Count Divisors

You have been given 3 integers - l, r and k. Find how many numbers between l and r (both inclusive) are divisible by k. You do not need to print these numbers, you just have to find their count.

Input Format The first and only line of input contains 3 space separated integers l, r and k.

Output Format

Print the required answer on a single line.

Constraints

SAMPLE INPUT :

- 1 10 1

SAMPLE OUTPUT :

- 10

In [1]:

```
def CountDivisors(l,r,k):
    count=0
    for x in range (l,r+1):
        if x%k==0:
            count+=1
    print(count)
    return

li=input()
li=li.split(" ")
l=int(li[0])
r=int(li[1])
k=int(li[2])

CountDivisors(l,r,k)
```

```
1 10 1
10
```

In [ ]:

```

```

Factorial!

You have been given a positive integer N. You need to find and print the Factorial of this number. The Factorial of a positive integer N refers to the product of all number in the range from 1 to N. You can read more about the factorial of a number here.

Input Format

The first and only line of the input contains a single integer N denoting the number whose factorial you need to find.

Output Format

Output a single line denoting the factorial of the number N.

SAMPLE INPUT

- 2

SAMPLE OUTPUT

- 2

In [2]:

```
def factorial(num):
    fact=1
    for x in range(1,num+1):
        fact*=x
    print(fact)
    return



num=int(input())
factorial(num)
```

```
2
2
```

In [ ]:

```

```

Toggle String

You have been given a String S consisting of uppercase and lowercase English alphabets. You need to change the case of each alphabet in this String. That is, all the uppercase letters should be converted to lowercase and all the lowercase letters should be converted to uppercase. You need to then print the resultant String to output.

Input Format

The first and only line of input contains the String S

Output Format

Print the resultant String on a single line.

SAMPLE INPUT

abcdE

SAMPLE OUTPUT

ABCDe

In [3]:

```
def ToggleString(string):
    print(string.swapcase())

string=input()
ToggleString(string)
```

```
abcdE
ABCDe
```

In [ ]:

```

```

Palindromic String

You have been given a String S. You need to find and print whether this string is a palindrome or not. If yes, print "YES" (without quotes), else print "NO" (without quotes).

Input Format

The first and only line of input contains the String S. The String shall consist of lowercase English alphabets only.

Output Format

Print the required answer on a single line.

Note : String S consists of lowercase English Alphabets only.

SAMPLE INPUT

aba

SAMPLE OUTPUT

YES

In [4]:

```
def palindrome(string):
    if(string==string[::-1]):
        print('YES')
    else:
        print('NO')
string=input()
palindrome(string)
```

```
aba
YES
```

In [ ]:

```

```

Prime Number

You are given an integer N. You need to print the series of all prime numbers till N.

Input Format

The first and only line of the input contains a single integer N denoting the number till where you need to find the series of prime number.

Output Format

Print the desired output in single line separated by spaces.

Constraints

1<=N<=1000

SAMPLE INPUT

9

SAMPLE OUTPUT

2 3 5 7

In [29]:

```
def Prime(m):
    flag=True
    for i in range(2,(m//2)+1):
        if m%i==0:
            flag=0
            break
        else:
            flag=1
    if flag:
        print(m,end=' ')
def NPrime(N):
    for count in range(2,N+1):
        Prime(count)

num=int(input())
NPrime(num)
```

```
9
2 3 5 7
```

In [ ]:

```

```

Two Strings

Given two strings of equal length, you have to tell whether they both strings are identical.

Two strings S1 and S2 are said to be identical, if any of the permutation of string S1 is equal to the string S2. See Sample explanation for more details.

Input:

First line, contains an intger 'T' denoting no. of test cases. Each test consists of a single line, containing two space separated strings S1 and S2 of equal length.

Output:

For each test case, if any of the permutation of string S1 is equal to the string S2 print YES else print NO.

Constraints:

1<= T <=100 1<= |S1| = |S2| <= 10^5 String is made up of lower case letters only.

Note : Use Hashing Concept Only . Try to do it in O(string length) .

SAMPLE INPUT

3

- sumit mitsu
- ambuj jumba
- abhi hibb

SAMPLE OUTPUT

- YES
- YES
- NO

In [7]:

```
def TwoStrings(s1,s2):
    y=0
    for str2 in s2:
        if (str2 in s1) and (s1.count(str2)==s2.count(str2)):
            y=1
        else:
            print("NO")
            y=0
            break
    if y:
        print("YES")
    return
T=int(input())

for x in range(0,T):
    string=input()
    string=string.split()
    s1=string[0]
    s2=string[1]
    TwoStrings(s1,s2)
```

```
3
sumit mitsu
YES
ambuj jumba
YES
abhi hibb
NO
```

In [ ]:

```

```

Duration

Rahul is a very busy persion he dont wan't to waste his time . He keeps account of duration of each and every work. Now he don't even get time to calculate duration of works, So your job is to count the durations for each work and give it to rahul.

Input :

- First line will be given by N number of works
- Next N line will be given SH,SM,EH and EM each separated by space(SH=starting hr, SM=starting min, EH=ending hr, EM=ending min)

Output :

N lines with duration HH MM(hours and minutes separated by space)

SAMPLE INPUT

- 2
- 1 44 2 14
- 2 42 8 23

SAMPLE OUTPUT

- 0 30
- 5 41

In [6]:

```
def Duration(sh,sm,eh,em):

    MM=60-sm+em
    HH=abs(eh-sh)-1
    if MM<60:
        print(HH,MM)
    else:
        HH+=MM//60
        MM=MM%60
        print(HH,MM)

N=int(input())
for x in range(0,N):
    wt=input();time=[]
    for x in wt.split():
        time.append(int(x))

    Duration(time[0],time[1],time[2],time[3])
```

```
2
1 44 2 14
0 30
2 42 8 23
5 41
```

In [ ]:

```

```

Char Sum

Consider All lowercase Alphabets of the English language. Here we consider each alphabet from a to z to have a certain weight. The weight of the alphabet a is considered to be 1, b to be 2, c to be 3 and so on until z has a weight of 26. In short, the weight of the alphabet a is 1, and the weight of all other alphabets is the weight of its previous alphabet + 1.

Now, you have been given a String S consisting of lowercase English characters. You need to find the summation of weight of each character in this String.

For example, Consider the String aba

Here, the first character a has a weight of 1, the second character b has 2 and the third character a again has a weight of 1. So the summation here is equal to : 1+2+1=4

Input Format :

The first and only line of input contains the String S.

Output Format:

Print the required answer on a single line

Constraints :

$1 \leq |S| \leq 100$

SAMPLE INPUT

aba

SAMPLE OUTPUT

4

In [8]:

```
def CharSum(string):
    sum1=0
    for x in string:
        num=ord(x)
        x=num-96
        sum1+=x
    print(sum1)
string=input()
CharSum(string)
```

```
aba
4
```

In [ ]:

```
```

Play with numbers

You are given an array of n numbers and q queries. For each query you have to print the floor of the expected value(mean) of the subarray from L to R.

Input :

- First line contains two integers N and Q denoting number of array elements and number of queries.
- Next line contains N space seperated integers denoting array elements.
- Next Q lines contain two integers L and R(indices of the array).

**Output :

print a single integer denoting the answer.

Constrints :

- 1<=N ,Q,L,R <=10^6
- 1<= Array elements <= 10^9

NOTE: Use Fast I/O

SAMPLE INPUT

- 5 3
- 1 2 3 4 5
- 1 3
- 2 4
- 2 5

SAMPLE OUTPUT :

- 2
- 3
- 3

In [28]:

```
# By ANIL PETER
def PlayWithNumbers(ll,ul,num):
    x=ul-1;y=ll-2
    if y>=0:
        print((int((num[ul-1]-num[ll-2])/(ul-ll+1))))
    else:
        print((int(num[x]/(ul-ll+1))))
def split(list1):
    final=[]
    for x in list1.split():
        final.append(int(x))
    return final

input1=split(input())
N=int(input1[0])


li=split(input())

mx=[];my=0
for x in range(0,len(li)):
    my+=li[x]
    mx.append(my)


for x in range(0,input1[1]):
    limit=input()
    x=split(limit)
    PlayWithNumbers((x[0]),(x[1]),mx)
```

```
5 3
1 2 3 4 5
1 3
2
2 4
3
2 5
3
```

In [31]:

```
#Clear Description by Akash Sinha but same logic

# Read no of array elements and no of queries
n = input().split()
n[0], n[1] = int(n[0]), int(n[1])

# Read array elements
a = input().split()

sum = [] # Initialize cumulative sum array

# Cumulative Sum
for i in range(0, n[0]):
    if i == 0:
        sum.append(int(a[i]))
    else:
        sum.append(int(sum[i-1]) + int(a[i]))

del a

# Read each query and calculate the average
for k in range(0, n[1]):
    inq = input().split()
    i = int(inq[0])
    j = int(inq[1])
    if i > 1:
        print((sum[j-1] - sum[i-2]) // (j-i+1))
    else:
        print(sum[j-1] // (j-i+1))
```

```
5 3
1 2 3 4 5
1 3
2
2 4
3
2 5
3
```

In [ ]: