Name : Khushi Panwar

Roll No : 2021334

SEC (Python) PRACTICAL ASSIGNEMENT : 29.09.2022

Computer Science (3rd Sem)

**Question 10 (From Practical List ) : WAP that prints Armstrong numbers in the range 1 to 1000. An Armstrong number is a number whose sum of the cubes of the digits is equal to the number itself.**

**Solution 🡪**

print("The armstrong numbers in the range 1 to 1000 are : ")

for i in range(1,1000):

num=str(i)

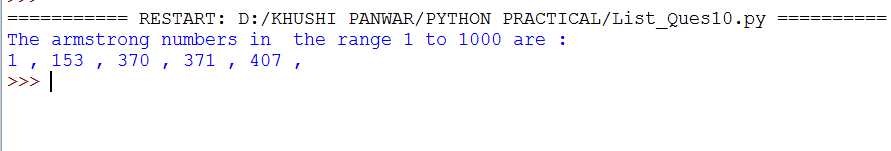
sum1=0

for j in num:

sum1=sum1+int(j)\*\*3

if sum1==int(num):

print(num, end=" , ")



**Question 11 (From Practical List ) : Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co-primes . Two numbers are said to be co-prime if they do not have any common divisor other than one.**

**Solution 🡪**

print("Write a function that takes two numbers as input parameters and returns True or False depending on whether they are co-primes . Two numbers are said to be co-prime if they do not have any common divisor other than one.")

print()

def commondivisor(a,b):

for i in range(2,a+1):

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

return 1

return 0

num1=int(input("Enter 1st number : "))

num2=int(input("Enter 2nd number : "))

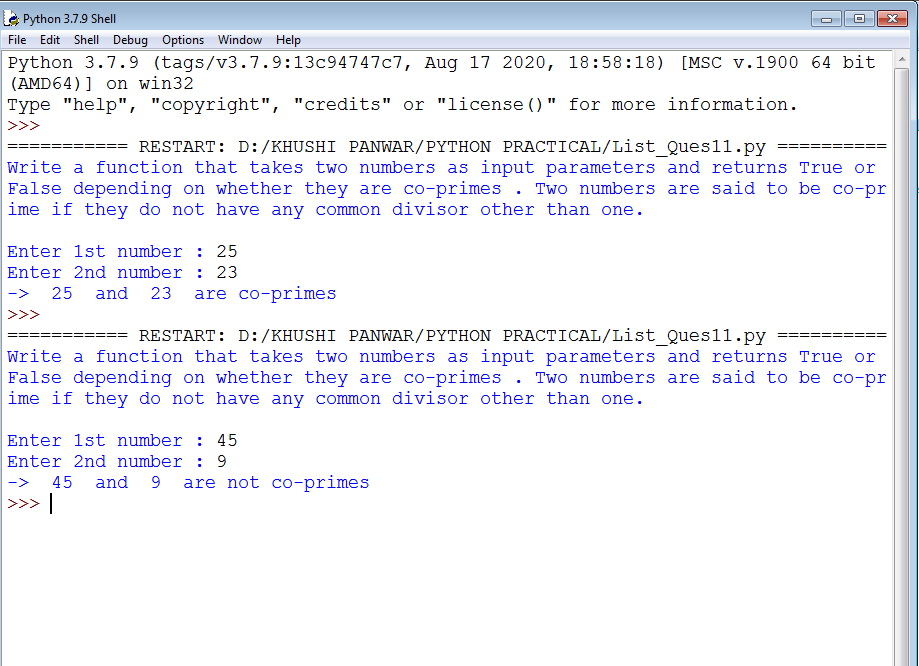
res=commondivisor(num1,num2)

if (res==1):

print("-> ",num1, " and ",num2, " are not co-primes")

else:

print("-> ",num1, " and ",num2, " are co-primes")



**Question 12 (From Practical List ) : Write a function sqrt that takes a non-negative number as an input and computes its square root. We can solve this problem iteratively. You will recall from high school mathematics that to find the square root of a number, say 2, we need to solve the equation f(x)=x-2 =0. To begin with, choose two numbers a and b so that f(a)<0 and f(b)>0. Now, for the equation f(x)=x^2-2=0, f(1)<0 and f(2)>0. So, the root of the equation must lie in the interval [a,b] (i.e. [1,2]). We find the midpoint, say mid of the interval [a,b]. If f(a)<0 and f(mid)>0, we know that the root of the equation f(x)=0 lies in the interval [a,b] (i.e. [1,2]). We find the midpoint, say, mid of the interval [a,b]. If f(a)<0 and f(mid)>0, we know that the root of the equation f(x)=0 lie sin the interval [a,mid]. However, in other case, (f(mid)<0 and f(mid)>0), the root of the equation f(x)=0 must lie in the interval [mid,b].Thus, for the next iteration, we have reduced the search interval for the root of the equation to half, i.e. form [a,b] to [a,mid] or [mid,b].Proceeding in this way, we find a good approximation to the root of the equation when the length of the search interval becomes sufficiently small, say, 0.01. The following table depicts the steps for computing square root approximation for the number 2.**

**Solution 🡪**

print("Write a function sqrt that takes a non-negative number as an input and computes its square root. \n")

def square(n, i ,j):

mid=(i+j)/2

mul=mid\*mid;

if ((mul==n) or (abs(mul-n)<0.00001)):

return mid

elif (mul<n):

return square(n,mid,j)

else:

return square(n,i,mid)

def findSqrt(n):

i=1

found=False

while(found==False):

if (i\*i==n):

print(i)

found=True

elif (i\*i>n):

res=square(n,i-1,i)

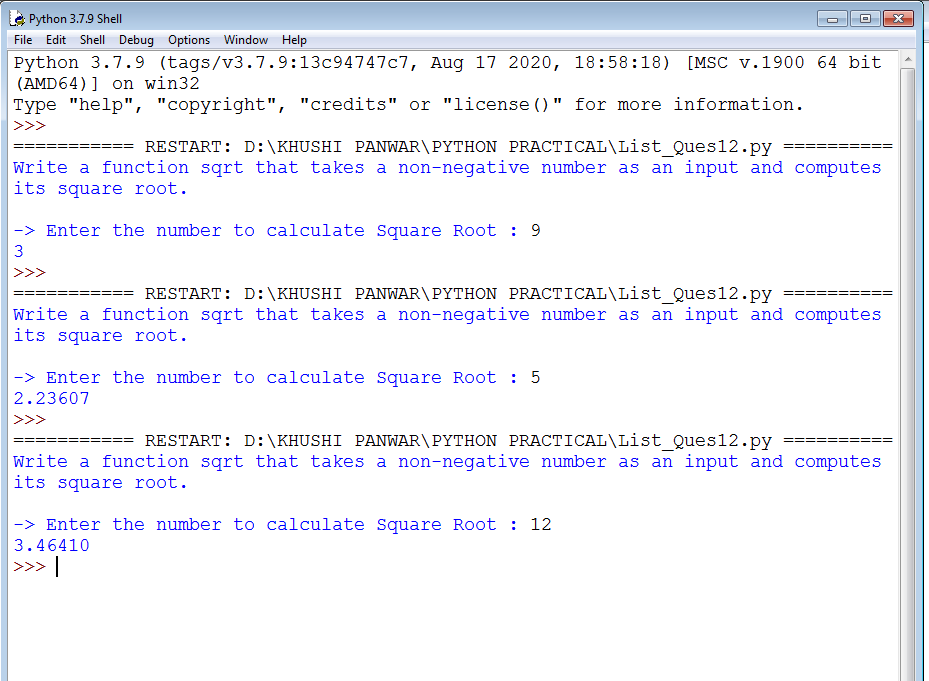
print("{0:.5f}".format(res))

found=True

i+=1

num=int(input("-> Enter the number to calculate Square Root : "))

findSqrt(num)



**Question 13 (From Practical List ) : Write a function to multiply two non-negative numbers by repeated addition. For example, 7\*5 = 7+7+7+7+7.**

**Solution 🡪**

print("Write a function to multiply two non-negative numbers by repeated addition. \n")

num1=int(input("Enter a number : "))

num2=int(input("Enter another number : "))

result=0

if (num1>0 and num2>0):

for i in range(0,num1):

result+=num2

print(num2,end=" + ")

print("0 = ", result)

else:

print("Enter valid numbers!")

