leqslant a < m$), такие что НОД$(a, m) = 1$.

Решить относительно x сравнение

$$ax \equiv 1 \pmod{m}$$

То есть требуется найти такое $0 \leqslant x < m$, что выполнено указанное сравнение.

Input	Output
7 12	7
765136569612394569781235 64 237649782365716527385612567	164643367340550431775137442

D. *Возведение в степень по модулю*
Даны натуральные числа a, n, m ($a < 10^9, n < 10^{100}, m < 10^{100}, a < m$). Вычислить

$$a^n \pmod m$$

Указание: Реализуйте быстрое возведение в степень и на каждой итерации берите остаток по указанному модулю, иначе даже быстрое возведение будет работать долго — числа очень большие.

Input	Output
2 10 1000	24
1785616459659263459 763457926347126953461987456163456 10000000000000000000000000	4287923560528638902346241
1785616459659263459 763457926347126953461987456163456 23645364574367523786	1724514821491748503

E. *Вычисление функции Эйлера*
Функция Эйлера $\varphi(n)$ равна количеству чисел от 1 до n, взаимно простых с n. Вот основные её свойства:

- Если p — простое число, то $\varphi(p) = p - 1$
- Если p — простое число, а n — натуральное, то $\varphi(p^n) = p^n - p^{n-1}$
- Если p и q взаимно простые, то $\varphi(pq) = \varphi(p) \cdot \varphi(q)$ (т.е. функция Эйлера "почти" мультипликативная)

  Из приведённых свойств есть очень простое и очень важное с практической точки зрения следствие:

  Если $n = pq$ и p и q — простые, то $\varphi(n) = (p - 1)(q - 1)$
- Если известно разложение числа n на простые множители:

  $$n = p_1^{a_1} \cdot p_2^{a_2} \cdot \dots \cdot p_k^{a_k}$$

  функцию Эйлера, опираясь на свойства выше, вычисляется по формуле:

  $$\varphi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdot \dots \cdot \left(1 - \frac{1}{p_k}\right)$$

  Или, что то же самое:

  $$\varphi(n) = p_1^{a_1 - 1}(p_1 - 1) \cdot p_2^{a_2 - 1}(p_2 - 1) \dots p_k^{a_k - 1}(p_k - 1)$$

Дано натуральное число n ($n < 10^{12}$), вывести $\varphi(n)$.

Input	Output
12	4
1000000009	1000000008
50064232156	25021166112
49	42

F. *Извлечение дискретного корня по модулю*
Даны натуральные числа n, b, m ($n < 10^{1000}, m < 10^{12}, b < m, \text{НОД}(n, \varphi(m)) = 1$).
Решить относительно x уравнение

$$x^n \equiv b \pmod m$$

Указание: алгоритм и его обоснование приведено в главе 3 (Заметки об алгоритме RSA, стр. 6).

Input	Output
3 14 55	9
17 12 50	22
2345764347 872345692 12874165001	6574312268