In [ ]:
```
# Function to toggle string characters
s="abc"
s.islower() # check the case of character
s.isupper()

s.lower()   #check the case
s.upper()
s.split()
list(s)
```

In [ ]:
```
# Function to Toffle string characters

def togglestring(s):                                    # Convert the string into a
    s=list(s)
    t=[]
    for c in s:
        if c.islower():
            t.append(c.upper())
        else:
            t.append(c.lower())
    return"".join(t)

togglestring("abC")
```

Problem : Duration

Input: start time , end time (HH,MM)

```
HH-{00,23,00, 02,03..... 23}
MM-{00,01,02,03.......59}
HH MM-{00 00, 23,59}
```

output: time differnce in HH MM

In [ ]:
```
# Calculate the time difference as total number of
# Convert the total mintues into HH MM

s= "1 44 2 14"
def minuteDiffernce(s):
    s=s.split()
    sh=int(s[0])
    sm=int(s[1])
    eh=int(s[2])
    em=int(s[3])
    startmintues = (sh*60) + sm
    endmintues = (eh*60) +em
    return  endmintues-startmintues
def outputTimeformat(mintues):                          #convert minutes to HH MM
    hh=minutes//60
    mm=minutes%60
    print(hh,mm)
    return
minutes=minuteDiffernce(s)
outputTimeformat(minutes)
```

play with numbers

line1 : array size(n),no of queries(q) line2 : n array elements next q lines : query -sub array of the original

5 3 1 2 3 4 5

In [ ]:

In [ ]:

In [ ]:
```
# Two strings comparision
def twoStrings(s1,s2):
    s1=sorted(s1)
    s2=sorted(s2)
    if s1==s2:
        print("YES")
    else:
        print("NO")

s=int(input())
for i in range(s):
    s1=input()
    l=s1.split()
twoStrings(l[0],l[1])
```

In [ ]:

In [ ]:
```
# prime number

def prime(n):
    for i in range(2,n):
        if n%i==0:
            return False
    return True
def generateprime(num):
    for i in range (2,num+1):
        if prime(i):
            print(i, end=" ")
    return
num=int(input())
generateprime(num)
```

In [ ]:
```
# toggle string
def toggle(n):
    for i in n:
        if i==i.upper():
            s=i.lower()
        elif i==i.lower():
            s=i.upper()
        print(s,end="")
n=input()
toggle(n)
```

In [ ]:
```
#character sum
def charactersum(s):
    sum=0
    for i in range(0, len(s)):
        if s>=chr(97) and s<= chr(122) :
             sum= sum+(ord(s[i])-96)
    print(sum)
s=input(" ")
charactersum(s)
```

In [26]:
```
#proper number
def perfect(n):
    sum=0
    for i in range(1,n):
        if n%i==0:
            sum = sum+i
    if sum==n:
        return True
    return False
def generateperfect(num):
    for n in range(1,num):
        if perfect(n):
            print(n)
    return
num=int(input())
#n=int(input())
generateperfect(num)
```

```
50
6
28
```

In [4]:
```
#count letter and divisors
def countLetters(s):
    count1=0
    count2=0
    for i in s:
        if (i.isdigit()):
            count1=count1+1
        elif (i.isalpha()):
            count2=count2+1
    print(count1)
    print(count2)
s=input()
countLetters(s)
```

```
23rewvd
2
5
```

In [25]:
```
#maxremainder
def maxremainder(n):
    rem=0
    s=[]
    maxn=0
    r=0
    for i in range(1,n):
         #if n%i:
            s.append(n%i)

    #r=max(s)
    print(s)
    print(s.index(max(s)))

n=int(input())
maxremainder(n)
```

```
5
[0, 1, 2, 1]
2
```

In [27]:
```
def isSpecialNumber(n,p):
    if numberprimeFactors(n) >= p:
        return True
    return False

# Function to check if number is prime

def isprime(n):
    flag=1
    if n==2:
        return True
    for i in range(3,n//2+1):
        if n%i==0:
            flag=0
            return False
    if flag==1:
        return True
#Function to check if number of prime factors  for a given number

def numberprimeFactors(n):
    if isprime(n):
        return 1
    count = 0
    for i in range(2,n//2+1):
        if isprime(i) and n%i == 0:
            count+=1
    return count

numberprimeFactors(30)
isSpecialNumber(8,2)
```

Out[27]: True

In [1]:
```
def palindrome(s):
    sl=s[::-1]
    if sl==s:
        print("YES")
    else:
        print("NO")
    return
s=input()
palindrome(s)
```

```
abd
NO
```

Factorial

In [2]:
```
def factorial(n):
    fact=1
    for i in range(1,n+1):
        fact=fact*i
    print(fact)
    return
n=int(input())
factorial(n)
```

```
4
24
```

Play with number

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.

In [3]:
```
n=input().split()
n[0],n[1]=int(n[0]),int(n[1])

a = input().split()
sum = []

#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
#print(sum[n[0]-1])
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
2 3
2
2 5
3
2 4
3
```

In [ ]:
```
```