**Prime factors**

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num\*\*0.5) + 1):

if num % i == 0:

return False

return True

def sum\_of\_prime\_factors(n):

prime\_factors\_sum = 0

for i in range(2, n + 1):

if n % i == 0 and is\_prime(i):

prime\_factors\_sum += i

return prime\_factors\_sum

# Input

n = int(input())

# Output

print(sum\_of\_prime\_factors(n))

**intersection**

def find\_intersection\_union(arr1, arr2):

# Find intersection

intersection = list(set(arr1) & set(arr2))

intersection.sort()

# Find union

union = list(set(arr1) | set(arr2))

union.sort()

return intersection, union

# Input reading

N, M = map(int, input().split())

arr1 = list(map(int, input().split()))

arr2 = list(map(int, input().split()))

# Find intersection and union

intersection, union = find\_intersection\_union(arr1, arr2)

# Output

if not intersection:

print(-1)

else:

print(\*intersection)

if not union:

print(-1)

else:

print(\*union)

**count**

s=input()

c=0

for I in s:

if I == " ":

c+=1

print(c+1)

**largest no**

def largestNumber(array):

if len(array)==1:

return str(array[0])

for i in range(len(array)) :

array[i]=str(array[i])

for i in range(len(array)):

for j in range(1+i,len(array)):

if array[j]+array[i]>array[i]+array[j]:

array[i],array[j]=array[j],array[i]

result=' '.join(array)

if(result=='0'\*len(result)):

return '0'

else:

return result

a = [89,54,5,532,3,4]

print(largestNumber(a))

**diff in array**

n=int(input())

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

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

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

x,y,z=0,0,0

for i in range(n):

x=(x^a[i])

for i in range(n-1):

y=(y^b[i])

for i in range(n-2):

z=(z^c[i])

a1=x^y

a2=y^z

print(a1)

print(a2)

8) def extractMaximum(ss):

num, res = 0, 0

# start traversing the given string

for i in range(len(ss)):

if ss[i] >= "0" and ss[i] <= "9":

num = num \* 10 + int(int(ss[i]) - 0)

else:

res = max(res, num)

num = 0

return max(res, num)

# Driver Code

ss = "100 564 365"

print(extractMaximum(ss))

12) n=int(input())

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

count=0

sum=0

for i in range(0,n-1):

for j in range(i+1):

sum=arr[i]+arr[j]

if sum%60==0:

count+=1

print(count)

**indexes**

n=int(input())

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

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

x=[0]\*n

for i in range (n) :

x[b[i]]=a[i]

for i in x:

print (i)

def make\_palindrome(s):

if s == s[::-1]:

return s # If the string is already a palindrome, return it as is

additional\_chars = ""

for i in range(len(s)):

if s[i:] == s[i:][::-1]:

additional\_chars = s[:i][::-1]

break

palindrome = s + additional\_chars

return palindrome

# Example usage:

input\_str = input()

result = make\_palindrome(input\_str)

print(f"To make '{input\_str}' a palindrome: '{result}'")