In [1]:
```
# funtion to print yes or no if the factors of the given number is prime or

def prime(n):
    c=0
    for i in range (1,n+1):
        if (n%i==0):
            c=c+1
    if(c==2):
        print("YES")
    else:
        print("NO")
n=int(input())
prime(n)
```

```
7
YES
```

Problem Play with numbers

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

a=input().split()
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
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))

#print(sum[n[0]-1])
```

```
5 3
1 2 3 4 5
2 3
6
1 4
2
```

Type *Markdown* and LaTeX: α^2

In [ ]:

```
def factor(n):
    fact=0
    for i in range(2,n):
        if(n%i==0):
            fact=fact+1
    if(fact==2):
        print("YES")
    else:
        print("NO")
n1=int(input())
for i in range(n1):
    n=int(input())
    factor(n)
```

Problem:Special Number

In [7]:

```
# Function to determine if a number is special or not
# Function to check prime  if number is prime
# Function to determine the number of prime factors

def isSpecialNumber(n,p):
    if numberPrimeFactors(n)>=p:
        return True
    return False

def is_prime(n):
    flag=1
    if n==2:
        return True
    for i in range(2,n//2+1):
        if (n%i==0):
            flag=0
            return False

    if flag==1:
        return True

def numberPrimeFactors(n):
    if is_prime(n):
        return 1
    count=0
    for i in range(2,n//2+1):
        if is_prime(i) and n%i==0:
            count=count+1
    return count
isSpecialNumber(6,2)



def solution2():
    p=int(input())
    t=int(input())
    for i in range(0,t):
        n=int(input())
        if isSpecialNumber(n,p):
            print("YES")
        else:
            print("NO")
#solution2()
isSpecialNumber(30,2)
```

Out[7]: True

In [11]:
```
def hi(n):
    hr=0
    v=n
    for i in  range(n-1,n//2,-1):
        r=n%i
        if r>hr:
            hr=r
            v=i
    print(v)
    return
hi(30)
```

```
16
```

Tuples

t1 = ()
li = []

Difference between Lists and Tuples

Lists are mutable - can be changed /modified

- Used to access ,Modify ,Add,Delete data

Tuple are immutable - Cannot be changed once initialised

- Tuples are generally used to access data only

In [12]:
```
t1 = (1,2,3,5,8)
t1
```

Out[12]:
```
(1, 2, 3, 5, 8)
```

In [16]:
```
t1=(1,2,3,5,8)
t1[3]
```

Out[16]:
```
5
```

In [29]:
```
t1[len(t1)//2:]
```

Out[29]:
```
(3, 5, 8)
```

Dictionories

IT Works on the concept of Set
Unique Data
Keys,Values

Key is the unique identifier for a value
Value is data that can be accessed with a key

In [30]:
```
d1={"k1":"value1","k2":"value2"}
d1
```

Out[30]: {'k1': 'value1', 'k2': 'value2'}

In [32]:
```
d1["k2"] # Accessing the value with key k2
```

Out[32]: 'value2'

In [35]:
```
d1.keys()# Returns list of all keys

```

Out[35]: dict_keys(['k1', 'k2'])

In [36]:
```
d1.values() # returns list of all values
```

Out[36]: dict_values(['value1', 'value2'])

In [37]:
```
d1.items()  #  returns list of tuples of keys and values
```

Out[37]: dict_items([('k1', 'value1'), ('k2', 'value2')])

In [38]:
```
d1["k3"]="value3"  # Adding an element to the dictonaty
```

Out[38]: {'k1': 'value1', 'k2': 'value2', 'k3': 'value3'}

In [39]:
```
d1["k3"]="value4" # This is Upadating an element
d1
```

Out[39]: {'k1': 'value1', 'k2': 'value2', 'k3': 'value4'}

In [44]:
```
d1.pop("k3") # Delete the last element

d1

```

Out[44]: {'k1': 'value1', 'k2': 'value2'}

In [53]:
```
"k1 " in d1
```

Out[53]: False

Contacts Application

- Add Contact
- Search for contact
- List all contacts
- Modify contact
- Remove contact

In [47]:
```
#contacts={}
def addContact(name,phone):
    # Verify the contact doesnot already exist
    if name not in contacts:
        contacts[name]=phone
        print("Contact Added")
    else:
        print("Contact %s already exists" % name)
    return
name=input()
phone=int(input())
print(contacts)
addContact(name,phone)

```

```
abc
123
{'abc': 1233}
Contact abc already exists
```

In [11]:
```
contacts


```

Out[11]:
```
{'anu': 123}
```

In [18]:
```
def searchContacts(name):
    if name in contacts:
        print(name,"exists",contacts[name])
    else:
        print("%s does not exist" % name)
    return
name=input()
searchContacts(name)
```

```
sdfreg
sdfreg does not exist
```

In [27]:
```
def importContacts(newConctacts):
    contacts.update(newContacts)
    print(len(newContacts.keys()),"contacts  added successfully")
    return
name2=input()
name3=input()
phone2=input()
phone3=input()
newContacts={name2:phone2,name3:phone3}
importContacts(newContacts)
```

```
fdgh
4566
hjuj
789
2 contacts  added successfully
```

In [33]:
```
contacts

```

Out[33]:
```
{'anu': 123,
 'name2': 'phone2',
 'name3': 'phone3',
 'fdgh': 'hjuj',
 '4566': '789'}
```

In [43]:
```
def removeContacts(name):
    if name in contacts:
        contacts.pop(name)
        print("%s removed"% name)
    else:
        print("%s not does not exists "% name)
    return
name=input()
removeContacts(name)


```

```
gfhth
gfhth not does not exists
```

In [42]:
```
contacts

```

Out[42]:
```
{}
```

Packages and Modules

Package---> Collection of Modules(PythonFile.py)

Sub Package --->

Module----> Sinle Python file containing various classes and methods(Functions) (Collection of Modules called Package)

Package-->Subpackages-->Modules-->Functions

In [ ]:
```
import math

math.ceil(123.45467)

math.floor(2344.56688999)

```

In [88]:
```
math.pi
```

Out[88]:
```
3.141592653589793
```

In [83]:
```
1 math.tan(90)
```

Out[83]: -1.995200412208242

In [84]:
```
1 from math import  ceil
```

In [85]:
```
1 math.pi
```

Out[85]: 3.141592653589793

In [86]:
```
1 ceil(34.5)
```

Out[86]: 35

In [87]:
```
1 floor(3.677)
```

Out[87]: 3

In [90]:
```
1 import random
2 def GenerateRandomNumbers(n,lb,ub):
3     for i in range(0,n):
4         print(random.randint(lb,ub),end=" ")
5 GenerateRandomNumbers(20,0,100)
```

59 80 88 62 31 47 64 48 61 26 10 80 53 11 13 42 88 27 20 10

In [97]:
```
1 from Packages import *
2 numerical.numberPrimeFactors(7)
```

Out[97]: 1

In [10]:
```
1 li=[9,1,3,0,2]
2 li.sort()
3 li
```

Out[10]: [0, 1, 2, 3, 9]

In [11]:
```
1 li[-2:]
```

Out[11]: [3, 9]

In [12]:
```
1 sum(li[-2:])
```

Out[12]: 12

In [24]:
```
n=int(input())
s=input().split()
li=[]
b=[]
for i in  s:
    li.append(int(i))
li.sort()
li


```

```
1
0 4 -3 8 9 2
```

Out[24]:
```
[-3, 0, 2, 4, 8, 9]
```

In [27]:
```
n1=int(input())
for i in range (1,n1+1):
    n=int(input())
    print(n)
    if(n==42):
        break
```

```
5
1
1
2
2
4
4
42
42
```

In [ ]:
```

```