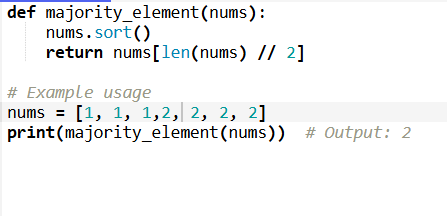
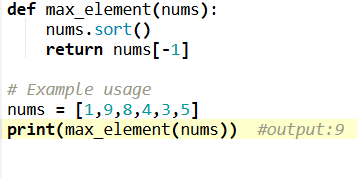
**#Array Questions:**

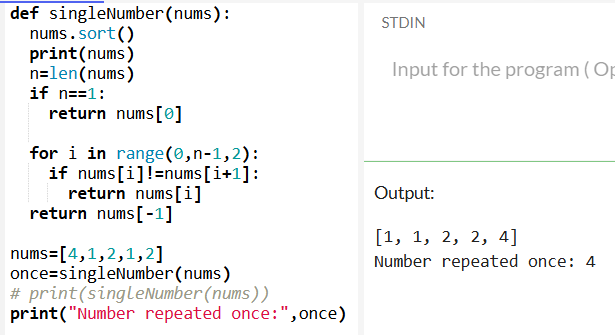
1. Find out the most repeated number such that it is more than num/2



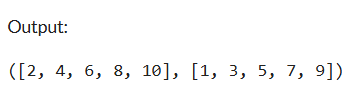
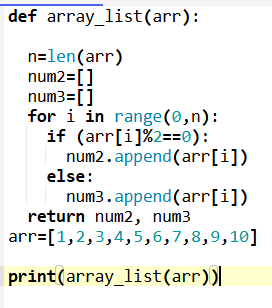
1. Find out the maximum number in the array



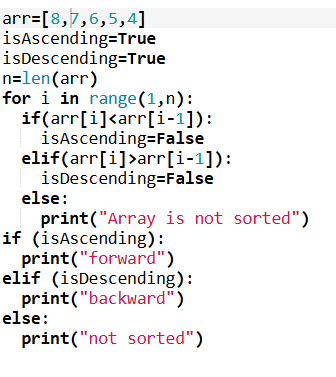
1. To find the number which is repeated only once while all other numbers are repeated twice

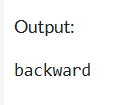


1. To print even and odd numbers in different arrays

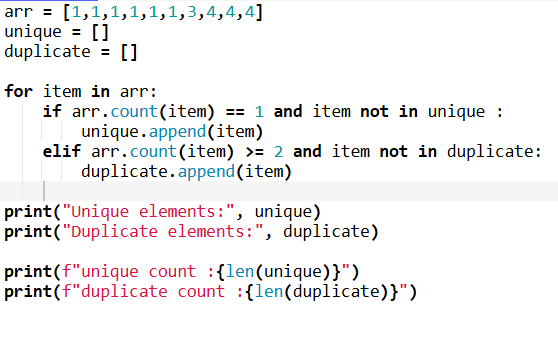


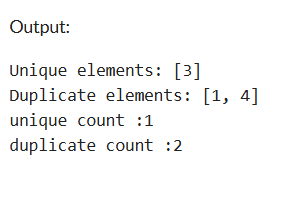
1. To find if the array is in ascending order or descending order



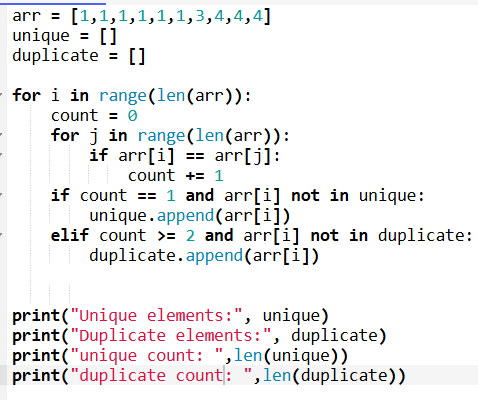


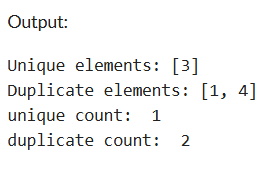
1. To count the unique and duplicate elements in an array:



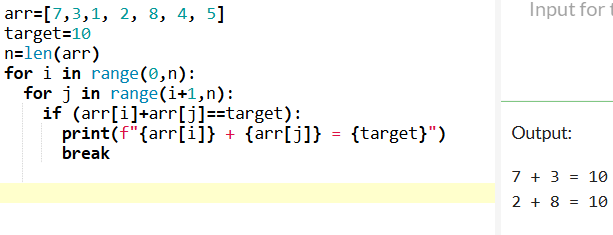


Or without count function

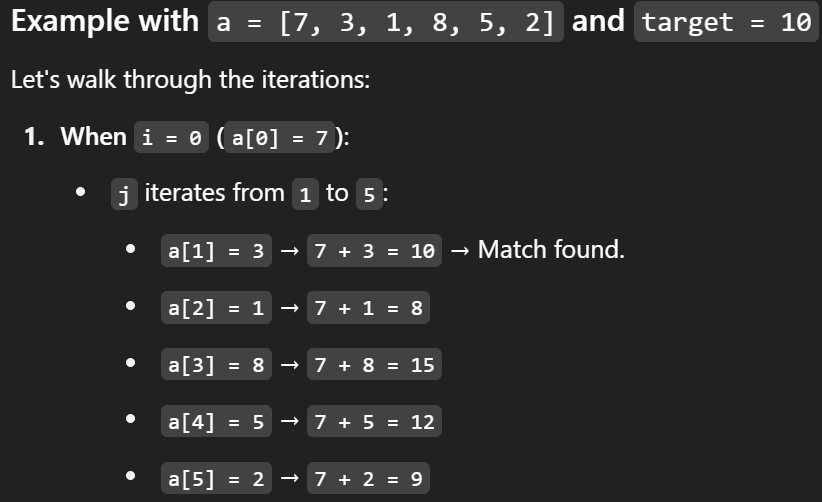




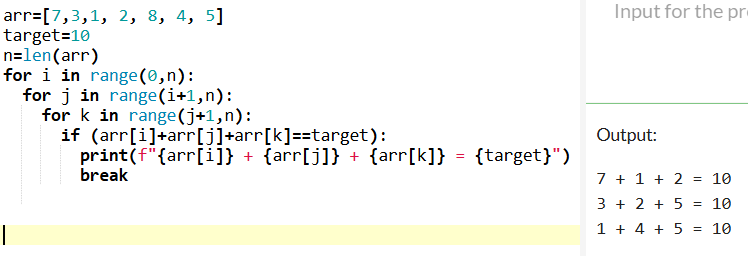
1. Check if two elements exist with a sum equal to a target value.



**Explanation:**



1. Check if two elements exist with a sum equal to a target value



**# 2-D Matrix**

1. Print a matrix row-wise and column-wise:

Code:

arr=[[1,2,3],

[4,5,6]]

print("row wise")

for row in arr:

for num in row:

print(num,end=" ")

print()

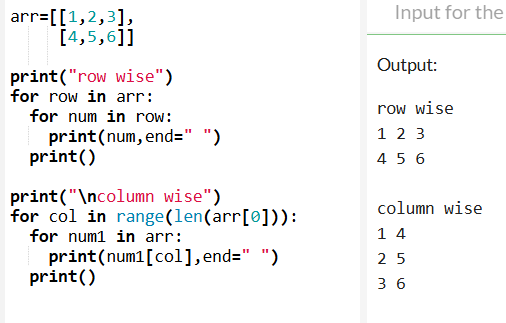
print("\ncolumn wise")

for col in range(len(arr[0])):

for num1 in arr:

print(num1[col],end=" ")

print()



1. Calculate the total sum of elements in a matrix

Code:

arr=[[1,2,3],

[4,5,6]]

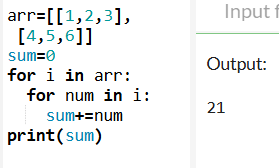
sum=0

for i in arr:

for num in i:

sum+=num

print(sum)



1. Calculate the sum of each row and each column in a matrix

Code:

arr = [

[1, 2, 3],

[4, 5, 6]

]

print("Each row sum")

for i in range(len(arr)):

sum=0

for j in range(len(arr[0])):

sum+=arr[i][j]

print(f"sum of row {i} = ",sum)

print("\nEach column sum")

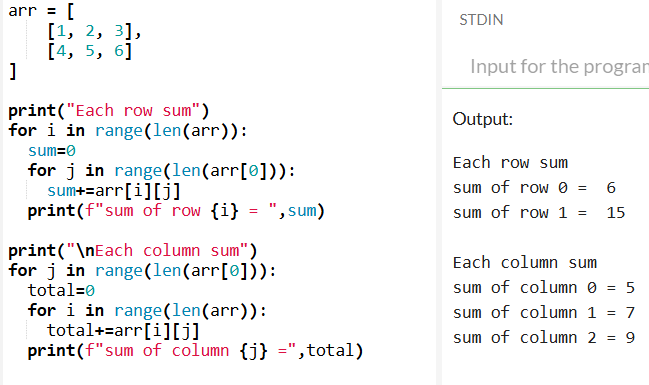
for j in range(len(arr[0])):

total=0

for i in range(len(arr)):

total+=arr[i][j]

print(f"sum of column {j} =",total)



1. Calculate the maximum and minimum element in each row of a matrix

Code:

a = [

[-1,-5,-8],

[-3,-7,-2] ]

for i in range(len(a)):

    max\_value=a[i][0]

    min\_value=a[i][0]

    for j in range(len(a[0])):

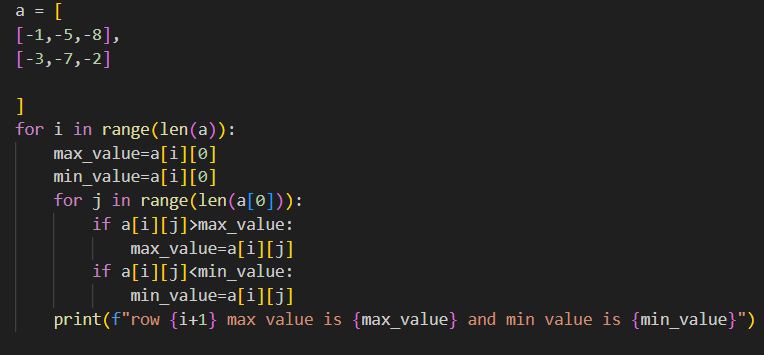
        if a[i][j]>max\_value:

            max\_value=a[i][j]

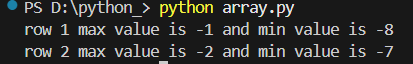
        if a[i][j]<min\_value:

            min\_value=a[i][j]

    print(f"row {i+1} max value is {max\_value} and min value is {min\_value}")



Output:



1. Calculate the maximum and minimum element in each column of a matrix

code:

arr = [ [1, 8, 3],

    [4, 5, 6] ]

for j in range(len(arr[0])):

    max\_value=arr[0][j]

    min\_value=arr[0][j]

    for i in range(len(arr)):

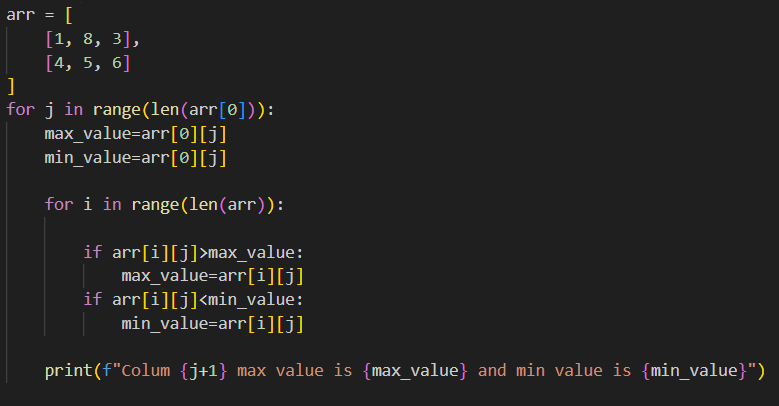
        if arr[i][j]>max\_value:

            max\_value=arr[i][j]

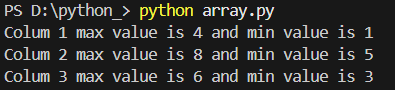
        if arr[i][j]<min\_value:

            min\_value=arr[i][j]

    print(f"Column {j+1} max value is {max\_value} and min value is {min\_value}")



Output:



1. Addition and Substraction of two matrix

Code:

arr=[[2 ,4 ,9],

     [16, 7, 10]]

b=[[1 ,2 ,3],

     [4, 5, 6]]

a=[]

s=[]

for i in range(len(arr)):

    sum\_row = []

    diff\_row = []

    for j in range(len(arr[0])):

        total=arr[i][j]+b[i][j]

        sub=arr[i][j]-b[i][j]

        sum\_row.append(total)

        diff\_row.append(sub)

    a.append(sum\_row)

    s.append(diff\_row)

print("Sum:")

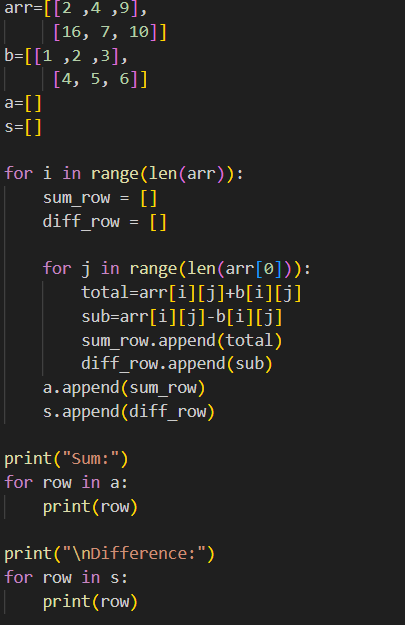
for row in a:

    print(row)

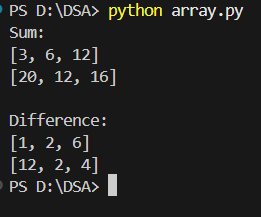
print("\nDifference:")

for row in s:

    print(row)



Output:



1. Print the upper triangle and lower triangle of a matrix.

Code:

arr=[[2 ,4 ,9],

[9, 7, 1],

[1, 6 ,8]]

upper=[]

lower=[]

for i in range(len(arr)):

upper\_matrix=[]

lower\_matrix=[]

for j in range(len(arr[0])):

if (i>j):

upper\_matrix.append(0)

else:

upper\_matrix.append(arr[i][j])

if (i<j):

lower\_matrix.append(0)

else:

lower\_matrix.append(arr[i][j])

upper.append(upper\_matrix)

lower.append(lower\_matrix)

print("Upper Triangle")

for i in upper:

for n in i:

print(n,end=" ")

print()

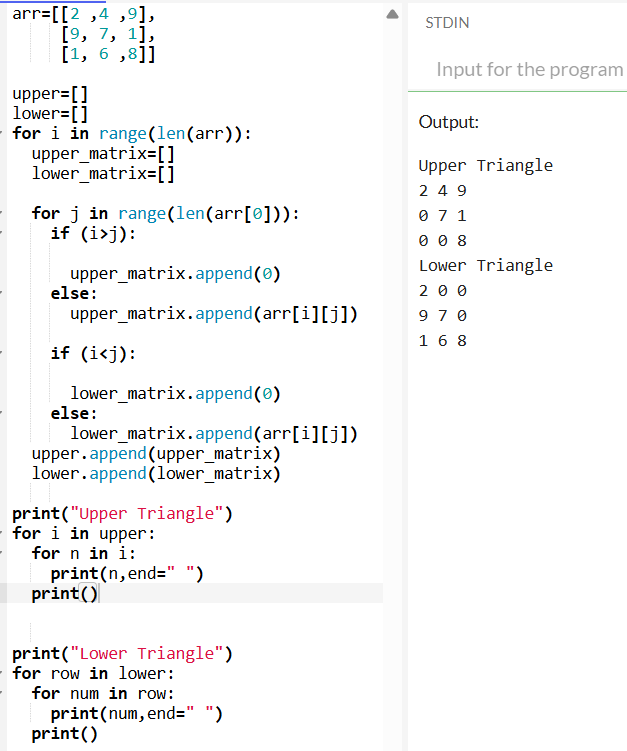
print("Lower Triangle")

for row in lower:

for num in row:

print(num,end=" ")

print()



1. Print the left and right diagonals of a matrix

Code:

arr=[[2 ,4 ,9],

[9, 7, 1],

[1, 6 ,8]]

left=[]

right=[]

for i in range(len(arr)):

left\_matrix=[]

right\_matrix=[]

for j in range(len(arr[0])):

if (i==j) :

left\_matrix.append(arr[i][j])

else:

left\_matrix.append(0)

if i + j == len(arr) - 1:

right\_matrix.append(arr[i][j])

else:

right\_matrix.append(0)

left.append(left\_matrix)

right.append(right\_matrix)

print("Left Diagonal :")

for row in left:

for num in row:

print(num,end=" ")

print()

print("Right Diagonal :")

for row in right:

for num in row:

print(num,end=" ")

print()



1. Print the boundary elements of a matrix

Code:

arr=[[2 ,4 ,9, 8],

[9, 7, 1, 9],

[1, 6 ,8, 5],

[2, 3, 5, 6]]

bound=[]

for i in range(len(arr)):

boundary=[]

for j in range(len(arr[0])):

if (i==0):

boundary.append(arr[i][j])

elif (i==len(arr)-1):

boundary.append(arr[i][j])

elif (j==0):

boundary.append(arr[i][j])

elif (j==len(arr[0])-1):

boundary.append(arr[i][j])

else:

boundary.append(0)

bound.append(boundary)

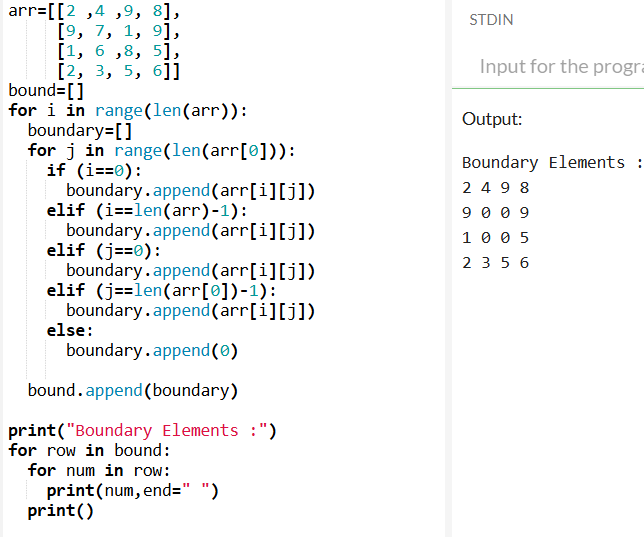
print("Boundary Elements :")

for row in bound:

for num in row:

print(num,end=" ")

print()



1. Sort the matrix row-wise and column-wise

Code:

arr = [

[12, 4, 9],

[9, 7, 1],

[1, 9, 8]

]

row=[]

col=[]

for i in range(len(arr)):

row\_wise=[]

for j in range(len(arr[0])):

row\_wise.append(arr[i][j])

row\_wise.sort()

row.append(row\_wise)

for j in range(len(arr[0])):

col\_wise=[]

for i in range(len(arr)):

col\_wise.append(arr[i][j])

col\_wise.sort()

col.append(col\_wise)

print("Row-wise Sorting")

for i in row:

for num in i:

print(num, end=" ")

print()

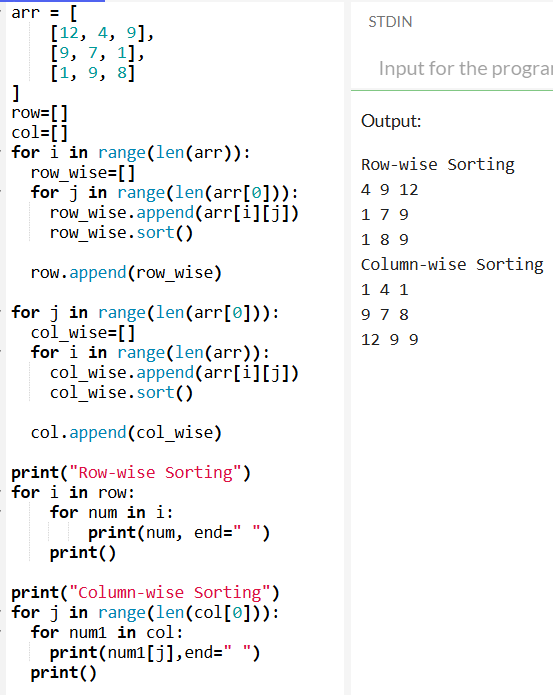
print("Column-wise Sorting")

for j in range(len(col[0])):

for num1 in col:

print(num1[j],end=" ")

print()



1. Print the matrix in a zig zag pattern

Code:

arr = [

[12, 4, 9],

[5, 7, 1],

[1, 9, 8]

]

zig=[]

for i in range(len(arr)):

if (i%2==0):

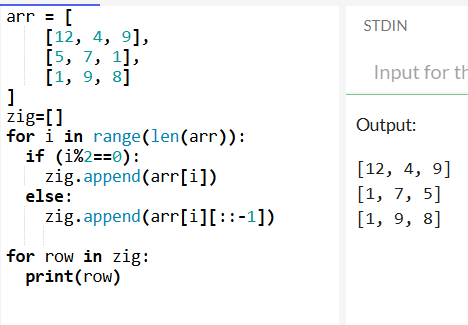
zig.append(arr[i])

else:

zig.append(arr[i][::-1])

for row in zig:

print(row)



1. Check if a matrix is symmetric

Code:

def symmetric(arr):

for i in range(len(arr)):

for j in range(len(arr[0])):

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

return "Matrix is not symmetric"

return "Matrix is symmetric"

arr=[

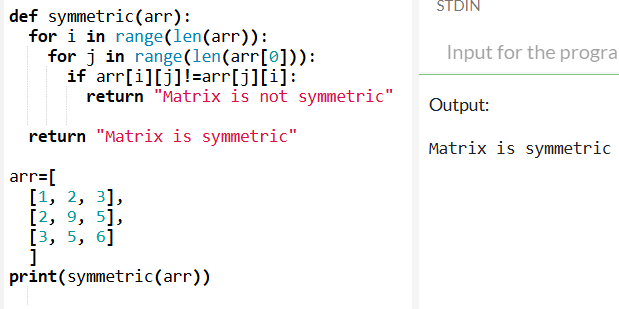
[1, 2, 3],

[2, 9, 5],

[3, 5, 6]

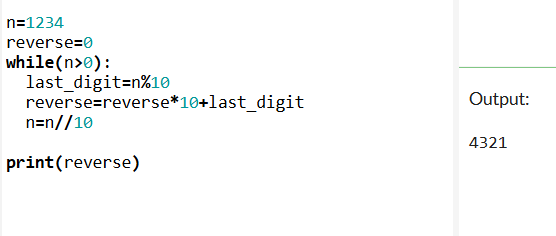
]

print(symmetric(arr))

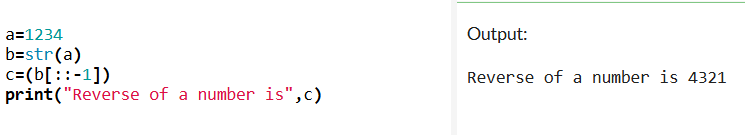


**# Maths Basic Questions**

1. Reverse:



#or



1. Palindrome:

Code:

a=1001

b=a

reverse=0

while(a>0):

last\_digit=a%10

reverse=reverse\*10+last\_digit

a=a//10

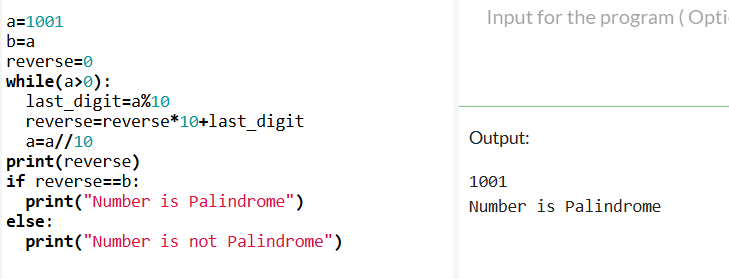
print(reverse)

if reverse==b:

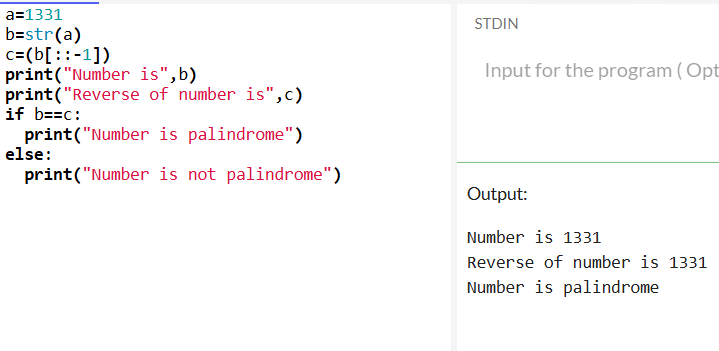
print("Number is Palindrome")

else:

print("Number is not Palindrome")



#or by converting int into str



1. Armstrong Number:

Code:

#Armstrong Number

a=153

b=a

sum\_digit=0

while(b>0):

rem=b%10

sum\_digit=sum\_digit+(rem\*rem\*rem)

b=b//10

if sum\_digit==a:

print(f"{a} is an Armstrong number.")

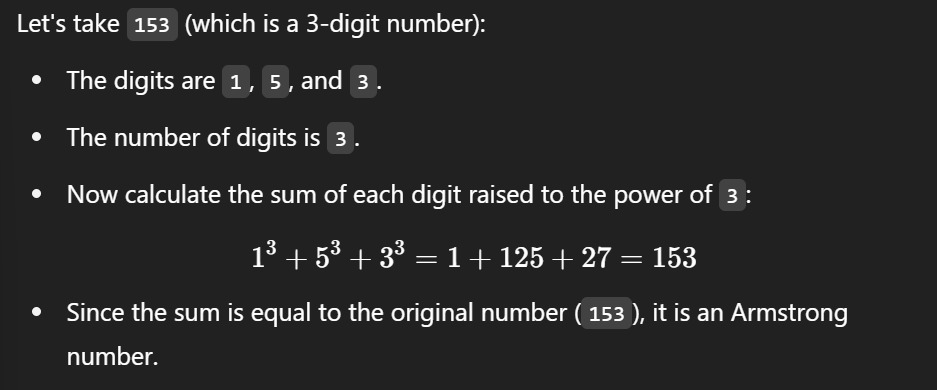
else:

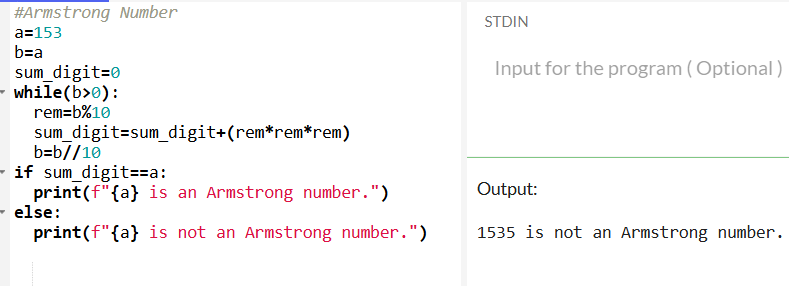
print(f"{a} is not an Armstrong number.")

**explanation:**

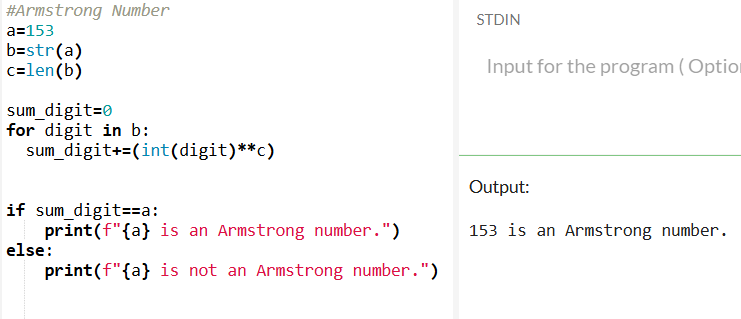
**First iteration (b = 153)**:

* rem = 153 % 10 = 3 (last digit is 3)
* sum\_digit = 0 + (3 \* 3 \* 3) = 0 + 27 = 27
* b = 153 // 10 = 15 (last digit removed)





#or



1. Count the total occurrences of the digit ‘1’ in all positive integers less than or equal to a given integer n.

Code:

a=15

count=0

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

b=i

while(b>0):

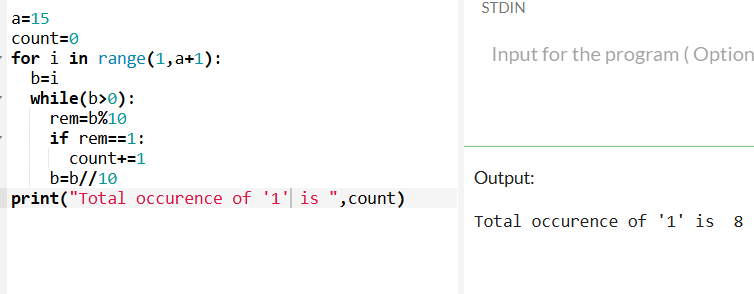
rem=b%10

if rem==1:

count+=1

b=b//10

print("Total occurence of '1' is ",count)



1. Generate Fibonacci numbers up to a given limit

Code:

a=5

sum\_digit=0

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

b=i

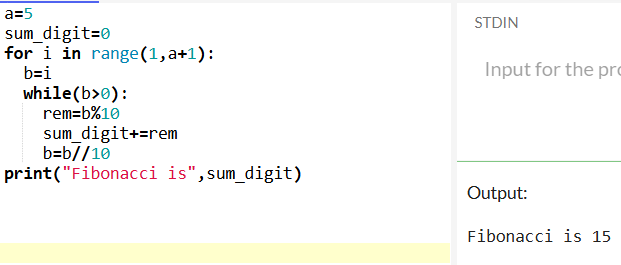
while(b>0):

rem=b%10

sum\_digit+=rem

b=b//10

print("Fibonacci is",sum\_digit)



1. Calculate the factorial of a number

Code:

a=7

sum\_digit=1

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

b=i

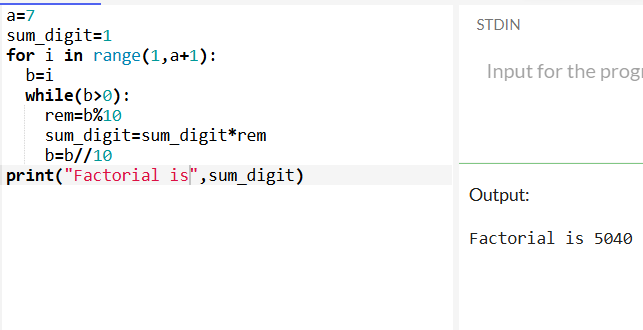
while(b>0):

rem=b%10

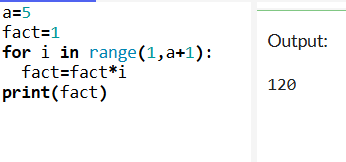
sum\_digit=sum\_digit\*rem

b=b//10

print("Factorial is",sum\_digit)



#or



1. Find the number of trailing zeroes in a factorial of a number

Code:

n=24

fact=1

count=0

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

fact=fact\*i

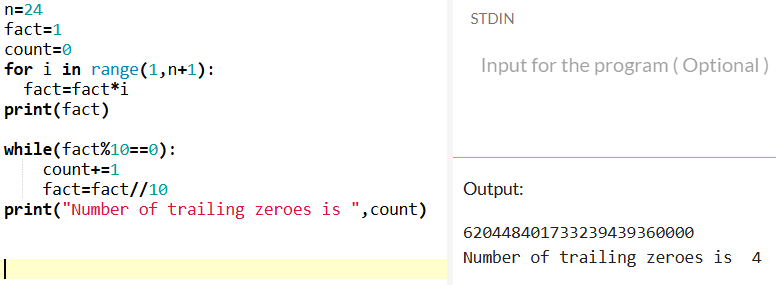
print(fact)

while(fact%10==0):

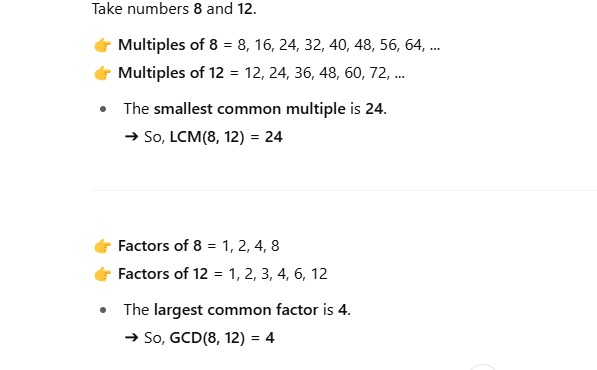
count+=1

fact=fact//10

print("Number of trailing zeroes is ",count)



1. Calculate the LCM and GCD of two numbers



Code:

a=12

b=8

gcd=1

for i in range(1,min(a,b)+1):

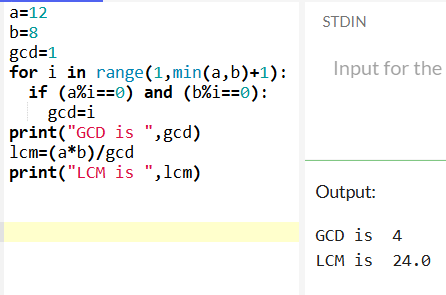
if (a%i==0) and (b%i==0):

gcd=i

print("GCD is ",gcd)

lcm=(a\*b)/gcd

print("LCM is ",lcm)



#or lcm without the formula:

a=12

b=8

gcd=1

for i in range(1,min(a,b)+1):

if (a%i==0) and (b%i==0):

gcd=i

print("GCD is ",gcd)

lcm=max(a,b)

while True:

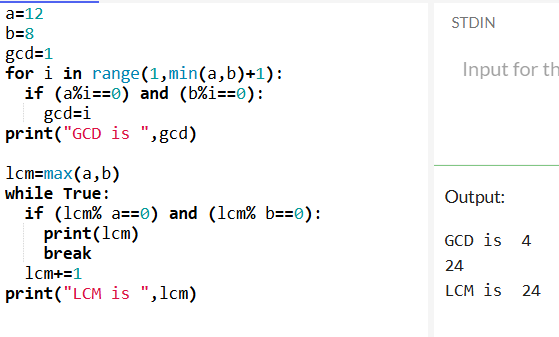
if (lcm% a==0) and (lcm% b==0):

print(lcm)

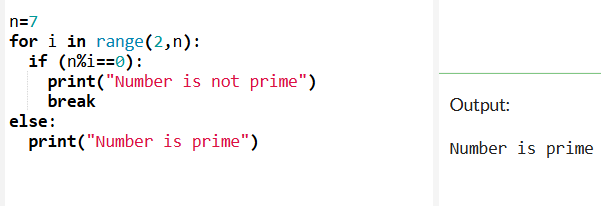
break

lcm+=1

print("LCM is ",lcm)

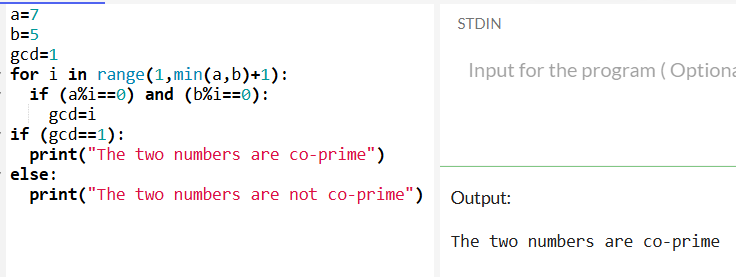


1. Check if the number is prime



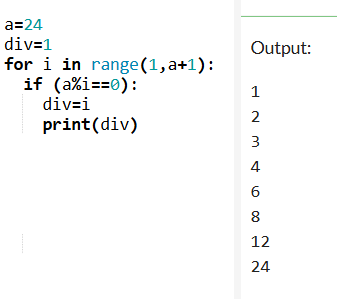
1. Check if the two numbers are co-prime

Code:



1. Find all divisors of a given number

Code:



1. Find out the elements less and more than the half size of list

