#Input & Output - Say "Hello, World!"

print("\"Hello, World!\"")

#Input & Output - Hello World with tab

print("Hello World\t"\*2)

#Input & Output - Hello World with a new line

print("Hello World")

print("Hello World")

#Input & Output - Student Details

name=str(input())

age=int(input())

CGPA=float(input())

grade=input()

cg=int(CGPA\*100)/100.0

print(f"Name: {name}")

print(f"Age: {age}")

print(f"CGPA: {cg:.2f}")

print(f"Grade: {grade}")

#Input & Output - ASCII Values - I

char=input()

num=ord(char)

print(num)

#Input & Output - ASCII Values - II

num=int(input())

char=chr(num)

print(char)

#Input & Output - Round Off

import math

num=float(input())

c=math.floor(num)

d=math.ceil(num)

f=int(num)

print(c)

print(d)

print(f)

#Operators - Fencing the Ground

l=int(input())

b=int(input())

peri=2\*(l+b)

area=l\*b

print(f"The required length is {peri} m")

print(f"The required area of carpet is {area} sqm")

#Operators - The Newspaper Agency

copies=int(input())

sp=int(input())

cp=int(input())

sold=copies\*sp

printed=copies\*cp

profit=sold-printed-100

print(profit)

#Operators - Harry Potter

import string

n=input()

n=n.strip()

n=n.strip(string.punctuation)

print(int(n[0])+int(n[-1]))

#Operators - Splitting into the teams

num\_friends=int(input())

num\_teams=int(input())

team=num\_friends//num\_teams

left=num\_friends%num\_teams

print(f"The number of friends in each team is {team} and left out is {left}")

#Operators - Debt Repay

p=float(input())

n=float(input())

r=float(input())

interest=(p\*n\*r)/100

amount=p+interest

discount=(interest\*2)/100

final=amount-discount

print(f"{interest:.2f}")

print(f"{amount:.2f}")

print(f"{discount:.2f}")

print(f"{final:.2f}")

#Operators - 3 Psychos

x1=int(input())

y1=int(input())

x2=int(input())

y2=int(input())

x=(x1+x2)/2

y=(y1+y2)/2

print(f"Arun's house is located at({x},{y})")

#Operators - Hop n Hop

import math

x1=3

y1=4

x2=int(input())

y2=int(input())

z=math.sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1))

print(int(z))

#Operators - Dollars & Cents

d1=int(input())

c1=int(input())

d2=int(input())

c2=int(input())

dollar=d1+d2

cents=c1+c2

c=cents//100

dollar+=c

cents%=100

print(dollar)

print(cents)

#Operators - Treasure Hunter

total=int(input())

ben=int(input())

black=int(input())

ben\_gold=total\*ben//100

total=total-ben\_gold

black\_gold=total\*black//100

total=total-black\_gold

pirate\_gold=total//3

print(f"{ben\_gold}")

print(f"{black\_gold}")

print(f"{pirate\_gold}")

#Operators - Reverse a 3-digit number

n=int(input())

last=n%10

middle=int((n/10)%10)

first=int(n/100)

print(f"{last}{middle}{first}")

#Operators - Tic Tac Toe

n=int(input())

a= n-1

row=a//3

column=a%3

print(f"{row} {column}")

#Decision Making - Checking Alphabets

d=input()

if d=='a'or d=='e'or d=='i'or d=='o'or d=='u'or d=='A'or d=='E'or d=='I'or d=='O'or d=='U':

print("Vowel")

elif('A'<=d<='Z')or('a'<=d<='z'):

print("Consonant")

else:

print("Not an alphabet")

#Decision Making - Electricity Bill

units=int(input())

if units<=200:

cost=int(units\*0.5)

print(f"Rs.{cost}")

elif units<=400:

cost=int((units\*0.65)+100)

print(f"Rs.{cost}")

elif units<=600:

cost=int((units\*0.80)+200)

print(f"Rs.{cost}")

else:

cost=int((units\*1.25)+425)

print(f"Rs.{cost}")

#Decision Making - Online shopping

fp=int(input())

fd=int(input())

fs=int(input())

sp=int(input())

sd=int(input())

ss=int(input())

ap=int(input())

ad=int(input())

sa=int(input())

f=int(fp+fs-(fd/100)\*fp)

s=int(sp+ss-(sd/100)\*sp)

a=int(ap+sa-(ad/100)\*ap)

print(f"In Flipkart: Rs.{f}")

print(f"In Snapdeal: Rs.{s}")

print(f"In Amazon: Rs.{a}")

if f<s and f<a:

print("Choose Flipkart")

elif s<f and s<a:

print("Choose Snapdeal")

elif a<f and a<s:

print("Choose Amazon")

else:

print("Choose Flipkart")

#Decision Making - Hotel Tariff Calculator

month=int(input())

rent=int(input())

days=int(input())

if month==4 or month==5 or month==6 or month==11 or month==12:

tarrif=int(rent\*(20/100))

cost=(rent+tarrif)\*days

print(cost)

elif month==1 or month==2 or month==3 or month==7 or month==8 or month==9 or month==10:

cost=rent\*days

print(cost)

else:

print("Invalid Input")

#Decision Making - Gift for Birthday

year=int(input())

if (year%4==0 and year%100!=0) or year%400==0:

print(f"{year} is a leap year")

else:

print(f"{year} is not a leap year")

#Decision Making - Trendy number

n=int(input())

if 100<=n<=999:

mid=(n//10)%10

if mid%3==0:

print("Trendy Number")

else:

print("Not a Trendy Number")

else:

print("Invalid Number")

#Time Sheet

sun=int(input())

mon=int(input())

tues=int(input())

wed=int(input())

thur=int(input())

fri=int(input())

sat=int(input())

week=mon+tues+wed+thur+fri

sal=week\*100

if mon>8:

sal+=(mon-8)\*15

if tues>8:

sal+=(tues-8)\*15

if wed>8:

sal+=(wed-8)\*15

if thur>8:

sal+=(thur-8)\*15

if fri>8:

sal+=(fri-8)\*15

if week>40:

sal+=(week-40)\*25

if sat>0:

sat\_bon=int(sat\*0.25\*100)

sal+=(sat\_bon)+(sat\*100)

if sun>0:

sun\_sal=int(sun\*0.50\*100)

sal+=(sun\*100)+sun\_sal

print(sal)

#Decision Making - Number of Days

year=int(input())

month=int(input())

if 1900>year>9999:

print("0")

elif 1900<=year<=9999 and month==1 or month==3 or month==5 or month==7 or month==8 or month==10 or month==12:

print("31 Days")

elif 1900<=year<=9999 and month==4 or month==6 or month==9 or month==11:

print("30 Days")

elif (year%4==0 and year%100!=0) or year%400==0 and month==2:

print("29 Days")

else:

print("28 Days")

#Decision Making - Scholarship

age=int(input())

yop=int(input())

income=int(input())

arrear=int(input())

scholarship=float(input())

attendance=float(input())

if yop>=2021 and arrear<=2 and income<=200000 and 18<=age<21 and scholarship>=60 and attendance>=80:

print("Eligible")

elif yop>=2021 and arrear>2 and 200000<=income<250000 and 18<=age<21 and scholarship>=80 and attendance>=90:

print("Partially Eligible")

else:

print("Not Eligible")

#Decision Making - Mango tree

rows=int(input())

cols=int(input())

tree=int(input())

if tree<=cols: #1st row

print("Yes")

elif tree%cols==1: #1st cols

print("Yes")

elif tree%cols==0: #last cols

print("Yes")

else:

print("No")

#Decision Making - Cricket

import math

t\_balls=int(input())

t\_runs=int(input())

score\_runs=int(input())

n\_balls=int(input())

t\_overs=t\_balls//6

overs\_whole=n\_balls//6

overs\_half=n\_balls%6

overs\_finished=overs\_whole+(overs\_half)/10

crr=score\_runs/overs\_finished

trr=t\_runs/t\_overs

print(f"{t\_overs}")

print(f"{overs\_finished:.1f}")

print(f"{crr:.1f}")

print(f"{trr:.1f}")

if crr>=trr:

print("Eligible to Win")

else:

print("Not Eligible to Win")

#Control Statements - Multiplication Table

n=int(input("Enter n"))

m=int(input("\nEnter m"))

print(f"\nThe multiplication table of {n} is")

for i in range(1,m+1):

print(f"{i}\*{n}={i\*n}")

#Control Statements - Print Prime Numbers in a range

num=int(input())

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

flag=True

if n<2:

continue

if n==2:

print("2",end=" ")

continue

for i in range(2,n):

if n%i==0:

flag=False

break

if flag:

print(n,end=" ")

#Control Statements - Special number

m=int(input())

n=int(input())

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

if((i//10)+(i%10))+((i//10)\*(i%10))==i:

print(i)

#Control Statements - Amoeba Multiplication

n\_terms=int(input())

n1=0

n2=1

count=0

while count<n\_terms-1:

n3=n1+n2

n1=n2

n2=n3

count+=1

print(n1)

#Control Statements - Number Series

n=int(input())

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

if i%2==0:

num=(i\*\*2)-2

print(num,end=" ")

else:

num=(i\*\*2)-1

print(num,end=" ")

#Control Statements - Hollow square pattern

n=int(input())

for i in range(n):

for j in range(n):

if i==0 or i==n-1 or j==0 or j==n-1:

print("\*",end="")

else:

print(" ",end="")

print()

#Control Statements - Treasure Finder

import math

box1=int(input())

box2=int(input())

box3=int(input())

if box1>box2 and box1>box3:

if box2>box3:

print(f"The treasure is in the box which has the number {box2}")

else:

print(f"The treasure is in the box which has the number {box3}")

if box2>box1 and box2>box3:

if box1>box3:

print(f"The treasure is in the box which has the number {box1}")

else:

print(f"The treasure is in the box which has the number {box3}")

if box3>box1 and box3>box2:

if box1>box2:

print(f"The treasure is in the box which has the number {box1}")

else:

print(f"The treasure is in the box which has the number {box2}")

code=math.gcd(math.gcd(box1,box2),box3)

print(f"The code to open the box is {code}")

#Control Statements - Collatz problem

n=int(input())

print(n)

count=0

while n!=1:

if n%2==0:

n=n//2

print(n)

else:

n=(3\*n)+1

print(n)

count+=1

print(count)

#Control Statements - Strong Number

import math

n=int(input())

sum=0

num=n

while n!=0:

rem=n%10

sum+=math.factorial(rem)

n//=10

if sum==num:

print("Yes")

else:

print("No")

#Control Statements - Inverted right-angled triangle

n=int(input())

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

for j in range(i):

print("\*",end="")

print()

#Control Statements - Sum of digit till single digit

n=int(input())

while n>=10:

sum=0

while n>0:

rem=n%10

sum+=rem

n//=10

n=sum

print(n)

#Control Statements - Kaprekar number

n=int(input())

sq=n\*\*2

count=0

num=n

while n>0:

n//=10

count+=1

right=sq%(10\*\*count)

left=sq//(10\*\*count)

s=right+left

if s==num:

print("Kaprekar Number")

else:

print("Not a Kaprekar Number")

#Control Statements - Trapezium Pattern

n=int(input())

left=1

right=(n\*n)+1

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

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

print("--",end="")

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

print(left,end="\*")

left+=1

for k in range(1,i+1):

print(right,end="")

if k<i:

print("\*",end="")

right+=1

print()

right=right-(i-1)\*2-1

#Arrays 1D - Same or Not

import array as arr

n1=int(input())

n2=int(input())

a1=arr.array('i')

for i in range(n1):

a1.append(int(input()))

a2=arr.array('i')

for i in range(n2):

a2.append(int(input()))

length1=len(a1)

length2=len(a2)

s1=sum(a1)

s2=sum(a2)

if length1==length2 and s1==s2:

print("Same")

else:

print("Not Same")

#Arrays 1D - Count distinct elements

import array as arr

n=int(input())

a=arr.array('i')

count=0

for i in range(n):

a.append(int(input()))

s1=set(a)

for j in s1:

count+=1

print(f"There are {count} distinct element in the array.")

#Arrays 1D - Compatible array

n1=int(input().strip())

a1=[]

for i in range(n1):

a1.append(int(input().strip()))

n2=int(input().strip())

a2=[]

for i in range(n2):

a2.append(int(input().strip()))

if n1!=n2:

print("Incompatible")

else:

c=True

for i in range(n1):

if a1[i]<a2[i]:

c=False

break

if c:

print("Compatible")

else:

print("Incompatible")

#Arrays 1D - Sum of even numbers and odd numbers

n1=int(input().strip())

s\_even=0

s\_odd=0

a1=[]

for i in range(n1):

a1.append(int(input().strip()))

for j in a1:

if j%2==0:

s\_even+=j

else:

s\_odd+=j

print(f"The sum of the even numbers in the array is {s\_even}")

print(f"The sum of the odd numbers in the array is {s\_odd}")

#Arrays 1D - Ascending order

n=int(input())

a=[]

for i in range(n):

a.append(int(input()))

j=sorted(a)

print("The Sorted array is:")

for k in j:

print(k)

#Arrays 1D - Queue

n,m = map(int,input().split())#space separated input

groups=list(map(int,input().split()))#array elements space separted

buses=0 #n=4 m=4

capacity=0

for g in groups: #3 3 3 3

if capacity+g<=m: #3<=4? | 3+3<=4?|

capacity+=g

else:

buses+=1 #1

capacity=g #3

if capacity>0:

buses+=1

print(buses)

#Arrays 1D - Array insertion

n=int(input())

a=[]

for i in range(n):

a.append(int(input()))

position=int(input())

if position>len(a):

print("Invalid Input")

else:

element=int(input())

a.insert(position-1,element)

print("Array after insertion is")

for i in a:

print(i)

#Arrays 1D - Remove duplicate elements

from collections import OrderedDict

n=int(input())

L=[]

for i in range(n):

L.append(int(input()))

dup=list(OrderedDict.fromkeys(L))

for i in dup:

print(i)

#Arrays 1D - Online Game

n=int(input())

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

left=0

right=n-1

while left<right:

while a[left]%2==0:

left+=1

while a[right]%2!=0:

right-=1

if left<right:

t=a[left]

a[left]=a[right]

a[right]=t

left+=1

right-=1

print("Array after Segregation")

for i in range(len(a)):

print(a[i],end=" ")

#Arrays 1D - Toyland

n=int(input().strip())

house=[]

for i in range(n):

house\_num,pos=map(int,(input().strip().split()))

house.append((house\_num,pos))

def get\_position(houses):

return houses[1]

house.sort(key=get\_position)

max\_gap=0

result=(0,0)

for i in range(1,n):

gap=house[i][1]-house[i-1][1]

if gap>max\_gap:

max\_gap=gap

result=(house[i-1][0],house[i][0])

print(min(result),max(result))

#Array 1D - Pair the Container

N=int(input())

container=list(map(int,input().strip().split()))

container.sort()

pairs=[]

left=0

right=N-1

while left<right:

pairs.append((container[right],container[left]))

left+=1

right-=1

if left==right:

pairs.append((container[left],0))

for pair in pairs:

print(pair[0],pair[1])

#Arrays 1D - Smallest Positive Missing Number

n=int(input())

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

m=1

positive=[]

for i in range(n):

if m<a[i]:

positive.append(a[i])

elif m==a[i]:

m+=1

while(positive.count(m)):

positive.remove(m)

m+=1#3

print(m)

#another method

n=int(input())

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

m=1

positive=[]

for i in range(n):

if a[i]>0:

positive.append(a[i])

positive.sort()

while True:

if a not in positive:

print(m)

break

m+=1

#Arrays 2D - Transpose Matrix

n=int(input())

arr=[]

for i in range(n):

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

arr.append(L)

for i in range(n):

for j in range(n):

print(arr[i][j],end=" ")

print()

print("Transpose matrix is:")

for i in range(n):

for j in range(n):

print(arr[j][i],end=" ")

print()

#Arrays 2D - Upper triangular matrix

n=int(input())

arr=[]

for i in range(n):

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

arr.append(L)

utm=True

for i in range(n):

for j in range(i):

if arr[i][j]!=0:

utm=False

if utm==True:

print("Upper triangular matrix")

else:

print("Not an Upper triangular matrix")

#Arrays 2D - Maximum element in each column

m=int(input())

n=int(input())

matrix=[]

for i in range(m):

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

matrix.append(L)

for i in range(n):#column

maxi=matrix[0][i]

for j in range(1,m):#row

if matrix[j][i]>maxi:

maxi=matrix[j][i]

print(maxi)

#Arrays 2D - Matrix Multiplication

n1,n2=map(int,input().split())

m1=[]

for i in range(n1):

a1=list(map(int,input().split()))

m1.append(a1)

m2=[]

for i in range(n2):

a2=list(map(int,input().split()))

m2.append(a2)

mul=[[0]\*n2 for j in range(n2)]

for i in range(n1):

for j in range(n2):

for k in range(n2):

mul[i][j]+=m1[i][k]\*m2[k][j]

for m in mul:

print(' '.join(map(str,m)))

#Arrays 2D - Sum of Zig-Zag

m=int(input())

n=int(input())

a=[]

for i in range(n):

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

a.append(L)

sum\_of\_zigzag=0

for j in range(n):

sum\_of\_zigzag+=a[0][j]

for i in range(1,m-1):

sum\_of\_zigzag+=a[i][m-i-1]

for j in range(m):

sum\_of\_zigzag+=a[m-1][j]

print(f"Sum of Zig-Zag pattern is {sum\_of\_zigzag}")

#Arrays 2D - Move all zeroes

T=int(input())

L=[]

oz=[]

for i in range(T):

a=input()

L.append(a)

for i in L:

ones=i.count('1')

zeroes=len(i)-ones

ans='1'\*ones+'0'\*zeroes

oz.append(ans)

for i in oz:

print(i)

#Array 2D - Uniformity Matrix

n=int(input())

a=[]

odd=True

even=True

for i in range(n):

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

a.append(L)

for i in range(n):

for j in range(n):

if a[i][j]%2==0:

even=False

if a[i][j]%2!=0:

odd=False

if odd or even:

print("Yes")

else:

print("No")

#Arrays 2D - Matrix Rotation

n=int(input())

arr=[]

for i in range(n):

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

arr.append(L)

transpose=[[arr[i][j] for i in range(n)]for j in range(n)]

for i in transpose:

i.reverse()

for i in range(n):

for j in range(n):

print(transpose[i][j],end=" ")

print()

#Function- Sum of list

n=int(input())

L=[]

for i in range(n):

elements=int(input())

L.append(elements)

def sum\_elements(list\_sum):

return sum(list\_sum)

total=sum\_elements(L)

print(f"The sum of the list is: {total}")

#Function - return a list

def create\_list(L1,L2,L3,L4):

L=[]

L.extend((L1,L2,L3,L4))

return L

l1=int(input())

l2=int(input())

l3=int(input())

l4=int(input())

list\_=create\_list(l1,l2,l3,l4)

print(list\_)

#Function - List as a parameter

def display\_list(L1):

print("Elements in the list are:")

for i in L1:

print(i)

n=int(input())

L=[]

for i in range(n):

L.append(input())

display\_list(L)

#Function - Dictionary as a parameter

def myFavcosmetics(vDict):

print(vDict)

pLipstick=input()

pKajal=input()

pEyeliner=input()

cosmetics={

'Lipstick':pLipstick,

'Kajal':pKajal,

'Eyeliner':pEyeliner,

}

myFavcosmetics(cosmetics)

#Function - Positional Arguments

def movie\_data(mv\_name,mv\_actor,mv\_actress,s\_actress,mv\_director):

print(f"Movie Name: {mv\_name}")

print(f"Actor: {mv\_actor}")

print(f"Actress: {mv\_actress}")

print(f"Supporting actress: {s\_actress}")

print(f"Director: {mv\_director}")

mv\_name=input()

mv\_actor=input()

mv\_actress=input()

s\_actress=input()

mv\_director=input()

movie\_data(mv\_name,mv\_actor,mv\_actress,s\_actress,mv\_director)

#Function - Keyword arguments

def cosmetic\_display(pName,pBrand,pCost):

print("\*\*\*\*\*\*\*\*\*\*\*\*Product details\*\*\*\*\*\*\*\*\*\*\*\*\*")

print(pName)

print(pBrand)

print(pCost)

pName=input()

pBrand=input()

pCost=input()

cosmetic\_display(pName="Lipstick",pBrand="MyGlamm",pCost=199)

#Recursion - Factorial of a Number

def fact(n):

if n==0:

return 1

else:

return n\*fact(n-1)

num=int(input())

f=fact(num)

print(f"The factorial of {num} is {f}")

#Recursion - Sum of Array Elements

def sum\_array(arr):

if len(arr)==0:

return 0

else:

return arr[0]+sum\_array(arr[1:])

#execution start here...........

n=int(input())

arr=[]

for i in range(n):

arr.append(int(input()))

result=sum\_array(arr) #function call

print(result)