1.Input & Output - Say "Hello, World**!"**

x=str('"Hello, World!"')

print(x)

## 2.Input & Output - Hello World with tab

print("Hello World\tHello World")

## 3.Input & Output - Hello World with a new line

print("Hello World\nHello World")

## 4.Input & Output - Student Details

Name =input()

Age=int(input())

CGPA=float(input())

Grade=input()

c=int(CGPA\*100)/100.0

print("Name:",Name)

print("Age:",Age)

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

print("Grade:",Grade)

## 5 .Input & Output - ASCII Values - I

n=ord(input())

print(n)

## 6 .Input & Output - ASCII Values - II I

n=int(input())

print(chr(n))

7. Input & Output - Round Off

import math

x=float(input())

print(math.floor(x))

print(math.ceil(x))

print(math.trunc(x))

8.Operators - Fencing the Ground

l=int(input())

b=int(input())

perimeter=2\*(l+b)

area=l\*b

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

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

9.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)

10.Operators - Harry Potter

import string

n=input()

n=n.strip()

n=n.strip(string.punctuation)

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

11.Operators - Splitting into the teams

a=int(input())

b=int(input())

c=a//b

d=a%b

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

12.Operators - Debt Repay

p=float(input())

r=float(input())

n=float(input())

interest=(p\*r\*n)/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}")

13.Operators - 3 Psychos

x1=int(input())

y1=int(input())

x2=int(input())

y2=int(input())

a=(x1+x2)/2

b=(y1+y2)/2

print(f"Arun's house is located at({a},{b})")

14.Operators - Hop n Hop

import math

x=int(input())

y=int(input())

n=math.pow((x-3),2)

n1=math.pow((y-4),2)

z=n+n1

w=math.pow(z,1/2)

u=math.trunc(w)

print(u)

15.Operators - Dollars & Cents

dollar1=int(input())

cent1=int(input())

dollar2=int(input())

cent2=int(input())

dollar=dollar1+dollar2

cent=cent1+cent2

if cent>100:

cent-=100

dollar+=1

print(dollar)

print(cent)

16.Operators - Treasure Hunter

gold=int(input())

ben\_share=int(input())

black\_share=int(input())

ben=(gold\*ben\_share)//100

rem\_ben=gold-ben

black=(rem\_ben\*black\_share)//100

rem\_black=rem\_ben-black

pirates=rem\_black//3

print(f"{ben}")

print(f"{black}")

print(f"{pirates}")

17.Operators - Reverse a 3-digit number

n=int(input()) #123

first=n//100 #1

middle=(n//10)%10 #12%10=2

last=n%10 #123%10=3

rev=(last\*100)+(middle\*10)+(first) #300+20+1

print(rev)

18.Operators - Tic Tac Toe

n=int(input())

if 1 <= n <= 9:

n-=1

row=n//3

column=n%3

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

18.Decision Making - Checking Alphabets

a=("a","e","i","o","u","A","E","O","I","U")

b=input()

if b.isalpha()==True:

if b in a:

print("Vowel")

elif b not in a:

print("Consonant")

else:

print("Not an alphabet")

19.Decision Making - Electricity Bill

import math

units=int(input())

if units<=200:

eb\_bill=(math.trunc(0.5\*units))

print(f"Rs.{eb\_bill}")

elif units>=200 and units<=400:

eb\_bill=(math.trunc(0.65\*units+100))

print(f"Rs.{eb\_bill}")

elif units>=400 and units<=600:

eb\_bill=(math.trunc(0.80\*units+200))

print(f" Rs.{eb\_bill}")

else:

eb\_bill=(math.trunc(0.25\*units+425))

print(f"Rs.{eb\_bill}")

20.Decision Making - Online shopping

import math

f\_shirt=int(input())

f\_discount=int(input())

f\_shipping=int(input())

s\_shirt=int(input())

s\_discount=int(input())

s\_shipping=int(input())

a\_shirt=int(input())

a\_discount=int(input())

a\_shipping=int(input())

f\_dismoney=(f\_discount/100)\*f\_shirt

f\_final=math.floor(f\_shirt-f\_dismoney+f\_shipping)

s\_dismoney=(s\_discount/100)\*s\_shirt

s\_final=math.floor(s\_shirt-s\_dismoney+s\_shipping)

a\_dismoney=(a\_discount/100)\*a\_shirt

a\_final=math.floor(a\_shirt-a\_dismoney+a\_shipping)

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

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

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

if f\_final<=s\_final and f\_final<=a\_final:

print("Choose Flipkart")

elif s\_final<=f\_final and s\_final<=a\_final:

print("Choose Snapdeal")

elif a\_final<=s\_final and a\_final<=f\_final:

print("Choose Amazon")

else:

print("Choose Flipkart")

21.Decision Making - Hotel Tariff Calculator

import math

n=int(input())

rent=int(input())

days=int(input())

rent=rent\*days

if n>12:

print("Invalid Input")

elif n==4:

add=(20/100)\*rent

new\_rent=math.trunc(rent+add)

print(new\_rent)

elif n==5:

add=(20/100)\*rent

new\_rent=math.trunc(rent+add)

print(new\_rent)

elif n==6:

add=(20/100)\*rent

new\_rent=math.trunc(rent+add)

print(new\_rent)

elif n==11:

add=(20/100)\*rent

new\_rent=math.trunc(rent+add)

print(new\_rent)

elif n==12:

add=(20/100)\*rent

new\_rent=math.trunc(rent+add)

print(new\_rent)

else:

print(rent)

22.Decision Making - Gift for Birthday

n=int(input())

if n%100==0:

if n%400==0:

print(n, "is a leap year")

else:

print(n, "is not a leap year")

elif n%4==0:

print(n, "is a leap year")

else:

print(n, "is not a leap year")

23.Decision Making - Trendy number

n=int(input())

if 100<=n<=999:

mid\_digit=(n//10)%10

if mid\_digit%3==0:

print("Trendy Number")

else:

print("Not a Trendy Number")

else:

print("Invalid Number")

24. Time Sheet

sun=int(input())

mon=int(input())

tue=int(input())

wed=int(input())

thur=int(input())

fri=int(input())

sat=int(input())

week=mon+tue+wed+thur+fri

sal=week\*100

if mon>8:

sal+=(mon-8)\*15

if tue>8:

sal+=(tue-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\_bon=int(sun\*0.50\*100)

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

print(sal)

25.Decision Making - Number of Days

year=int(input())

month=int(input())

if 1900<=year<=9999:

if 1<=month<=12:

if month==1:

days=31

print(days,"Days")

if month==2:

if year%100==0:

if year%400==0:

days=29

print(days,"Days")

else:

days=28

print(days,"Days")

elif year%4==0:

days=29

print(days,"Days")

else:

days=28

print(days,"Days")

elif month==3:

days=31

print(days,"Days")

elif month==5:

days=31

print(days,"Days")

elif month==7:

days=31

print(days,"Days")

elif month==8:

days=31

print(days,"Days")

elif month==10:

days=31

print(days,"Days")

elif month==12:

days=31

print(days,"Days")

else:

days=30

print(days,"Days")

else:

print("0")

26.Decision Making - Scholarship

age=int(input())

year\_passing=int(input())

f\_income=int(input())

arrears=int(input())

percentage=float(input())

attendance=float(input())

if year\_passing>=2021:

if arrears<=2:

if f\_income<=200000:

if 18<=age<=21:

if percentage>=60:

if attendance>=80:

print("Eligible")

elif arrears>2 and percentage>=80 and attendance>=90 and 200000<=f\_income<250000:

print("Partially Eligible")

else:

print("Not Eligible")

**26.Decision Making - Mango tree**

row=int(input())

cols=int(input())

tree=int(input())

if tree<=cols: #1st row

print("Yes")

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

print("Yes")

elif tree%cols==0: #last column

print("Yes")

else:

print("No")

27.Decision Making – Cricket

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")

28.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}")

29.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=" ")

30 .Control Statements - Special number

m=int(input())

n=int(input())

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

first=i//10

second=i%10

sum\_digits=first+second

product=first\*second

total\_sum=sum\_digits+product

if total\_sum==i:

print(i)

31.Control Statements - Amoeba Multiplication

month\_count=int(input())

n1=0

n2=1

count=2

if month\_count<=0:

print("")

else:

while count<month\_count:

n3=n1+n2

n1=n2

n2=n3

count+=1

print(n3)

32.Control Statements - Number Series

n=int(input())

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

if i%2==0:

num=pow(i,2)-2

print(num,end=" ")

else:

num=pow(i,2)-1

print(num,end=" ")

33.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()

34.Control Statements - Treasure Finder

import math

a=int(input())

b=int(input())

c=int(input())

if a>b and a<c or a<b and a>c:

second=a

elif b>a and b<c or b<a and b>c:

second=b

else:

second=c

print("The treasure is in the box which has the number",second)

hcf=math.gcd(math.gcd(a,b),c)

print("The code to open the box is",abs(hcf))

35.Control Statements - Collatz problem

import math

n=int(input())

count=0

print(n)

while n>1:

if n%2==0:

n=n/2

print(math.trunc(n))

count+=1

else:

n=(3\*n)+1

print(math.trunc(n))

count+=1

print(count)

36.Control Statements - Strong Number

import math

sum\_fact=0

n=int(input())

num=n

while n>0:

last=n%10

sum\_fact+=math.factorial(last)

n//=10

if sum\_fact==num:

print("Yes")

else:

print("No")

37.Control Statements - Inverted right-angled triangle

start=int(input())

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

for j in range (i):

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

print()

38.Control Statements - Sum of digit till single digit

n=int(input())

s=0

while n>0:

s=int(s)

last=n%10

s+=last

n=n//10

s=str(s)

a=len(s)

if a>1:

s=int(s)

last=s%10

sec=s//10

s=sec+last

print(s)

39.Control Statements - Kaprekar number

import math

n=int(input())

sq=n\*\*2

s=str(n)

l=len(s)

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

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

s=left+right

if s==n:

print("Kaprekar Number")

else:

print("Not a Kaprekar Number")

40.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 #21-(4-1)\*2-1

41.Arrays 1D - Same or Not

size1 = int(input())

size2 = int(input())

if size1 != size2:

print("Not Same")

else:

array1 = []

for \_ in range(size1):

array1.append(int(input()))

array2 = []

for \_ in range(size2):

array2.append(int(input()))

sum1 = sum(array1)

sum2 = sum(array2)

if sum1 == sum2:

print("Same")

else:

print("Not Same")

42.Arrays 1D - Count distinct elements

x=int(input())

lst=[]

for i in range(0,x):

y=int(input())

lst.append(y)

uniquelist=[]

dulpicatelist=[]

for i in lst:

if i not in uniquelist:

uniquelist.append(i)

elif i not in dulpicatelist:

dulpicatelist.append(i)

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

43.Arrays 1D - Compatible array

n1 = int(input())

array1 = []

for i in range(n1):

array1.append(int(input()))

n2 = int(input())

array2 = []

for j in range(n2):

array2.append(int(input()))

if n1 == n2 and all(x >= y for x, y in zip(array1, array2)):

print("Compatible")

else:

print("Incompatible")

44.Arrays 1D - Sum of even numbers and odd numbers

import array as arr

n=int(input())

array=[]

for i in range(n):

elements=int(input())

if i<=15:

array.append(elements)

sum\_even=0

sum\_odd=0

for i in array:

if i%2==0:

sum\_even+=i

else:

sum\_odd+=i

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

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

45.Arrays 1D - Ascending order

n = int(input())

array1=[]

for i in range(n):

array1.append(int(input()))

array1.sort()

print("The Sorted array is:")

for element in array1:

print(element)

46.Arrays 1D – Queue

n,m=map(int,input().split())

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

buses=0

capacity=0

for g in groups:

if capacity+g<=m:

capacity+=g

else:

buses+=1

capacity=g

if capacity>0:

buses+=1

print(buses)

47.Arrays 1D - Array insertion

import array as arr

size=int(input())

lst=[]

for i in range(size):

lst.append(int(input()))

index=int(input())

if size>=index:

num=int(input())

a=arr.array('i',lst)

a.insert(index-1,num)

print(f"Array after insertion is")

for i in a:

print(i)

else:

print("Invalid Input")

48.Arrays 1D - Remove duplicate elements

n=int(input())

lst=[]

unique=[]

for i in range(n):

lst.append(int(input()))

for i in lst:

if i not in unique:

unique.append(i)

for i in unique:

print(i)

49.Arrays 1D - Online Game

n=int(input())

a=list(map(int,input().split())) #73 4 63 23 65

left=0

right=n-1

while left<right: #1<0--->failed

while a[left]%2==0: #a[0]%2==0--X | a[1]%2==0

left+=1 #1+1=2

while a[right]%2!=0: #a[4]%2!=0: |a[3]%2!=0: | a[2]%2!=0:| a[1]%2!=0:

right-=1 #4-1=3| 3-1=2| 2-1=1 |

if left<right: #0<1

t=a[left]

a[left]=a[right]

a[right]=t

left+=1 #1

right-=1 #0

print("Array after Segregation")

for i in range(len(a)):

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

50.Arrays 1D – Toyland

num = int(input())

houses = []

for \_ in range(num):

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

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

houses.sort()

max\_gap = 0

house\_pair = (0, 0)

for i in range(1, num):

pos1, house1 = houses[i - 1]

pos2, house2 = houses[i]

gap = pos2 - pos1

if gap > max\_gap:

max\_gap = gap

house\_pair = (house1, house2)

house\_pair = tuple(sorted(house\_pair))

print(house\_pair[0], house\_pair[1])

51.Array 1D - Pair the Container

n=int(input())

left=0

right=n-1

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

for i in range(n):

a.sort()

for i in a:

while left<right:

t=a[left]

s=a[right]

left+=1

right-=1

print(f"{s} {t}")

if left==right:

print(f"{a[left]} 0")

**52.Arrays 1D - Smallest Positive Missing Number**

n=int(input())

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

m=1

postive=[]

for i in range(n):

if m<a[i]:

postive.append(a[i])

elif m==a[i]:

m+=1

while(postive.count(m)):

postive.remove(m)

m+=1

print(m)

53.Arrays 2D - Transpose Matrix

row\_col=int(input())

arr=[]

for i in range(row\_col):

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

arr.append(L)

for i in range(row\_col):

for j in range(row\_col):

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

print()

print("Transpose matrix is:")

for i in range(row\_col):

for j in range(row\_col):

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

print()

54.Arrays 2D - Upper triangular matrix

row\_col=int(input())

arr=[]

for i in range(row\_col):

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

arr.append(L)

upper=True

for i in range(row\_col):

for j in range(i):

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

upper=False

if upper==True:

print("Upper triangular matrix")

else:

print("Not an Upper triangular matrix")

## 55.Arrays 2D - Maximum element in each column

rows = int(input())

column = int(input())

matrix = []

for i in range(rows):

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

matrix.append(row)

for i in range(column): #i-->column

maximum=matrix[0][i]

for j in range(1,rows): #j-->row

if matrix[j][i]>maximum:

maximum=matrix[j][i]

print(maximum)

56.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)))

57.Arrays 2D - Sum of Zig-Zag

arr=[]

add=0

row=int(input())

coloumn=int(input())

for i in range(row):

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

arr.append(elements)

for i in range(row):

for j in range(coloumn):

if i==0:

add+=arr[i][j]

if i==row-1:

add+=arr[i][j]

if i==j==row//(row-1) :

add+=arr[i][j]

print(f"Sum of Zig-Zag pattern is {add}")

58.Arrays 2D - Move all zeroes

Test = int(input())

for \_ in range(Test):

binary\_str = input().strip()

one = []

zero = []

for i in binary\_str:

if i == '1':

one.append(i)

else:

zero.append(i)

result = ''.join(one + zero)

print(result)

59.Array 2D - Uniformity Matrix

n=int(input())

matrix=[]

o=True #odd

e=True #even

for i in range(n):

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

matrix.append(l)

for i in range(n):

for j in range(n):

if matrix[i][j]%2==1:

e=False

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

o=False

if o or e:

print("Yes")

else:

print("No")

60.Array 2D - Magic Square

n=int(input())

arr=[]

final\_total=0 #main diagonal

total=0 #anti diagonal

for i in range(n):

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

arr.append(l)

for i in range(n):

for j in range(n):

if i==j:

final\_total+=arr[i][j]

for i in range(n):

total+=arr[i][n-i-1]

if final\_total==total: #main diagonal==anti diagonal that is square matrix

print("Yes")

else:

print("No")

61.Arrays 2D - Sum of rows and columns

r=int(input()) #row

c=int(input()) #column

a=[]

for i in range(r):

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

a.append(l)

row=[]

col=[0]\*c

for i in range(r):

row\_sum=sum(a[i]) #first row sum

row.append(row\_sum) #sum is appended to the row list

for j in range(c): #columns

col[j]+=a[i][j] #col0]+=a[0][0],col[1]=a[0][1],col[2]+=a[0][2]

print("The Sum of rows is",end=" ")

for i in row:

print(i,end=" ")

max\_row=row.index(max(row))+1

print(f"\nRow {max\_row} has a maximum sum")

print("The Sum of columns is",end=" ")

for i in col:

print(i, end=" ")

max\_col=col.index(max(col))+1

print(f"\nColumn {max\_col} has the maximum sum")

62.Arrays 2D - Spiral Pattern

n=int(input())

matrix=[]

for i in range(n):

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

matrix.append(L)

top,bottom,left,right=0,n-1,0,n-1

result=[]

while top<=bottom and left<=right:

for col in range(left,right+1):

result.append(matrix[top][col])

top+=1

for row in range(top,bottom+1):

result.append(matrix[row][right])

right-=1

if top<=bottom:

for col in range(right,left-1,-1):

result.append(matrix[bottom][col])

bottom-=1

if left<=right:

for row in range(bottom,top-1,-1):

result.append(matrix[row][left])

left+=1

print(" ".join(map(str,result)))

63.Arrays 2D - Matrix Rotation

n=int(input())

arr=[]

for i in range(n):

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

arr.append(L)

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

64.Function- Sum of list

n=int(input())

l=[]

for i in range(n):

l.append(int(input()))

def sum\_of\_list(list\_sum):

return sum(list\_sum)

total=sum\_of\_list(l)

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

65.Function - return a list

n1=int(input())

n2=int(input())

n3=int(input())

n4=int(input())

def create\_list():

l=[]

l.extend((n1,n2,n3,n4))

print(l)

create\_list()

66.Function - List as a parameter

n=int(input())

l=[]

for i in range(n):

e=str(input())

l.append(e)

print(f"Elements in the list are:")

def list\_elements():

for i in l:

print(i)

list\_elements()

67.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)

68.Function - Positional Arguments

e1=input()

e2=input()

e3=input()

e4=input()

e5=input()

def elements(e1,e2,e3,e4,e5):

print("Movie Name:",e1)

print("Actor:",e2)

print("Actress:",e3)

print("Supporting actress:",e4)

print("Director:",e5)

elements(e1,e2,e3,e4,e5)

69.Function - Keyword arguments

def display(product\_name,brand\_name,cost):

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

print(product\_name)

print(brand\_name)

print(cost)

display("Lipstick","MyGlamm",199)

70.Function - Password generator

import random

import string

upper=string.ascii\_uppercase #it generates all uppercase

lower=string.ascii\_lowercase #it generates all lowercase

digits=string.digits # it generates all digits

symbols=string.punctuation #it genreates all punctutation

#password generator function

def password\_generator(size):

chars=upper+lower+digits+symbols

password=''

for i in range(size):

password+=random.choice(chars)

return password

#execution starts here...

dig=int(input())

print(password\_generator(dig)) #function call

71.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) #function call

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

72.Recursion - Fibonacci Series

def fibonacci(n):

if n==0:

return 0

elif n==1:

return 1

else:

return fibonacci(n-1)+fibonacci(n-2)

num=int(input())

f=fibonacci(num-1)

print(f"The term {num} in the Fibonacci series is {f}")

73.Recursion - Compute a^n

def cal\_power(x,y):

if y==0:

return 1

else:

return x\*cal\_power(x,y-1)

a=int(input())

n=int(input())

result=cal\_power(a,n)

print(f"The value of {a} power {n} is {result}")

74.Recursion - Number of digits

def count\_num(n): #435

if n<10:

return 1 #return 1 when n becomes a single digit number

else:

return 1+count\_num(n//10)

#execution starts here...

num=int(input()) #435

dig=count\_num(num) #function call

print(f"The number of digits in {num} is {dig}")

75.Recursion - Sum of Positive Odd Numbers

def cal\_sum(o\_arr):

if len(o\_arr)==0:

return 0

else:

return o\_arr[0]+cal\_sum(o\_arr[1:])

#exectuion start here..

n=int(input()) #no of elements in array

arr=[] #empty array

for i in range(n):

arr.append(int(input())) #append n eleements in array

odd\_arr=[] #odd array empty array

for i in arr:

if i%2!=0 and i>0:

odd\_arr.append(i)

s=cal\_sum(odd\_arr) #function call

print(f"Sum = {s}")

76.Recursion - Maximum Element in an Array

def find\_max(arr,len):

if len==1:

return arr[0]

else:

max\_in\_rest=find\_max(arr,len-1)

return max(arr[len-1],max\_in\_rest)

#execution starts here..

n=int(input())

arr=[]

for i in range(n):

arr.append(int(input()))

l=len(arr)

print("Maximum element in the array is",find\_max(arr,l))

77.Recursion - Prime Number

def check\_prime(n,i=2):

if n==i:

return True

elif n%i==0:

return False

return check\_prime(n,i+1)

num=int(input())

if check\_prime(num):

print("Prime Number")

else:

print("Not a Prime Number")

78.Recursion - GCD of 2 numbers

def gcd(a,b): #a=4 b=6

if b==0: #(2,0-->b==0)

return a #(2)

else:

return gcd(b,a%b) # (6,4)(4,2)(2,0)

n1=int(input())

n2=int(input())

GCD=gcd(n1,n2) #fucntion call --->GCD=2

print(GCD)

79.Strings - count the vowels

string=input()

l=['a','e','i','o','u','A','E','I','O','U']

count=0

for i in string:

if i in l:

count+=1

print(f"Number of vowels: {count}")

80.Strings - Palindrome

string=input()

rev=''

for i in string:

rev=string[::-1]

if string==rev:

print("Palindrome")

else:

print("Not a Palindrome")

81.Recursion - Decimal To Binary

def getBinaryNum(dec\_num):

if dec\_num==0:

return 0

else:

return(dec\_num%2+10\*getBinaryNum(int(dec\_num//2)))

d\_num=int(input())

print(getBinaryNum(d\_num))

82.Strings – Sorting

string=input()#hello

list\_string=list(string)#['h,'e,'l,'l,'o']

length=len(list\_string)#5

for i in range(length):#5

for j in range(0,length-i-1):#(0,4)

if list\_string[j]>list\_string[j+1]:#0>1#1>2#2>3#3>4#it will check greater vlues by asscii value

list\_string[j],list\_string[j+1]=list\_string[j+1],list\_string[j]

sorted\_string=''

for i in list\_string:#e,h,l,l,o

sorted\_string+=i

print(str(sorted\_string))

83.Strings - Remove all characters in Second String that are present in First String

string1=input()

string2=input()

string1=list(string1)

string2=list(string2)

for i in string1:

if i in string2:

string2.remove(i)

print(''.join(string2))

84.Function - Password generator

import random

import string

upper=string.ascii\_uppercase #it generates all uppercase

lower=string.ascii\_lowercase #it generates all lowercase

digits=string.digits # it generates all digits

symbols=string.punctuation #it genreates all punctutation

#password generator function

def password\_generator(size):

chars=upper+lower+digits+symbols

password=''

for i in range(size):

password+=random.choice(chars)

return password

#execution starts here...

dig=int(input())

print(password\_generator(dig)) #function call

85.Strings - Encryption

n=input()

l=[]

for i in n:

l.append(i)

for i in range(0,len(l)-1,2):

l[i],l[i+1]=l[i+1],l[i]

print(''.join(l))

86.Random number

import random

start=int(input())

end=int(input())

print(random.randrange(start,end))

87.Strings - Wordakshari

words=[]

while True:

w=input().strip()

if w=="####":

break

words.append(w)

chain=[words[0]]

used\_words={words[0]}

current\_words=words[0]

while True:

next\_words=False

for w in words:

if w not in used\_words and w[0].lower()==current\_words[-1].lower():

chain.append(w)

used\_words.add(w)

current\_words=w

next\_words=True

break

if not next\_words:

break

for word in chain:

print(word)

Destructor 1

class Destructor:

def \_\_init\_\_(self):

print("Constructor called: Arya is created.")

def \_\_del\_\_(self):

print("Destructor called: Arya is destroyed.")

print("End of program.")

s=Destructor()

del s

parameterized constructor 1

class Student:

def \_\_init\_\_(self,name, age, sid):

self.name=name

self.age=age

self.sid=sid

def display(self):

print(f"Name: {self.name}\nAge: {self.age}\nStudent ID: {self.sid}")

s1=Student("Ria", "20", "S12345")

s2=Student("Raja", "22", "S67890")

print("Student 1:")

s1.display()

print()

print("Student 2:")

s2.display()

Non-parametrized constructor

class Student:

def \_\_init\_\_(self):

self.name="Default Name"

self.age="18"

self.sid="00000"

def display (self):

print("Non-parameterized constructor called!")

print()

print("Student 1:")

print(f"Name: {self.name}\nAge: {self.age}\nStudent ID: {self.sid}")

s=Student()

s.display()

Default constructor 2

class Student:

def \_\_init\_\_(self, name="Unknown", age=0,sid="N/A"):

self.name=name

self.age=age

self.sid=sid

def display (self):

print("Student 1:")

print(f"Name: {self.name}\nAge: {self.age}\nStudent ID: {self.sid}")

s=Student()

s.display()