**PYTHON PROGRAMMING**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No.** | **Description** | **Page No.** | **Signature** |
| 1 | Program to find The Odd and Even Numbers within a range | 1 |  |
| 2 | Program to print The Pattern within a range like -----  \*  \* \*  \* \* \*  \* \* \* \* | 2 |  |
| 3 | Program to print The Pattern Down-to-Earth within a range like -----  \*\*\*\*\*  \*\*\*\*  \*\*\*  \*\*  \* | 3 |  |
| 4 | Program to join The Previous Two Pattern into One Pattern like -----  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \* | 4-5 |  |
| 5 | Program to check whether a given Triangle is Scalene Triangle, Isosceles Triangle or Equilateral Triangle. | 6 |  |
| 6 | Program to find out The Factor of a Number. | 7 |  |
| 7 | Program to print The Multiplication Table of Any Number given by The User. | 8 |  |
| 8 | Program to check if The Number is Prime Number or not. | 9 |  |
| 9 | Program to check all The Prime Numbers within a range. | 10 |  |
| 10 | Program to show The Factorial Number using Recursion. | 11 |  |
| 11 | Program to show The Fibonacci Series using Recursion. | 12 |  |
| 12 | Program to calculate The Sum a List of Numbers. | 13 |  |
| 13 | Program to calculate The G.C.D of Two Numbers in recursive way. | 14 |  |
| 14 | Program to get The Sum of Digits of a Non-negative Integer Number. | 15 |  |
| 15 | Program to calculate The Sum of Positive Integer Numbers of n+( n-2)+(n-4 )+ (n-6)..... Until n-x<= 0 | 16 |  |
| 16 | Program to create a tuple :-----   1. Add an item. 2. Find the fourth element from the end. 3. Unpack the tuple. 4. Create a clone of the tuple. 5. Find the repetitive item and count its occurrence. 6. Remove an item by name. 7. Remove an item by its position or index. 8. Reverse the tuple. | 17-20 |  |
| 17 | Program where Two Python Dictionaries (dict1, dict2) contains Name and Marks of 4 and 5 Students respectively :   1. Add Another Key to dict1. 2. Concatenate Two Dictionaries to form a New Dictionary. 3. Sort The New Dictionary on The basis of Marks. | 21-22 |  |
| 18 | Program where Two Python Sets (setA, setB) contains some values respectively :   1. Add an Item in both The Sets. 2. Remove an Items from The Sets. 3. Print The Union of The Sets. 4. Print The Intersection of The Sets. 5. Find out The Difference of The Sets. | 23-24 |  |

**1) Write a Python Program to find The Odd and Even Numbers within a range.**

**Syntax :**

#odd and even number

num1 = int(input("Enter the Start Value = "))

num2 = int(input("Enter the End Value = "))

print("The even number from ",num1,"to ",num2," are ---- ")

for i in range(num1,num2+1,2):

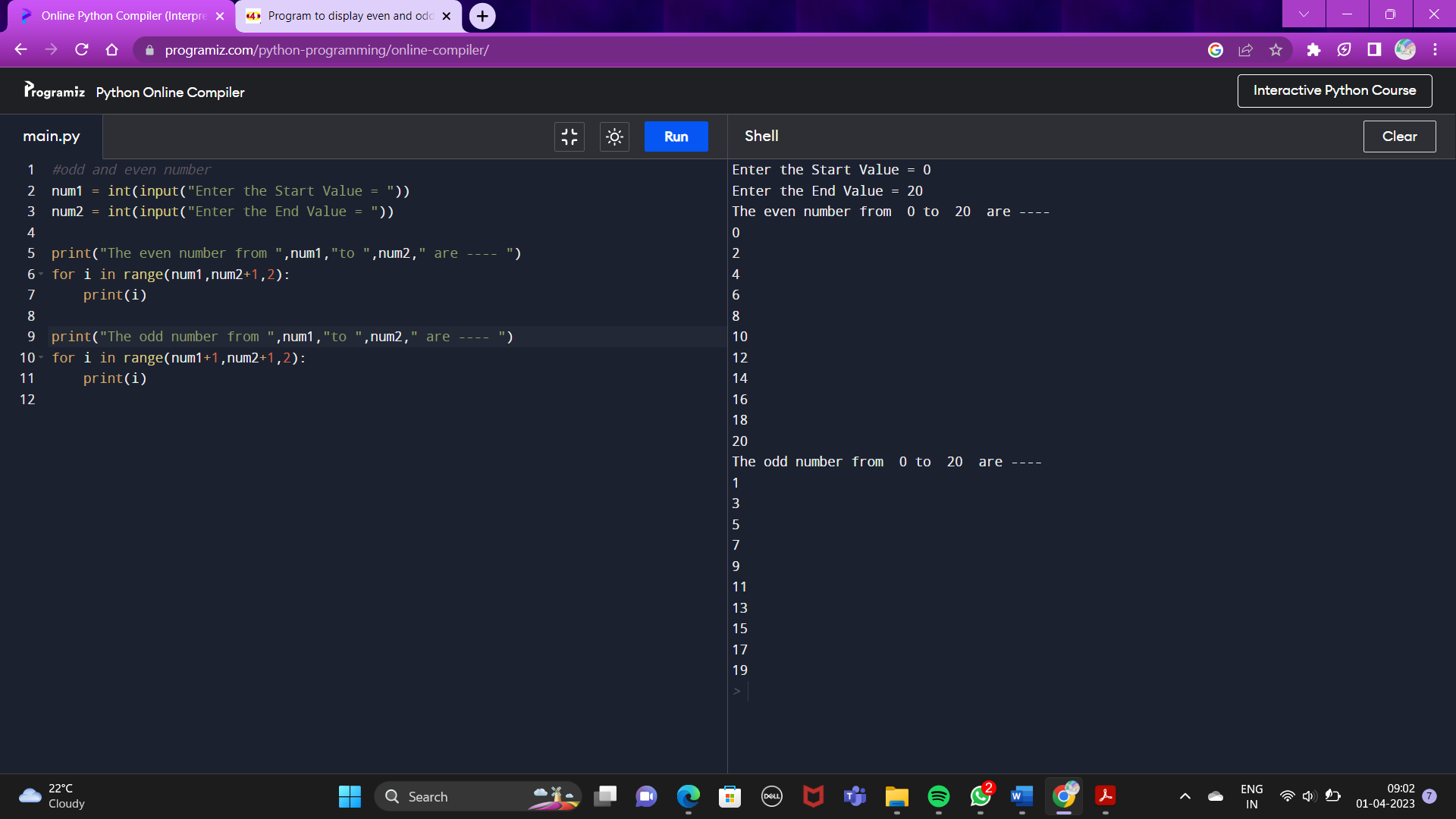
print(i)

print("The odd number from ",num1,"to ",num2," are ---- ")

for i in range(num1+1,num2+1,2):

print(i)

**Output :**



**2) Write a Python Program to print The Pattern within a range like -----**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**Syntax :**

#Pattern

val = int(input("Enter the number of pattern to be printed = "))

print("PATTERN = ")

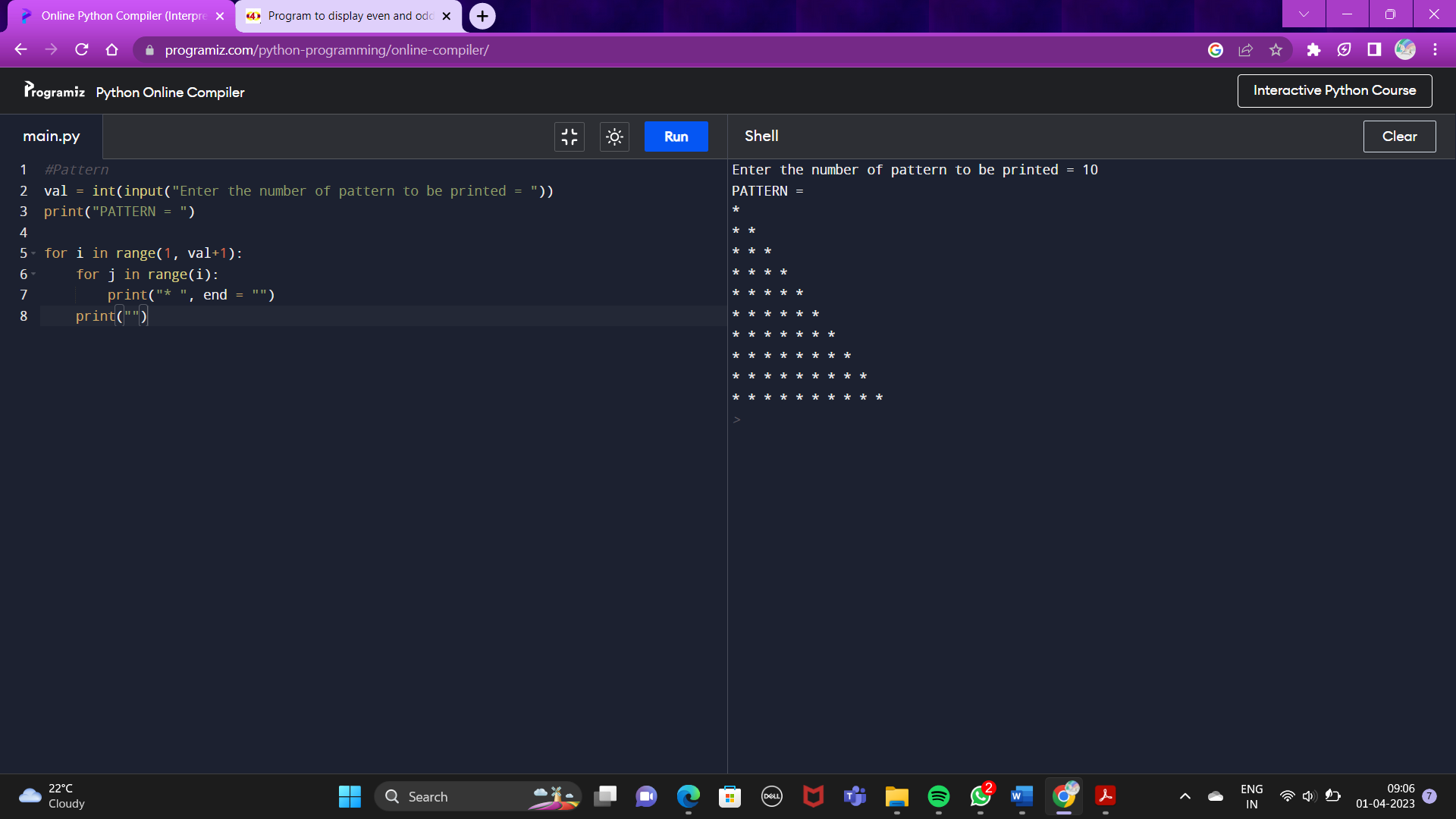
for i in range(1, val+1):

for j in range(i):

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

print("")

**Output :**



**3) Write a Python Program to print The Pattern Down-to-Earth within a range like -----**

**\*\*\*\***

**\*\*\***

**\*\***

**\***

**Syntax :**

#Down to earth pattern

val = int(input("Enter the number of pattern to be printed = "))

print("PATTERN = ")

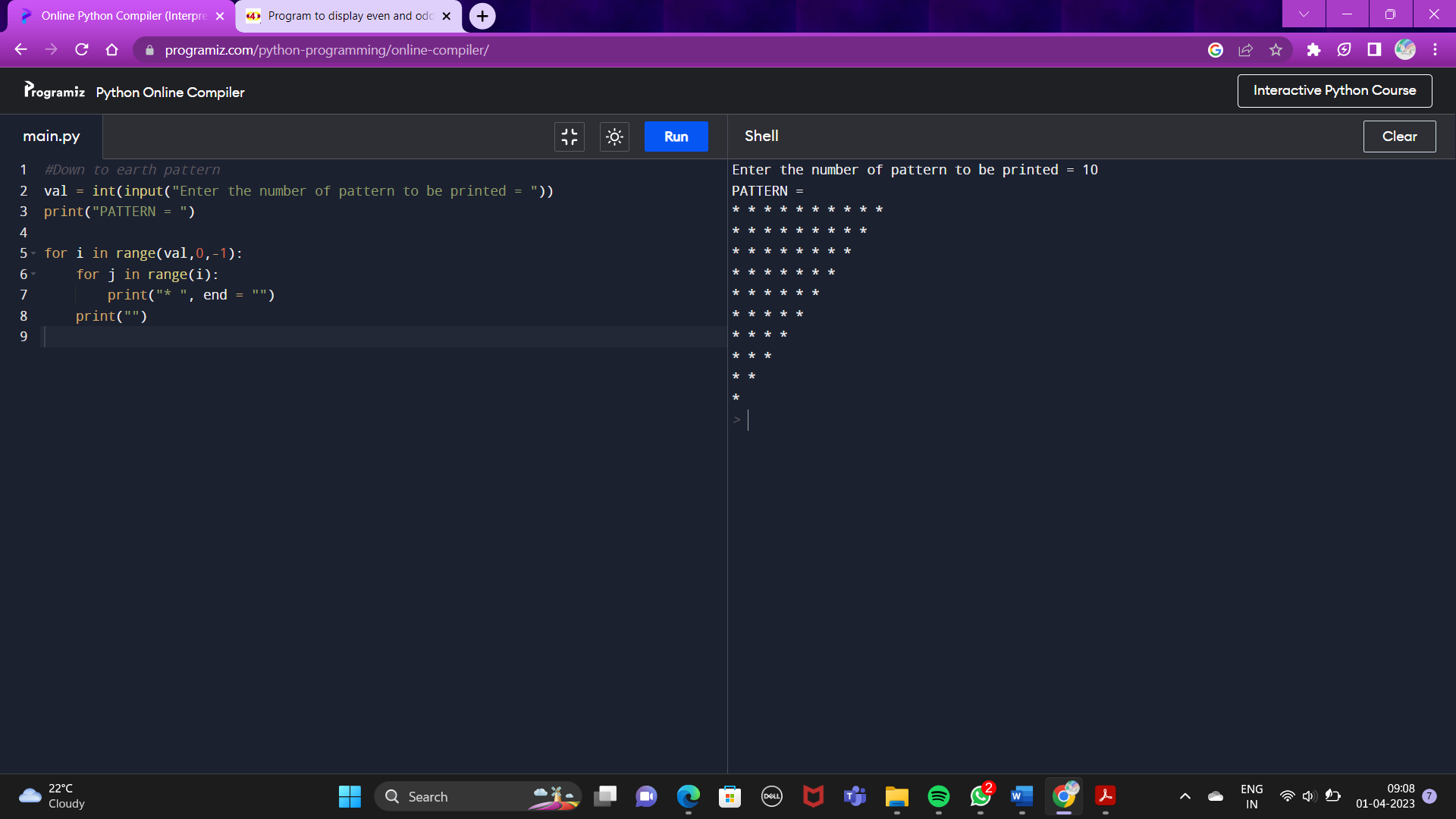
for i in range(val,0,-1):

for j in range(i):

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

print("")

**Output :**



**4) Write a Python Program To join The Previous Two Pattern into One Pattern like -----**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

**\* \* \* \***

**\* \* \***

**\* \***

**\***

**Syntax :**

#Joining both pattern

pat = int(input("Enter the number of pattern to be printed = "))

val=pat//2

print("PATTERN = ")

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

for j in range(i):

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

print("")

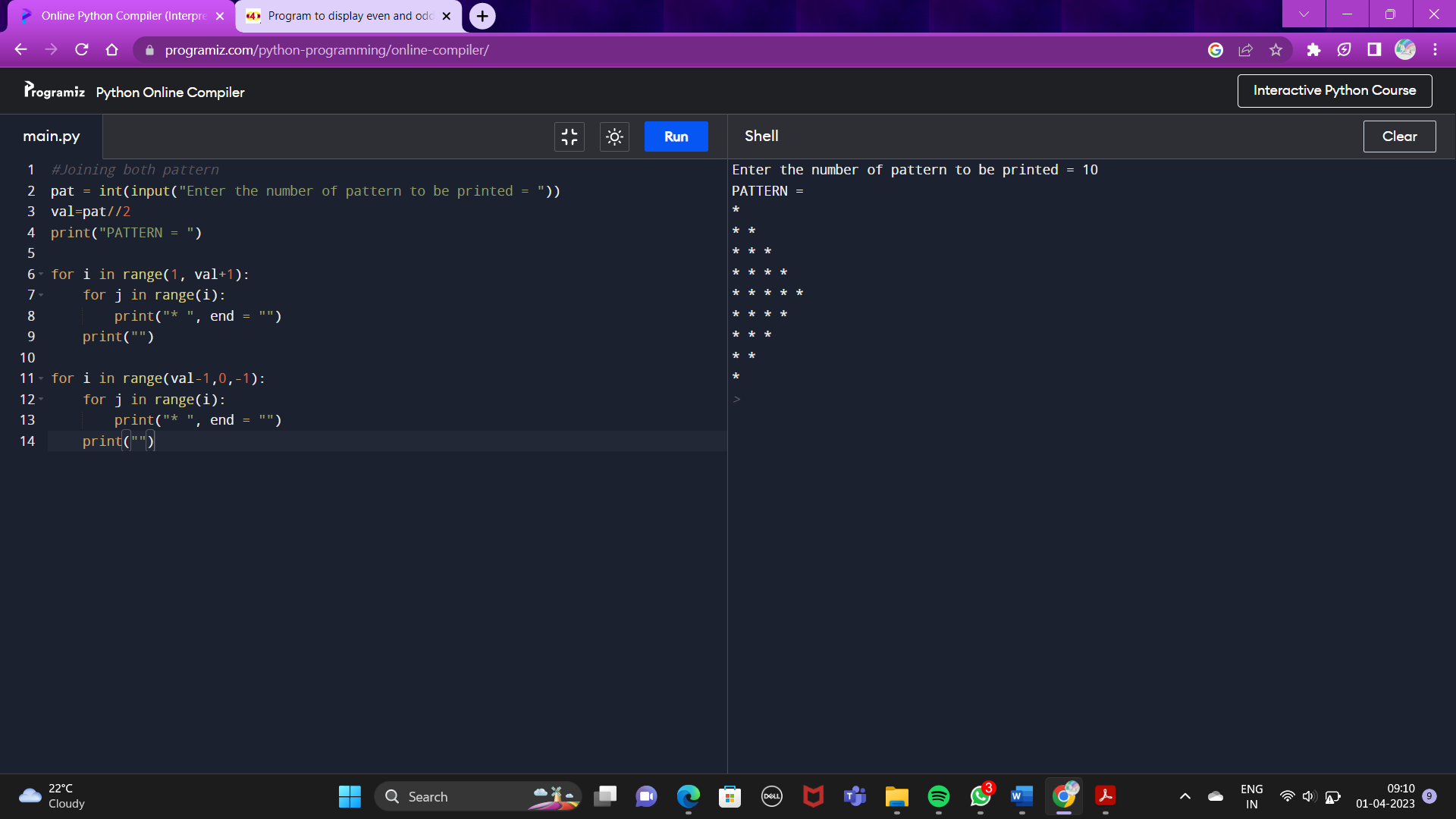
for i in range(val-1,0,-1):

for j in range(i):

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

print("")

**Output :**



**5) Write a Python Program to check whether a given Triangle is Scalene Triangle, Isosceles Triangle or Equilateral Triangle.**

**Syntax :**

#Types of triangle

num1 = float(input("Enter The Value of first side of a Triangle = "))

num2 = float(input("Enter The Value of second side of a Triangle = "))

num3 = float(input("Enter The Value of third side of a Triangle = "))

if(num1==num2 and num2==num3 and num3==num1):

print("This is a Equilateral Triangle.."

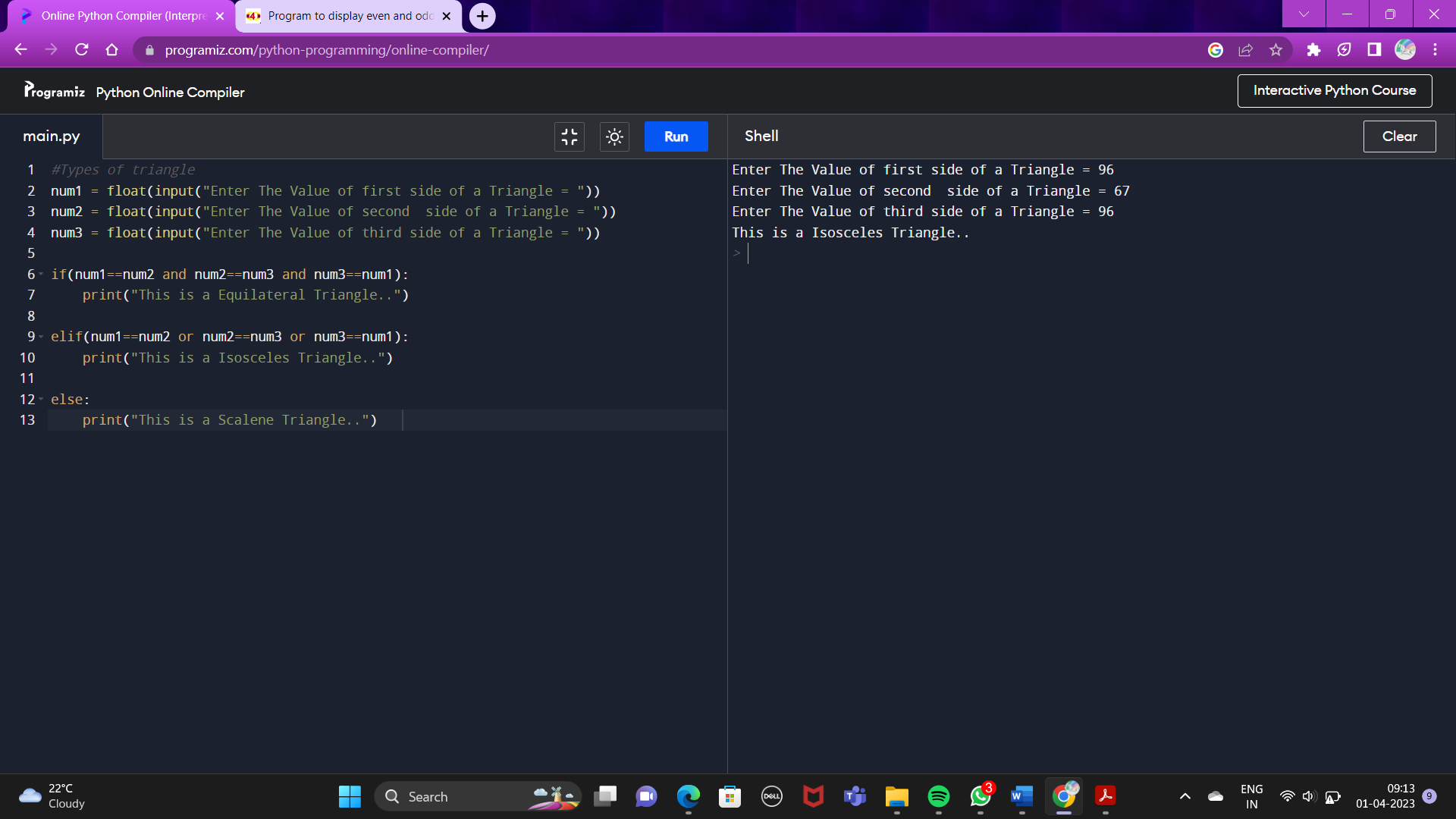
elif(num1==num2 or num2==num3 or num3==num1):

print("This is a Isosceles Triangle..")

else:

print("This is a Scalene Triangle..")

**Output :**

**6) Write a Python Program to find out The Factor of a Number.**

**Syntax :**

#Factors of a Number

num = int(input("Enter The Number = "))

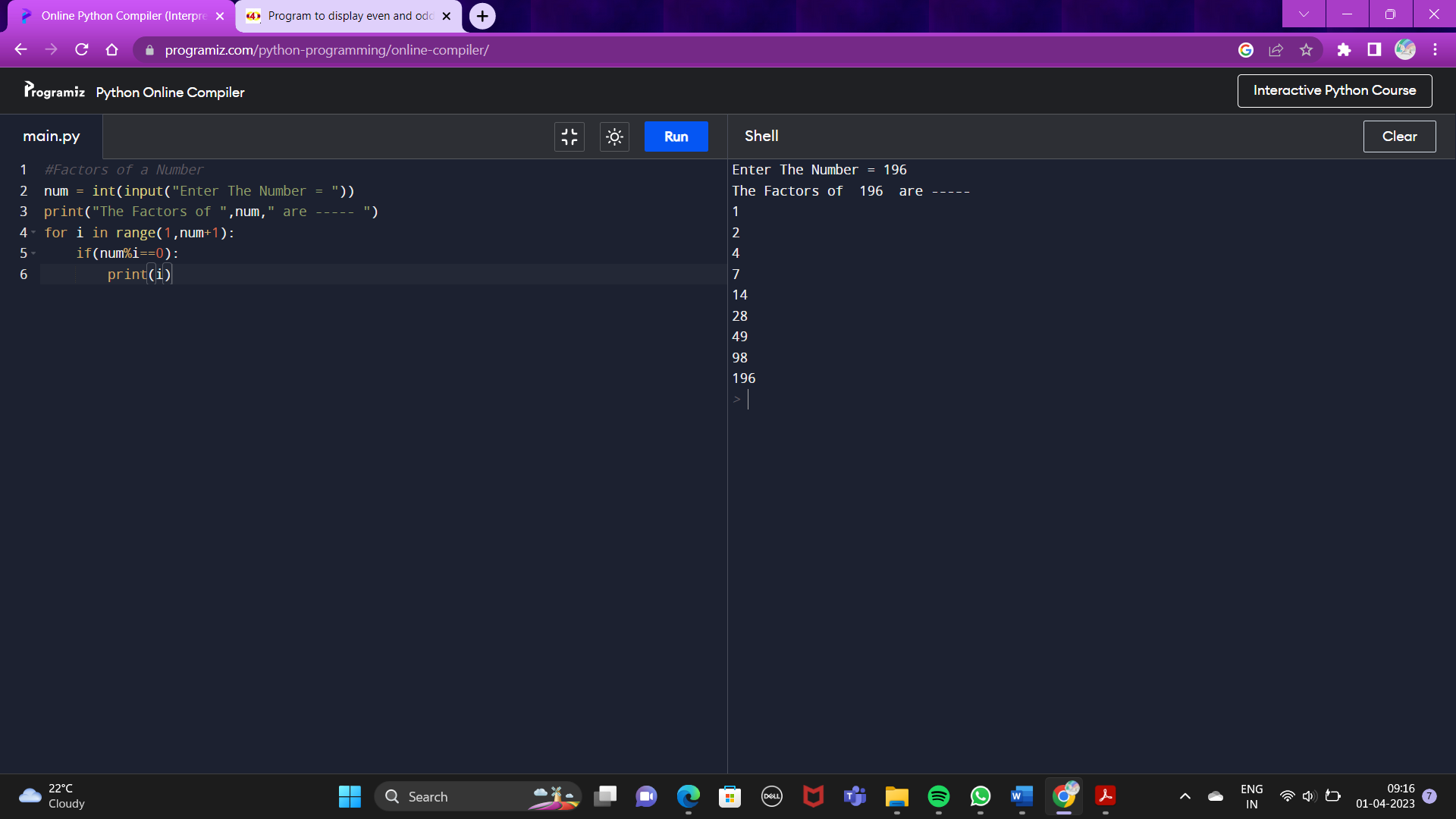
print("The Factors of ",num," are ----- ")

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

if(num%i==0):

print(i)

**Output :**



**7) Write a Python Program to print The Multiplication Table of Any Number given by The User.**

**Syntax :**

#Multiplication table

num = int(input("Enter The Number = "))

print("The Multiplication Table of ",num,"-----")

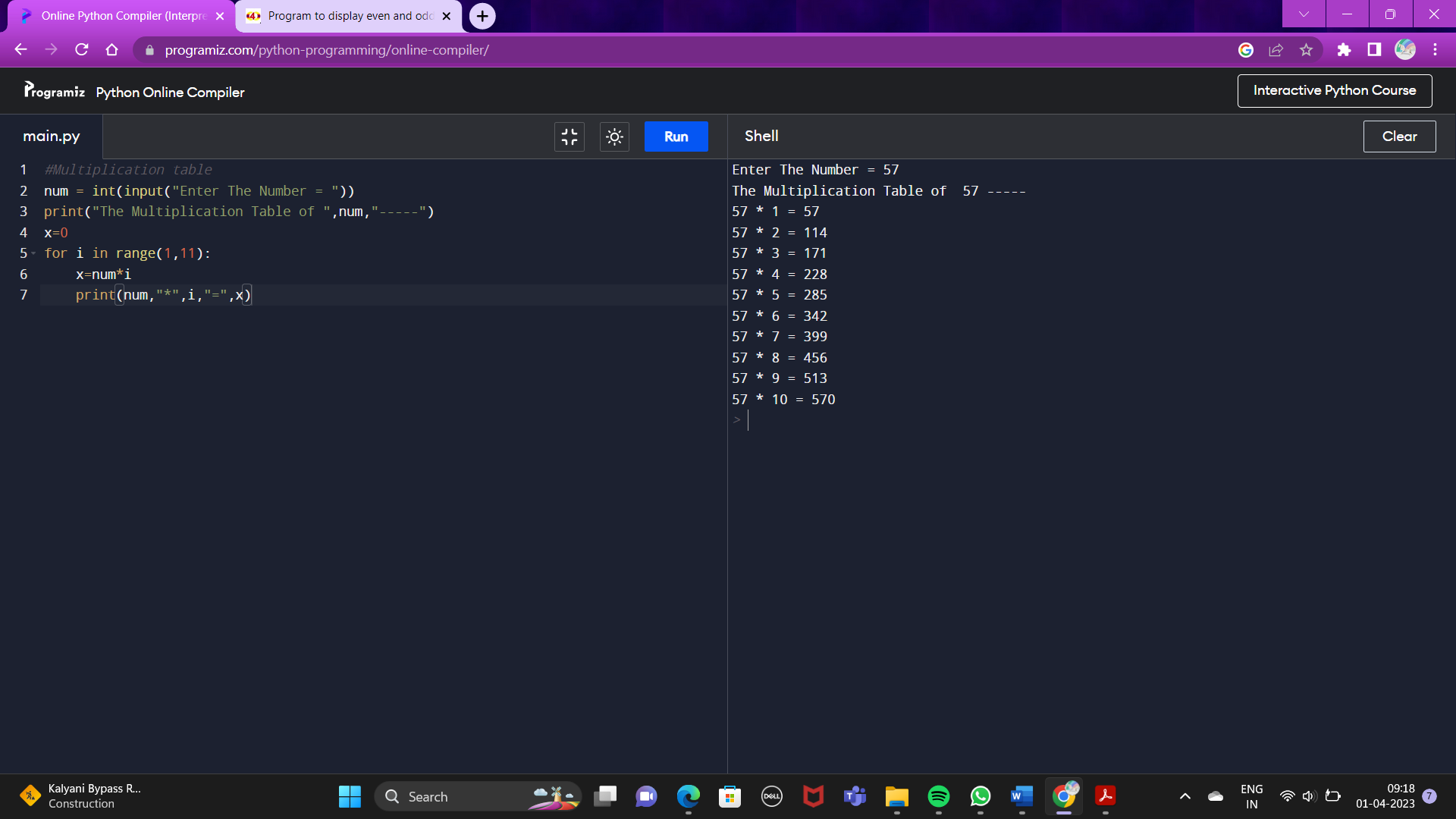
x=0

for i in range(1,11):

x=num\*i

print(num,"\*",i,"=",x)

**Output :**



**8) Write a Python Program to check if The Number is Prime Number or not.**

**Syntax :**

#Check The Prime Number

num = int(input("Enter The Number = "))

x=0

for i in range(1,(num//2)+1):

if(num%i==0):

x=x+1

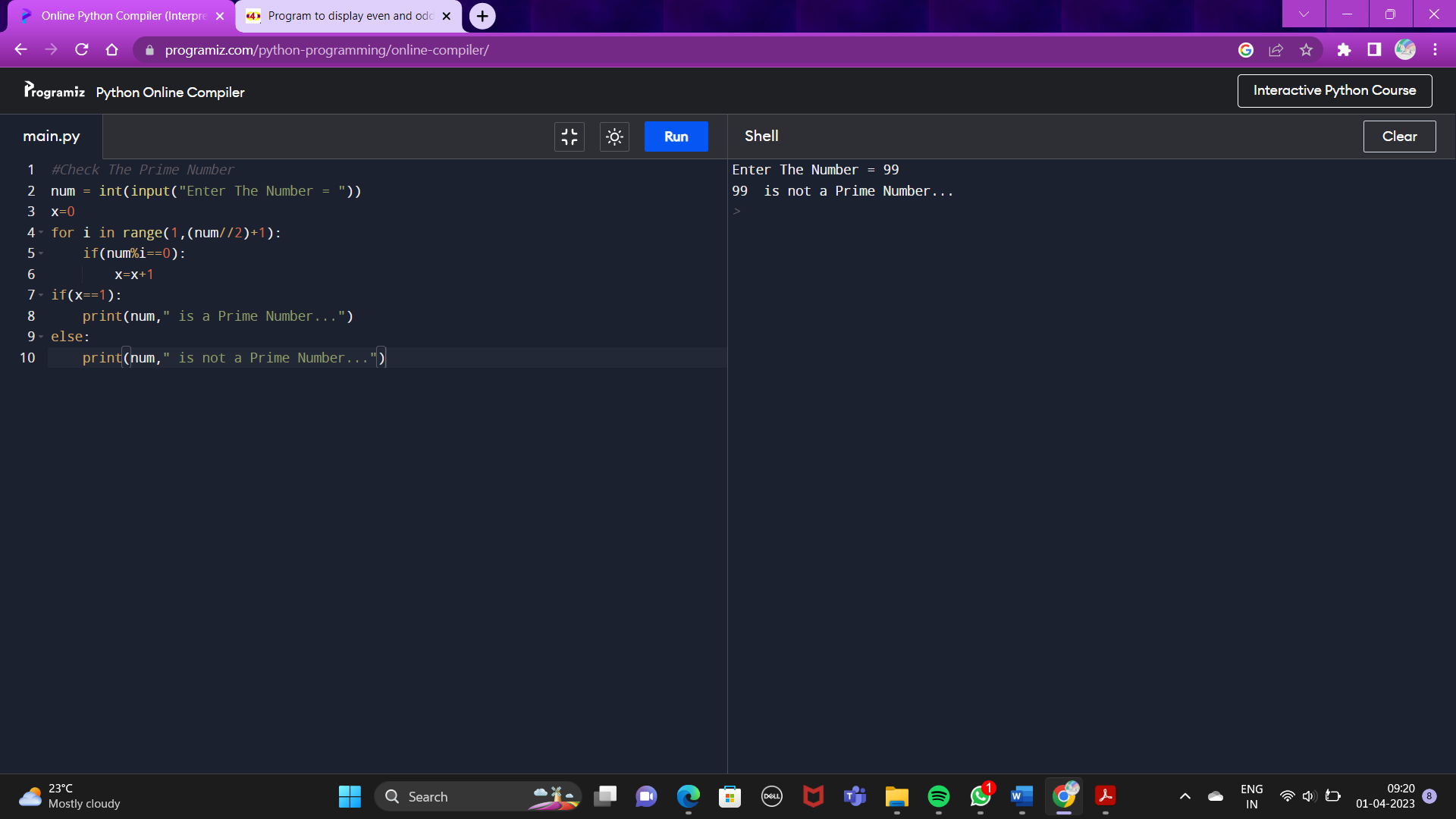
if(x==1):

print(num," is a Prime Number...")

else:

print(num," is not a Prime Number...")

**Output :**



**9) Write a Python program to check All The Prime Numbers within a range.**

**Syntax :**

#check prime number within a range

num1 = int(input("Enter The Number for The Lower Range = "))

num2 = int(input("Enter The Number for The Upper Range = "))

val=0

for i in range(num1,num2+1):

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

if(i%j==0):

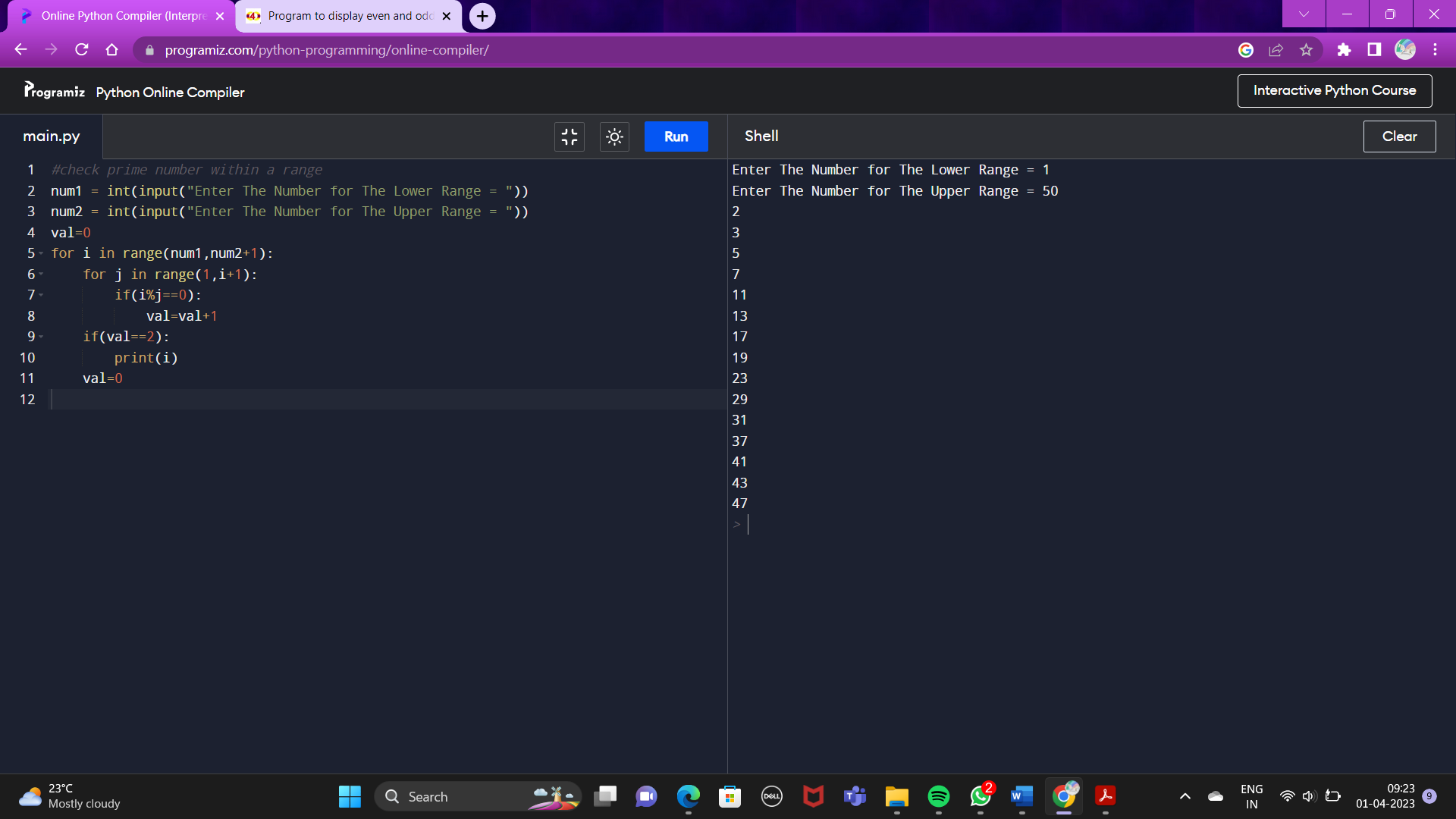
val=val+1

if(val==2):

print(i)

val=0

**Output :**



**10) Write a Python Program to show The Factorial Number using Recursion.**

**Syntax :**

#Factorial Number using Recursion

num = int(input("Enter The Number = "))

def my\_func(temp,num):

if(num>0):

temp=temp\*num

my\_func(temp,num-1)

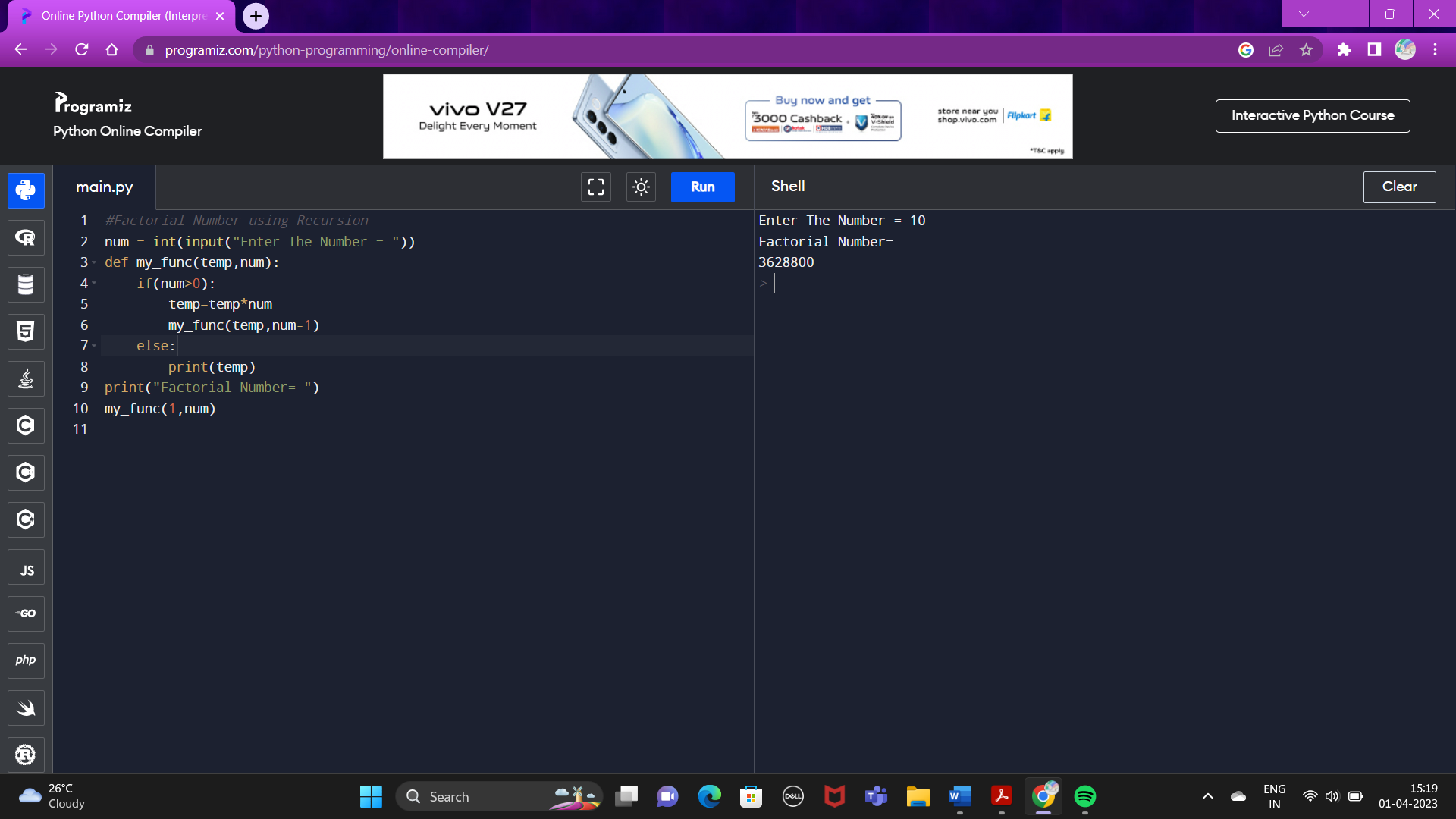
else:

print(temp)

print("Factorial Number= ")

my\_func(1,num)

**Output :**



**11) Write a Python Program to show The Fibonacci Series using Recursion.**

**Syntax :**

#Fibonacci Series using Recursion

temp = int(input("Enter The Number = "))

def my\_func(num1,num2,sum,temp):

if(temp>0):

print(sum)

my\_func(num2,sum,num2+sum,temp-1)

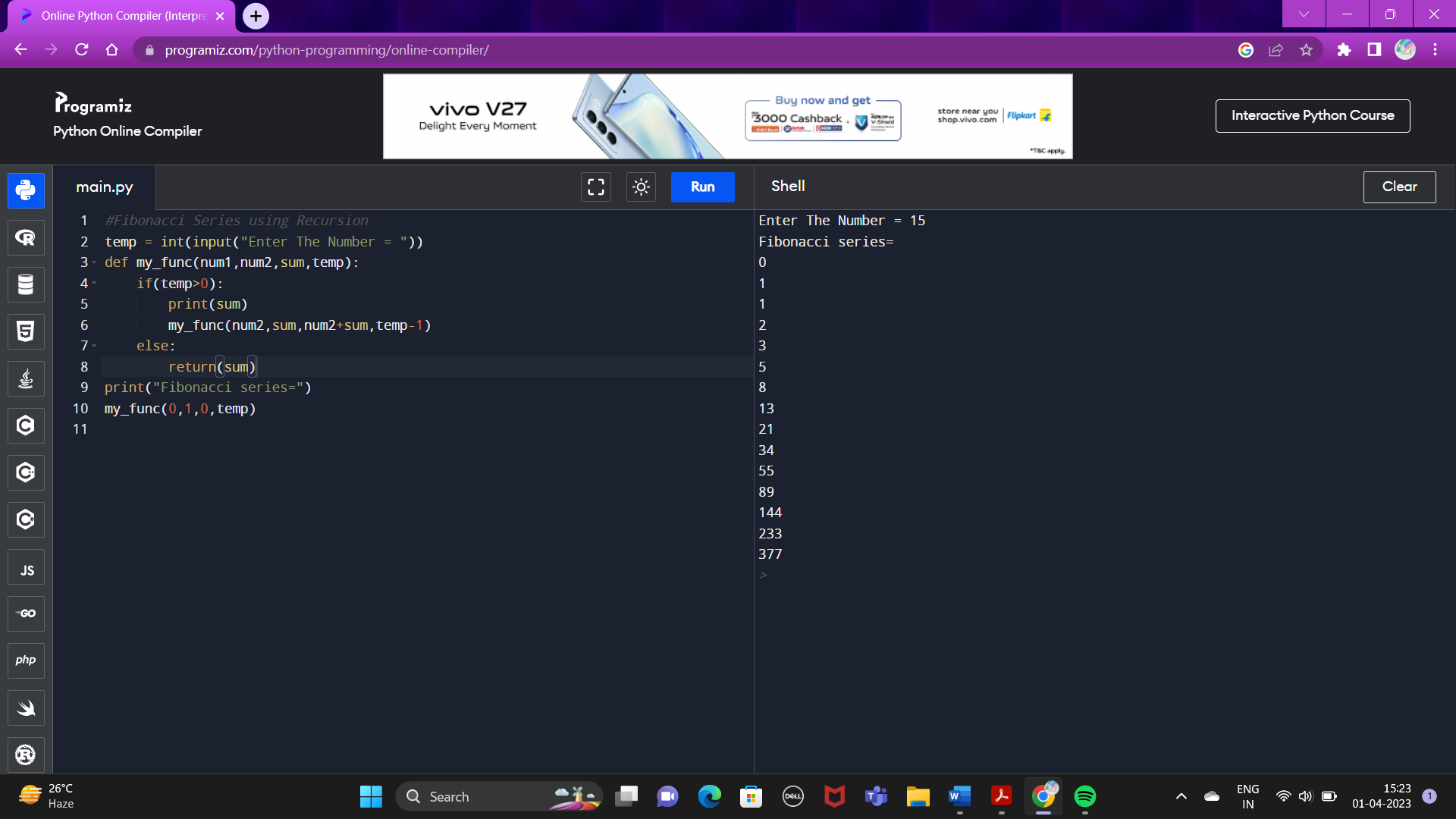
else:

return(sum)

print("Fibonacci series=")

my\_func(0,1,0,temp)

**Output :**



**12) Write a Python Program to calculate The Sum of a List of Numbers.**

**Syntax :**

#Sum of a List of Numbers

def my\_func(num):

sum=0

for i in num:

sum=sum+i

print("The Sum of This List of Number = ",sum)

temp=[]

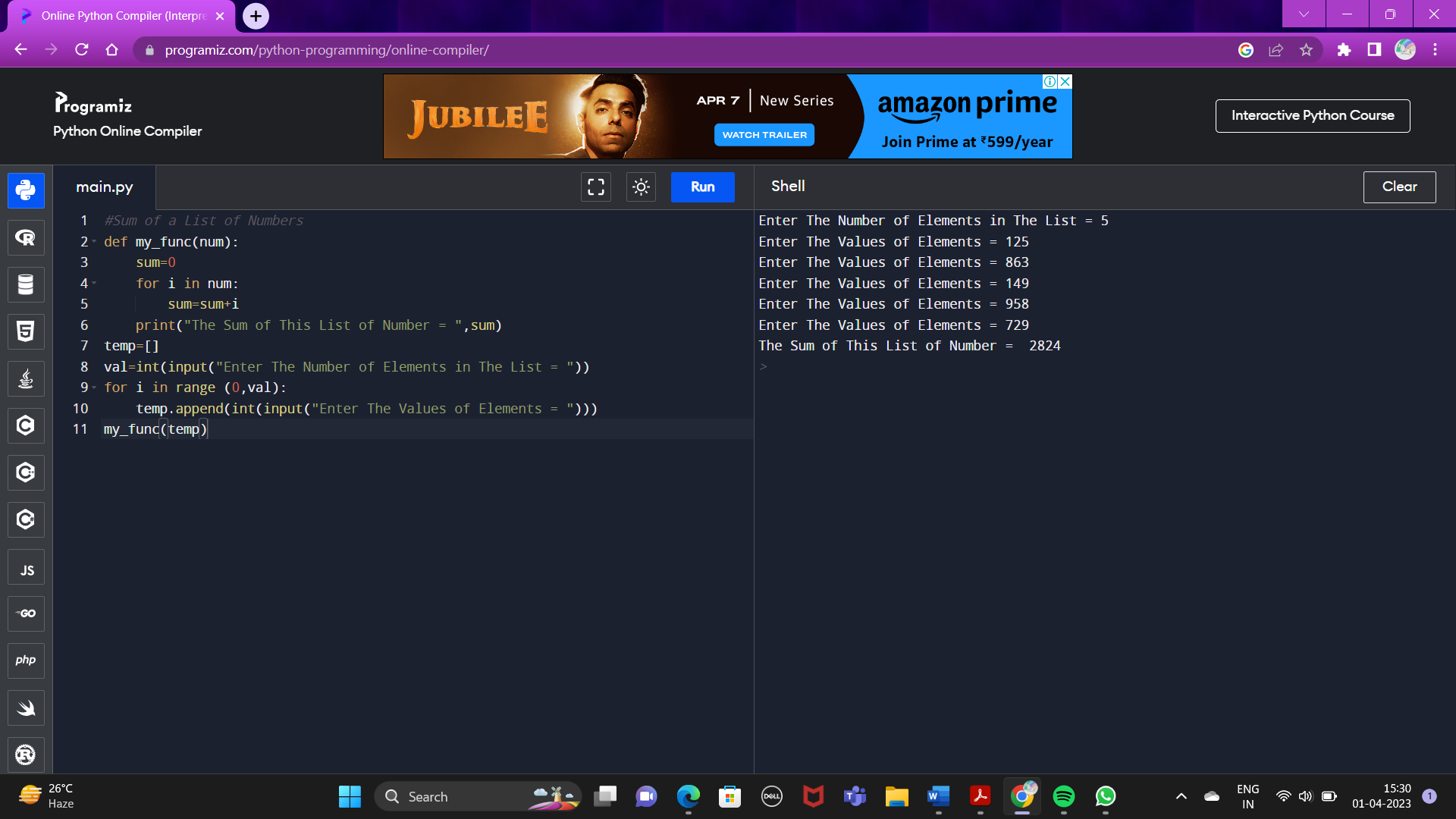
val=int(input("Enter The Number of Elements in The List = "))

for i in range (0,val):

temp.append(int(input("Enter The Values of Elements = ")))

my\_func(temp)

**Output :**



**13) Write a Python Program to calculate The G.C.D of Two Numbers in Recursive way.**

**Syntax :**

# Recursive Method for G.C.D calculation

def hcf(a, b):

if(b == 0):

return a

else:

return hcf(b, a % b)

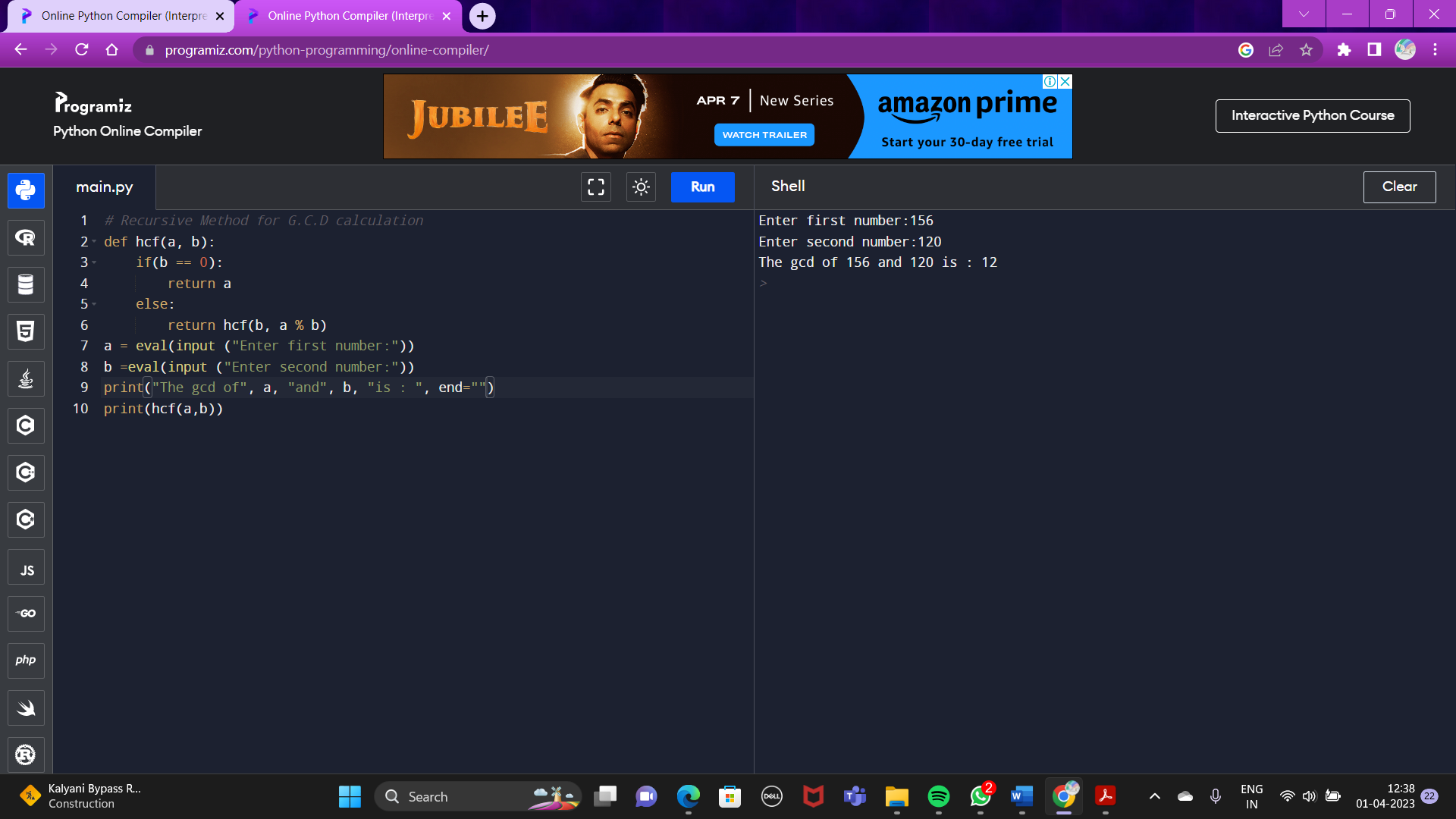
a = eval(input ("Enter first number:"))

b =eval(input ("Enter second number:"))

print("The gcd of", a, "and", b, "is : ", end="")

print(hcf(a,b))

**Output :**

**14) Write a Python Program to get The Sum of Digits of a Non-negative Integer Number.**

**Syntax :**

#Sum of digits of a non-negetive integer number

def my\_func(sum,num):

temp=0

if(num<=0):

print("Sum of The Given Number = ",sum)

else:

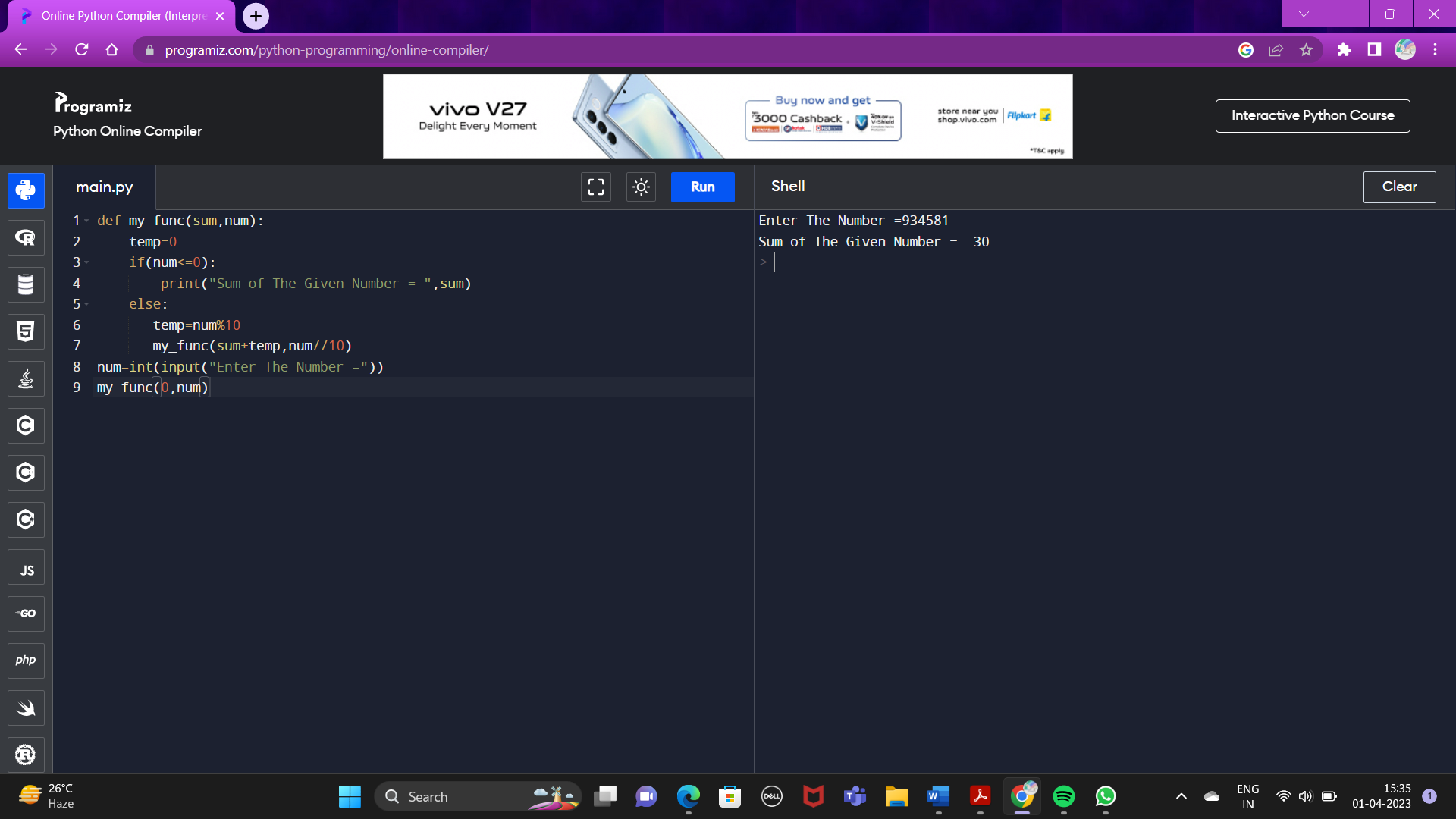
temp=num%10

my\_func(sum+temp,num//10)

num=int(input("Enter The Number ="))

my\_func(0,num)

**Output :**



**15) Write a Python Program to calculate The Sum of Positive Integer Numbers of n+( n-2)+(n-4 )+ (n-6)..... Until n-x<= 0**

**Syntax :**

#The sum of Positive Integers

def my\_func(num):

return lambda x:num-x

num=int(input("Enter The Number = "))

s=my\_func(num)

i=0

d=0

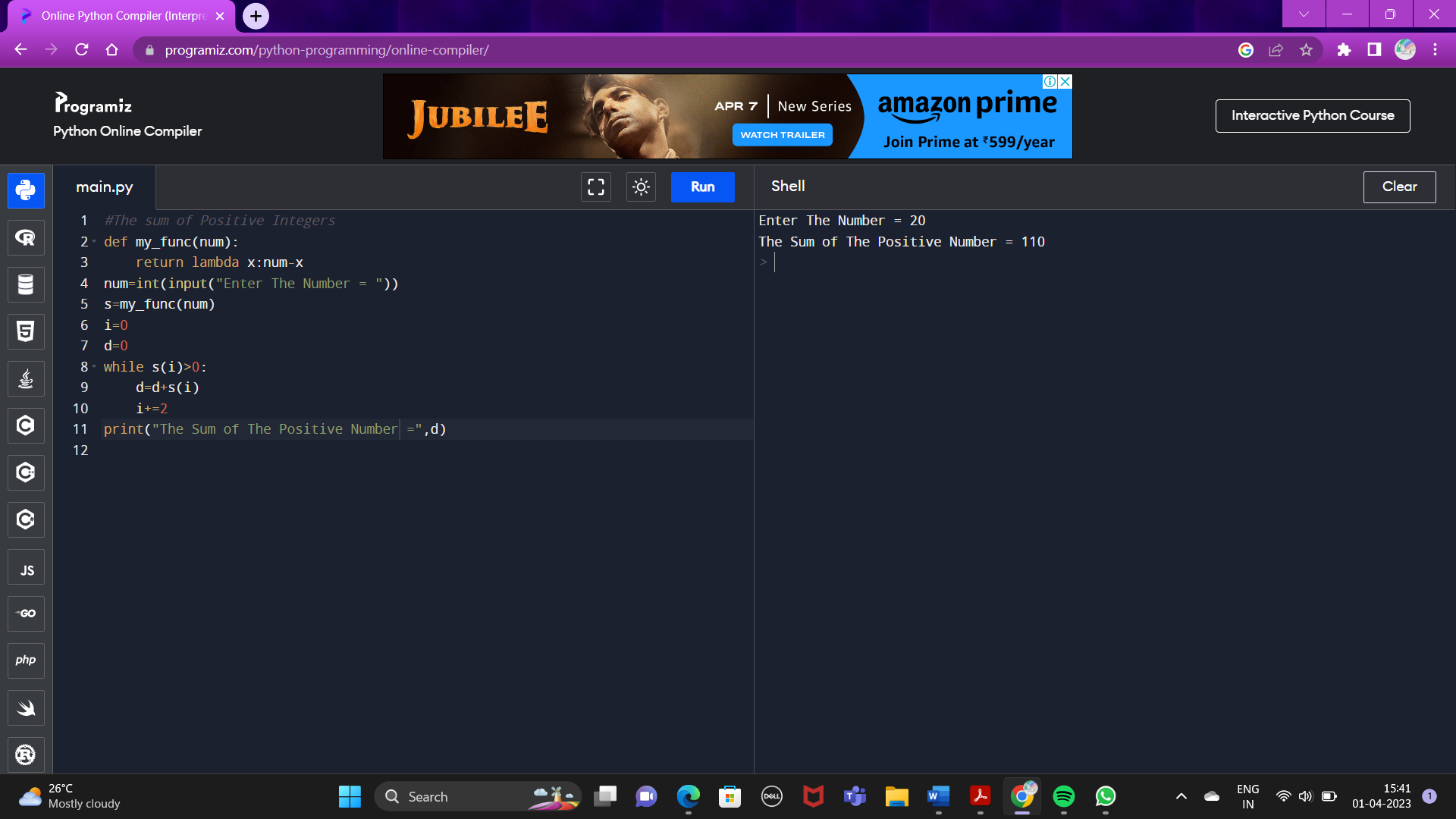
while s(i)>0:

d=d+s(i)

i+=2

print("The Sum of The Positive Number =",d)

**Output :**



**16) Write a Python Program to create a tuple :-----**

**1. Add an item.**

**2. Find the fourth element from the end.**

**3. Unpack the tuple.**

**4. Create a clone of the tuple.**

**5. Find the repetitive item and count its occurrence.**

**6. Remove an item by name.**

**7. Remove an item by its position or index.**

**8. Reverse the tuple.**

**Syntax :**

print("Tuple created...")

nm=()

l=[]

for i in range(0,11):

l.append(eval(input("Enter item in the tuple:")))

nm=nm+tuple(l)

print("Original Tuple..")

print(nm)

#adding item in tuple

l=[]

print("Adding an item....")

l.append(eval(input("Enter item to be added:")))

nm=nm+tuple(l)

print("New tuple:")

print(nm)

#printing the fourth element from the last

print("The Fourth item from the last=",nm[-4])

#unpacking a tuple

print("unpacking a collection...")

for i in nm:

print(i)

#cloning a tuple

print("Cloning a tuple...")

clo=()

clo=nm[:]

print(clo)

#find repeatative elements and its occurence

c=0

for i in nm:

c=c+nm.count(i)

if(c>1):

print(i,"is repeated",c,"times")

c=0

else:

c=0

#Removing an item by its name..

print("Removing an item by its name...")

r=eval(input("Enter item to be removed="))

l=list(nm)

l.remove(r)

nm=tuple(l)

print("New tuple:")

print(nm)

#Removing an item by its position or index..

print("Removing an item by its position or index...")

r=inp(r)

nm=tuple(l)

print("New tuple:")

print(nm)

#reversing a tuple

print("Reversing a tuple...")

rev=()

rev=nm[::-1]

print(rev)t(input("Enter index of the element to be removed="))

l=list(nm)

l.pop(r)

nm=tuple(l)

print("New tuple:")

print(nm)

#reversing a tuple

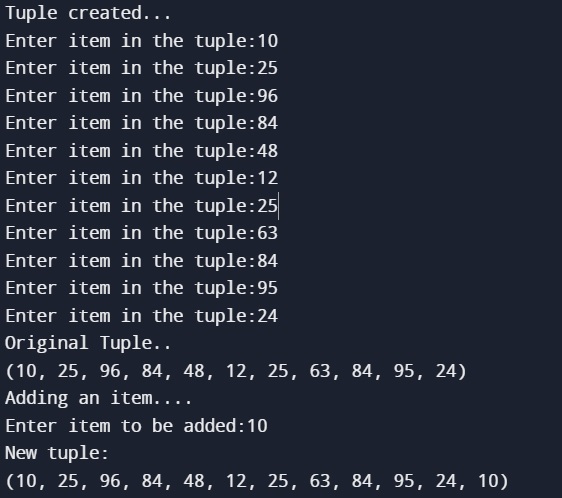
print("Reversing a tuple...")

rev=()

rev=nm[::-1]

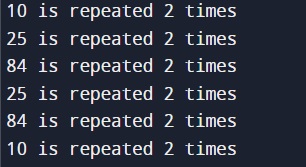
print(rev)

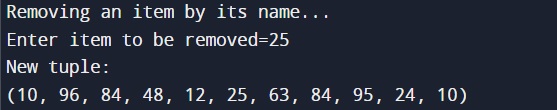
**Output :**

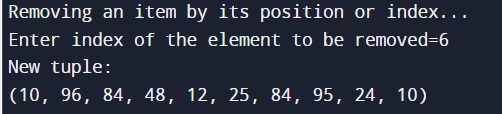


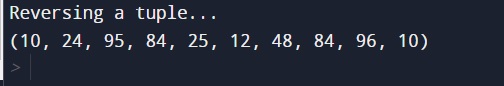


05.4.jpg









**17) Write a Python Program where Two Python Dictionaries (dict1, dict2) contains Name and Marks of 4 and 5 Students respectively :**

**1. Add Another Key to dict1.**

**2. Concatenate Two Dictionaries to form a New Dictionary.**

**3. Sort The New Dictionary on The basis of Marks.**

**Syntax :**

#Create the dictionary

dict1 = {"Sayan":90, "Sumit":95, "Ayan":80, "Anik":72}

dict2 = {"Subham":71, "Abhrodip":85, "Raj":69, "Lekhan":83,"Ankan":85}

print("The Dictionary dict1 = ", dict1)

print("The Dictionary dict1 = ", dict2)

# Add another key in Dict1

dict1["Nirmalya"] = 81

print("Now After Adding a Key, The New Dictionary Dict1 = ",dict1)

# Concatenate The Two Dictionaries in One

dict3 = dict1

dict3.update(dict2)

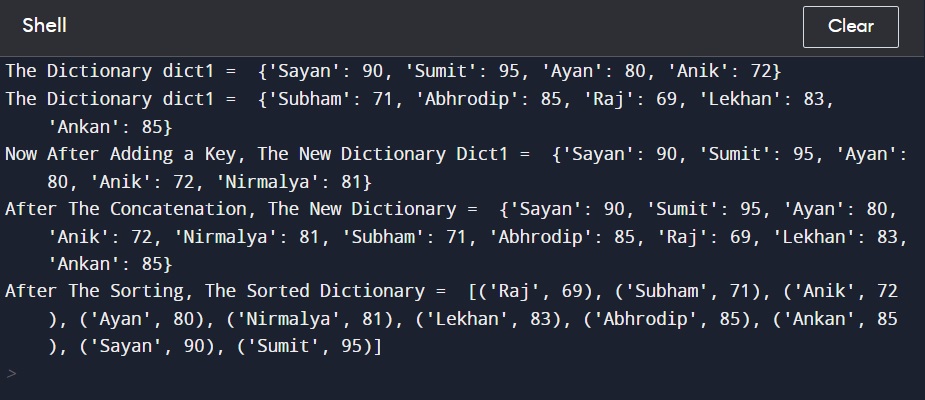
print("After The Concatenation, The New Dictionary = ", dict3)

#Sort The New Dictionary on The Basis of Marks

dict4 = sorted(dict3.items(), key=lambda items:items[1])

print("After The Sorting, The Sorted Dictionary = ", dict4)

**Output :**



**18) Write a Python Program where Two Python Sets (setA, setB) contains some values respectively :**

**1. Add an Item in both The Sets.**

**2. Remove an Items from The Sets.**

**3. Print The Union of The Sets.**

**4. Print The Intersection of The Sets.**

**5. Find out The Difference of The Sets.**

**Syntax :**

setA={0,1,3,5,9,7,6}

setB={2,4,8,5,9,3,7}

print("Original SetA:",setA)

print("Original SetB",setB)

#adding an element in a set

r=eval(input("Enter an element to be added in setA:"))

setA.add(r)

r1=eval(input("Enter an element to be added in setB:"))

setB.add(r1)

print("New SetA:",setA)

print("New SetB:",setB)

#removing element in a set

f=eval(input("Enter an element to be removed in setA:"))

setA.remove(f)

f1=eval(input("Enter an element to be removed in setB:"))

setB.remove(f1)

print("New SetA:",setA)

print("New SetB:",setB)

#union

setC={}

setC=setA

for i in setB:

setC.add(i)

print("UNION OF SET A and B=",setC)

#interesect

print("Intersection of A and B=",setA.intersection(setB))

#set difference of (A-B)

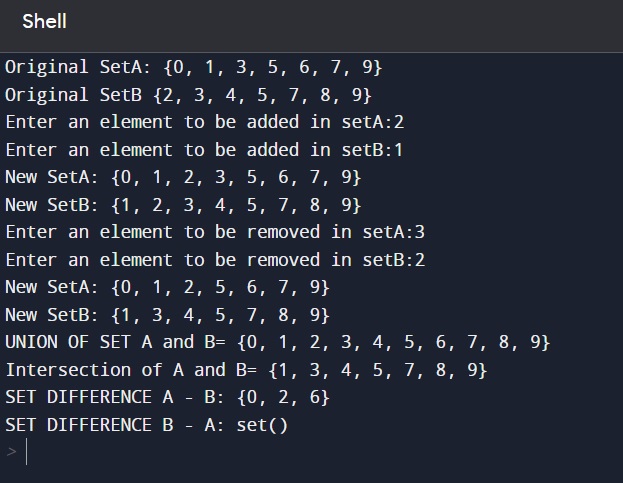
setD={}

setD=setA.difference(setB)

print("SET DIFFERENCE A - B:",setD)

#set difference of (B-A)

print("SET DIFFERENCE B - A:",setB.difference(setA))

**Output :**

**THANK YOU**