**Question 1:**

Alexander The great, while roaming the stretch of Turkey, came across a wise man.

He asked the wise man, “Who is the greatest conqueror of all?”. The wise man replied, “A person with great strength and intelligence. Whosoever can solve my puzzle will go on to become the greatest!”.

The puzzle is as follows:

Given two integers ‘n1’ and ‘n2’, select two integers ‘a’ and ‘b’, such as to solve the equation (n1 \* a + n2 \* b = x). But there is a catch, ‘x’ is the smallest positive integer which satisfies the equation. Can you help Alexander become the greatest?

***Constraints***

1 <= T <= 1000

-10^7 <= a, b <= 10^7

0 <= n1, n2 <= 10^7

***Input Format***

The first line contains the number of Test cases T.

Next T lines contains two space-separated integers, n1 and n2.

***Output***

Print the value of x.

Test Case

***Example 1***

Input

1

34818 45632

Output

2

Explanation

Given n1 = 34818 and n2 = 45632, if we choose a = 3553 and b = -2711, we get

=> n1 \* a + n2 \* b = x

=> 34818 \* 3553 + 45632 \* (-2711)

=> 2

**Answer:**

from math import gcd as g

a=int(input())

b,c=map(int,input().split())

d=g(b,c)

print(d)

--------------------------------------------------------------------------------------

**Question 2:**

Cliffs (or) Peaks are defined as the place having magnitude higher than both its neighbours. Crates or Plateau are defined as the place having magnitude lower than both its

neighbours. From the given set of magnitudes find the highest distance between two consecutive peaks.

***Input Format***:

The first line has a single number "n" which is the number of values

The second line has "n" values which has the magnitudes of the places

***Output format***:

A single integer which is the highest distance between two consecutive peaks

***Sample Input:***

7

2 5 4 7 9 6 1

***Sample Output:***

3

***Explanation:***

In the given set, 5 and 9 are peaks.

5(index = 1) is higher than both its neighbours 2 and 4.

9(index = 4) is higher than its neighbours 7 and 6.

The difference in distance between them is 3.

**Answer:**

n = int(input())

a = list(map(int, input().split()))

peaks=[]

for i in range(n):

if(i==0):

if(a[i]>a[i+1]):

peaks.append(i)

elif(i==n-1):

if(a[i]>a[i-1]):

peaks.append(i)

elif(a[i-1]<a[i] and a[i]>a[i+1]):

peaks.append(i)

diff = [abs(peaks[i]-peaks[i+1]) for i in range(len(peaks)-1)]

print(max(diff))

--------------------------------------------------------------------------------------

**Question 3:**

Little Jill jumbled up the order of the letters in our dictionary. Now, Jack uses this list to find the smallest lexicographical string that can be made out of this new order. Can you help him?

(In mathematics, the lexicographic or lexicographical order is a generalization of the way words are alphabetically ordered based on the alphabetical order of their component letters.)

You are given a string P that denotes the new order of letters in the English dictionary.

You need to print the smallest lexicographic string made from the given string S.

Constraints

1 <= T <= 1000

Length (P) = 26

1 <= length (S) <= 100

All characters in the string S, P are in lowercase

Input Format

The first line contains number of test cases T

The second line has the string P

The third line has the string S

Output

Print a single string in a new line for every test case giving the result

Time Limit

1

Explanation

Example 1

Input

2

polikujmnhytgbvfredcxswqaz

abcd

qwryupcsfoghjkldezxvbintma

ativedoc

Output

bdca

codevita

Explanation

The transformed smallest lexicographical strings are in order they would be if order of letters are changed to string P

**Answer:**

for \_ in range(int(input())):

p = input()

s = input()

ans = ""

for i in p:

if i in s:

ans += i\*s.count(i)

print(ans)