Important grading note: You must solve this problem without using any library methods that your programming language provides for greatest common divisor, modular exponentiation, modular multiplicative inverse, primality testing, or RSA key generation.

Each line of input poses a question from number theory; your program must print the answer, as detailed below.

Input

Each line of input consists of a category followed by one or more integers. All integers are non-negative and are less than 2^{31}. The five types of input lines are:

- **gcd a b**, where $a > 0$ and $b > 0$.
- **exp x y N**, where $N > 1$.
- **inverse a N**, where $N > a > 0$.
- **isprime p**, where $p > 5$.
- **key p q**, where p and q are prime and $p \neq q$.

Output

For each line of input, your program should produce a line of output, depending on the category:

- **gcd a b**: Print the greatest common divisor of a and b.
- **exp x y n**: Print x^y mod N, which must be non-negative and less than N.
- **inverse a N**: Print a^{-1} (mod N), which must be positive and less than N. If the inverse does not exist, print “none”.
- **isprime p**: Print “yes” if p passes the Fermat test for $a = 2$, $a = 3$, and $a = 5$; print “no” otherwise.
- **key p q**: Print the modulus, public exponent, and private exponent of the RSA key pair derived from p and q. The public exponent must be the smallest positive integer that works; q must be positive and less than N.