*"""  
The question is based on the pulverizer which means that there exist some  
values (positive or negative) of a and b for which ax + by = gcd(x,y).  
So if gcd(of all bills) = 0.1 ssd then the transaction is possible.  
  
Note the following properties of GCD which can be used in optimizing the code:  
gcd(x1,x2,x3,x4, ... ) = gcd(x1,gcd(x2,x3,x4,...)) this can be used to optimize the code  
  
Optimisations like finding GCD of pairs of numbers and making segment tree  
for of GCD can be done to make the worst case time complexity n ln n but  
is not expected in the solution.  
"""***from** math **import** gcd *# for finding GCD of numbers*n = int(input()) *# number of bills*values = [int(float(i) \* 10 \*\* 4) **for** i **in** input().split(**' '**)] *# converting input into integers*q = int(input()) *# number of queries*w = **False** *# flag for wrong query*negvals = 0 *# count of negative values***def** output(): *# prints the output* **global** w  
 **if** negvals == 0 **and not** w:  
 **if** 1000 % g == 0: *# Testing the GCD condition* print(**'y'**)  
 **else**:  
 print(**'n'**)  
 **else**:  
 w = **False** print(**'w'**)  
  
  
g = values[0]  
  
*# finding all GCDs***for** i **in** values[1:]:  
 **if** i > 0:  
 g = gcd(g, i)  
 **else**:  
 negvals += 1  
  
output()  
  
**for** i **in** range(q):  
 case, v = input().split(**' '**)  
 v = int(float(v) \* 10 \*\* 4)  
 **if** case == **'i'**:  
 **if** v > 0:  
 values.append(v)  
 g = gcd(g, v)  
 **else**:  
 values.append(v)  
 negvals += 1  
 **else**:  
 **if** v **in** values:  
 values.remove(v)  
 **if** v <= 0:  
 negvals -= 1  
 g = values[0]  
 **for** j **in** values[1:]:  
 g = gcd(g, j)  
 **else**:  
 w = **True** output()