Python Lab session 9 (05-03-2021)

Assignment

Name :- Purushottam Kumar

ID :- 2041

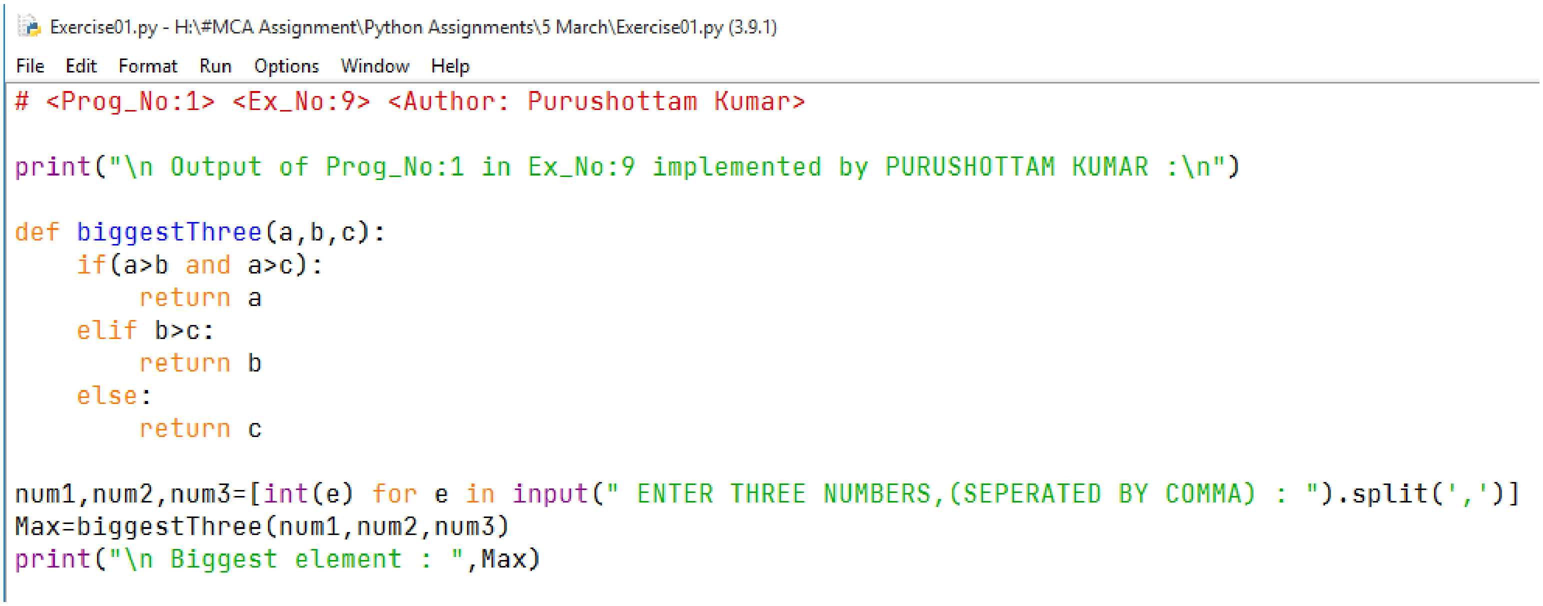
MCA I-Sem (R)

Submission Date :- 06-Mar-2021

(Program – 1)

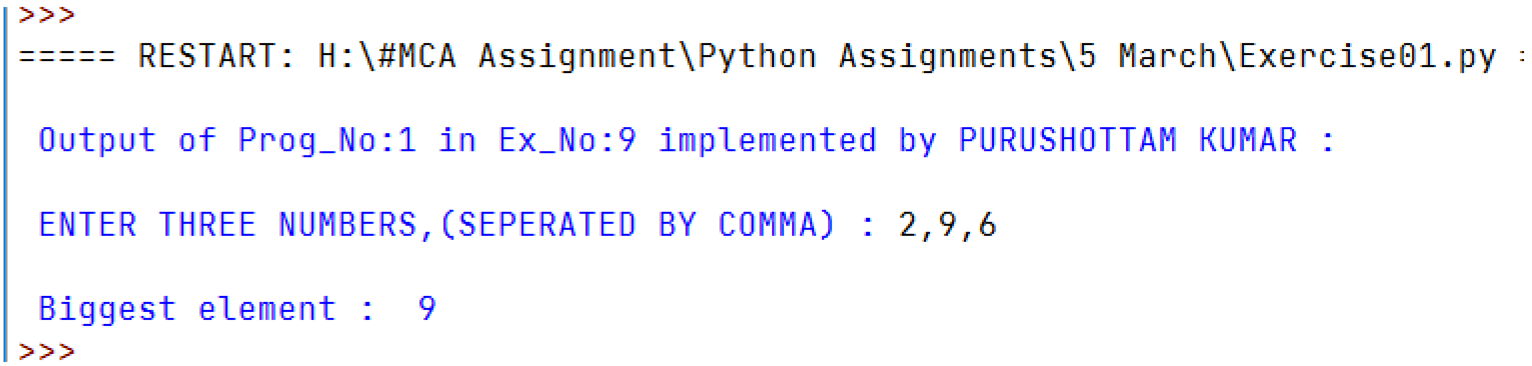
Assignment on Python Functions and Modules

1. Write a program that finds the greatest of three given numbers using functions. Pass the numbers as arguments.


Exercise01.py - H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py (3.9.1)
File Edit Format Run Options Window Help

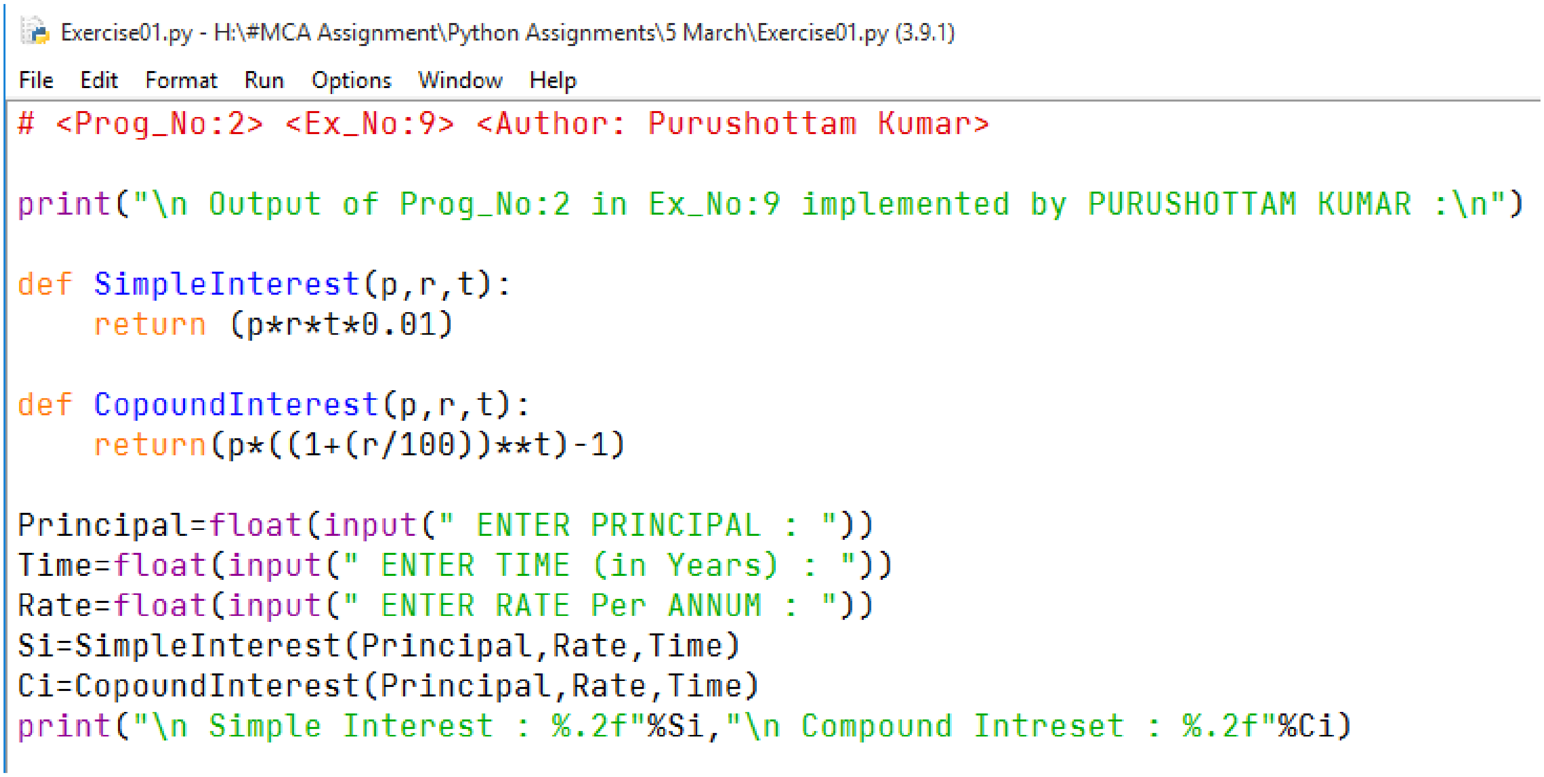
```python
# <Prog_No:1> <Ex_No:9> <Author: Purushottam Kumar>

print("\n Output of Prog_No:1 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def biggestThree(a,b,c):
    if(a>b and a>c):
        return a
    elif b>c:
        return b
    else:
        return c

num1,num2,num3=[int(e) for e in input(" ENTER THREE NUMBERS,(SEPERATED BY COMMA) : ").split(',')]
Max=biggestThree(num1,num2,num3)
print("\n Biggest element : ",Max)
```

OUTPUT



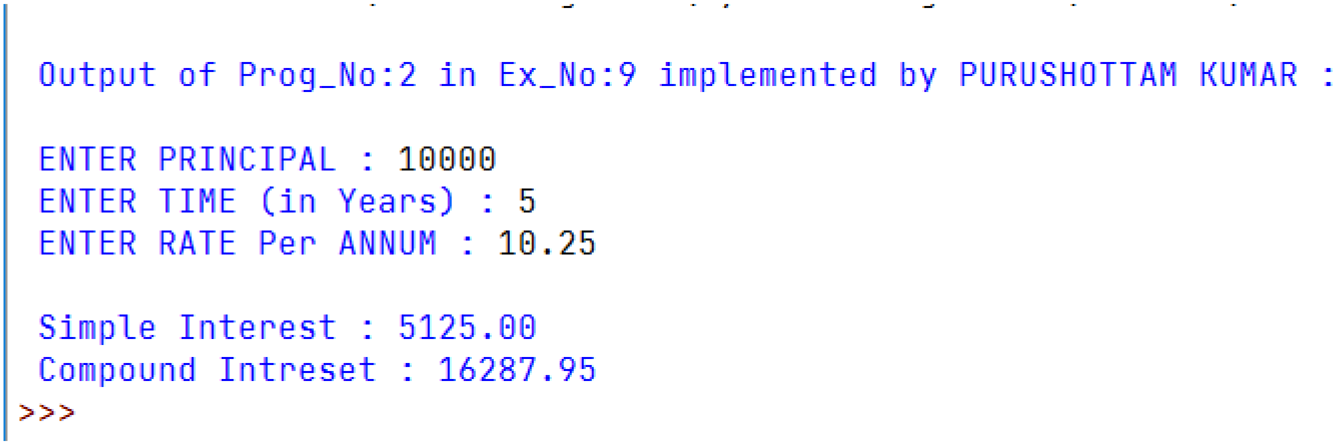
```
>>>
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py

 Output of Prog_No:1 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

 ENTER THREE NUMBERS,(SEPERATED BY COMMA) : 2,9,6

 Biggest element :  9
>>>
```

2.Write two functions simple_interest() and compound_interest() that returns the simple and compound interest if relevant arguments are passed to the functions.


Exercise01.py - H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py (3.9.1)

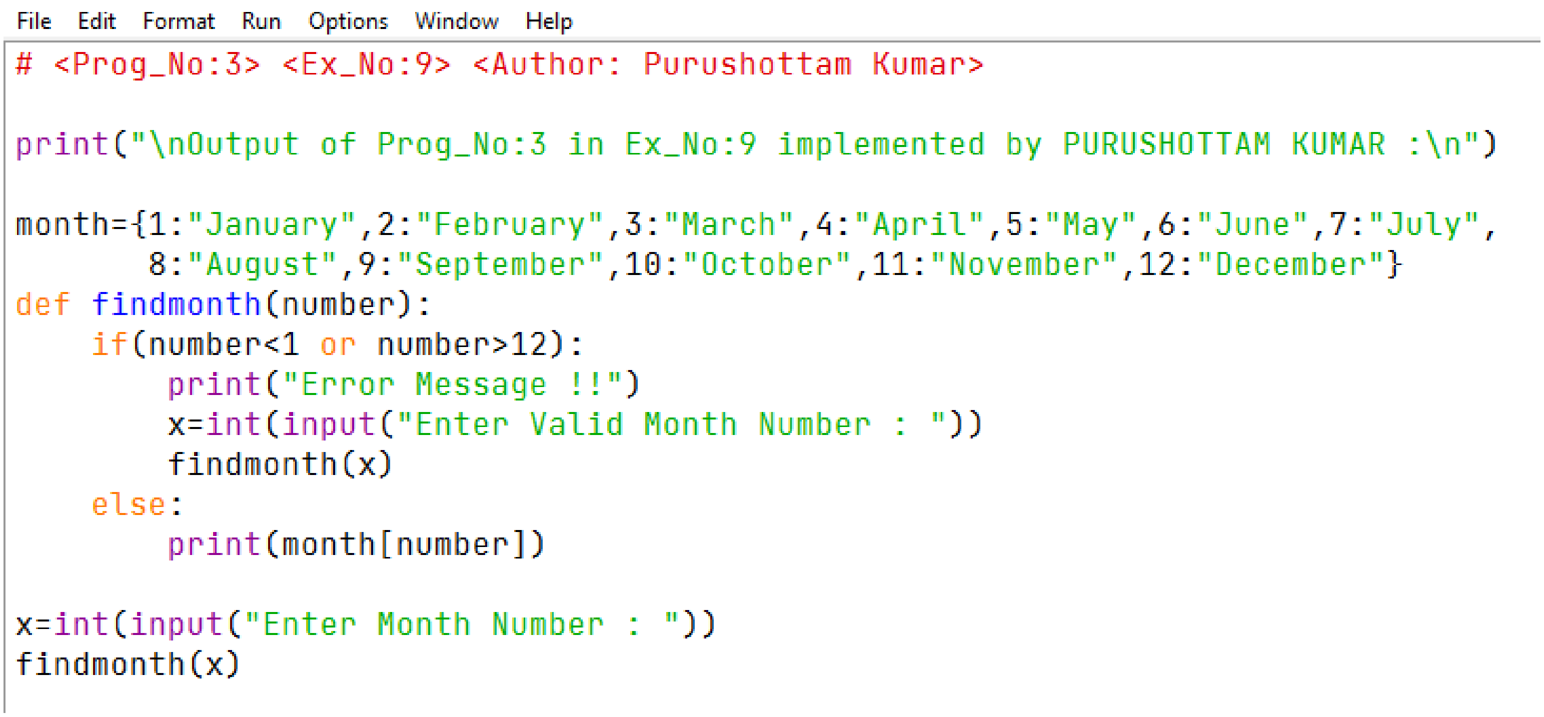
```python
# <Prog_No:2> <Ex_No:9> <Author: Purushottam Kumar>

print("\n Output of Prog_No:2 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def SimpleInterest(p,r,t):
    return (p*r*t*0.01)

def CopoundInterest(p,r,t):
    return(p*((1+(r/100))**t)-1)

Principal=float(input(" ENTER PRINCIPAL : "))
Time=float(input(" ENTER TIME (in Years) : "))
Rate=float(input(" ENTER RATE Per ANNUM : "))
Si=SimpleInterest(Principal,Rate,Time)
Ci=CopoundInterest(Principal,Rate,Time)
print("\n Simple Interest : %.2f"%Si,"\n Compound Intreset : %.2f"%Ci)
```

OUTPUT



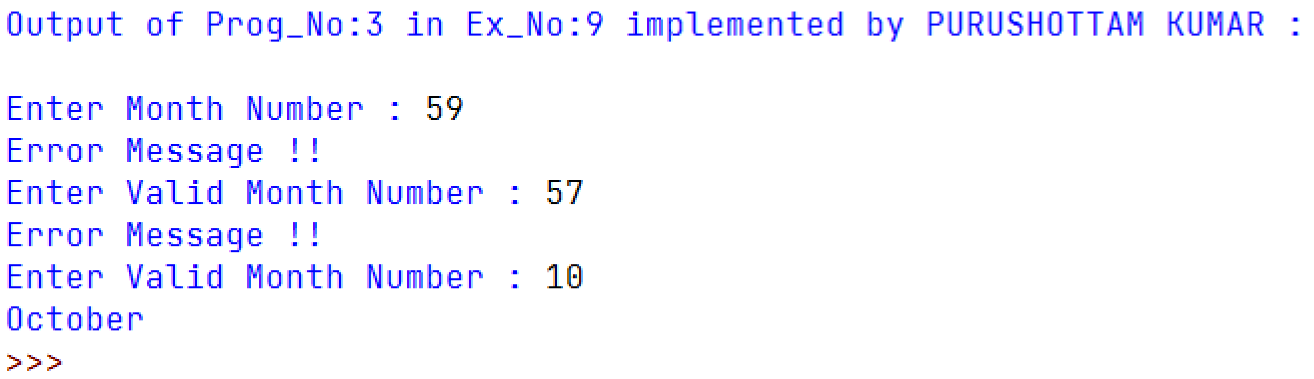
```
 Output of Prog_No:2 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

 ENTER PRINCIPAL : 10000
 ENTER TIME (in Years) : 5
 ENTER RATE Per ANNUM : 10.25

 Simple Interest : 5125.00
 Compound Intreset : 16287.95
>>>
```

3. Write a function accepts an integer between 1 and 12 to represent the month number and displays the corresponding month of the year (For example, if month = 1, then display JANUARY). If the function receives any value which is not in the range 1 to 12 display error message and prompt for correct input.



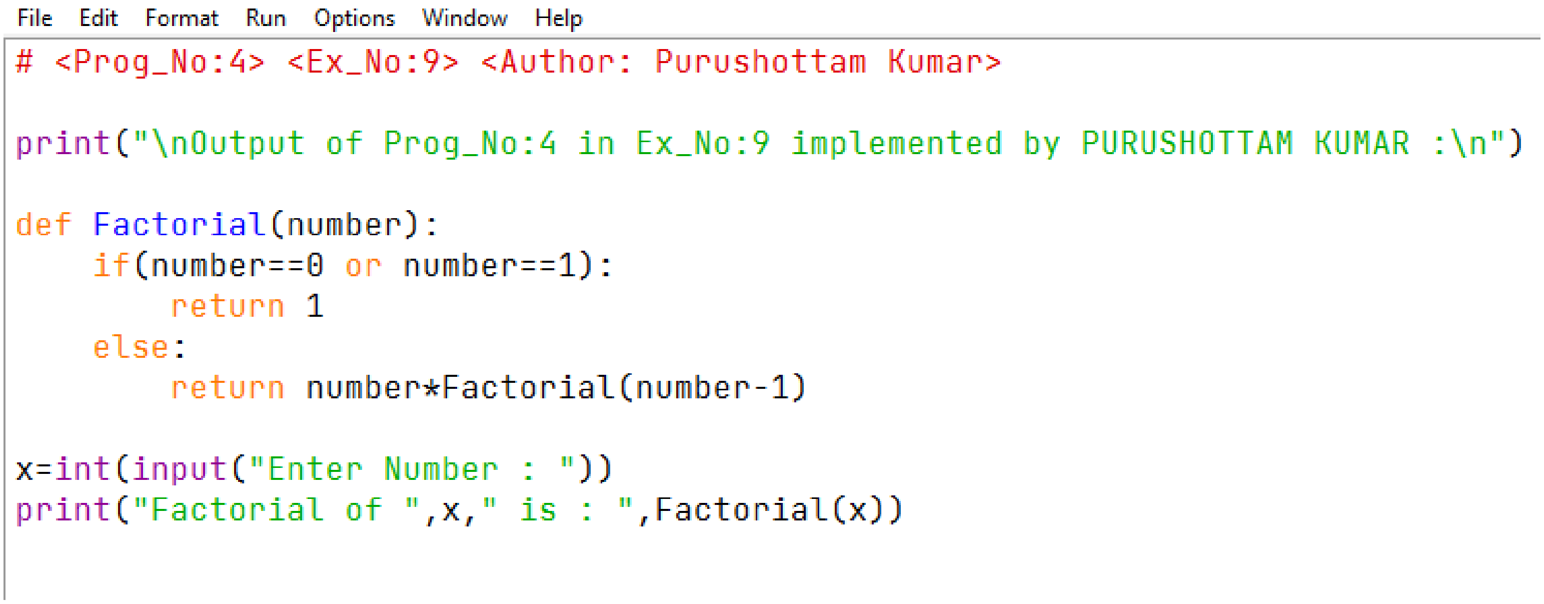
```
# <Prog_No:3> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:3 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

month={1:"January",2:"February",3:"March",4:"April",5:"May",6:"June",7:"July",
       8:"August",9:"September",10:"October",11:"November",12:"December"}
def findmonth(number):
    if(number<1 or number>12):
        print("Error Message !!")
        x=int(input("Enter Valid Month Number : "))
        findmonth(x)
    else:
        print(month[number])

x=int(input("Enter Month Number : "))
findmonth(x)
```

OUTPUT



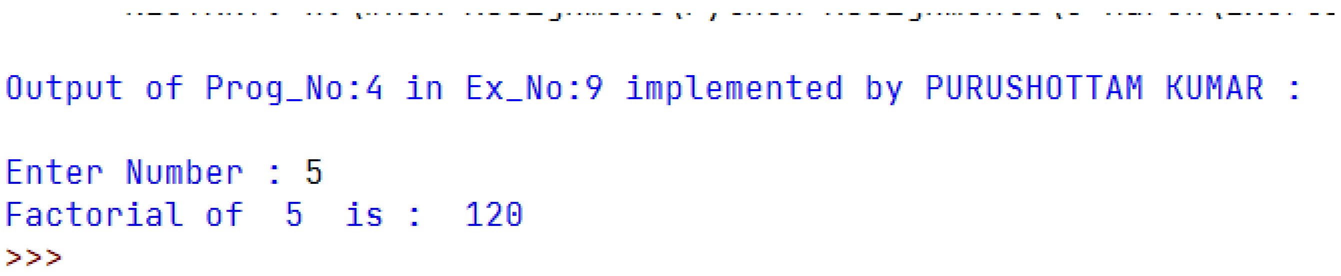
```
Output of Prog_No:3 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter Month Number : 59
Error Message !!
Enter Valid Month Number : 57
Error Message !!
Enter Valid Month Number : 10
October
>>>
```

4. Write a program to calculate the factorial of a given number using function and recursion.



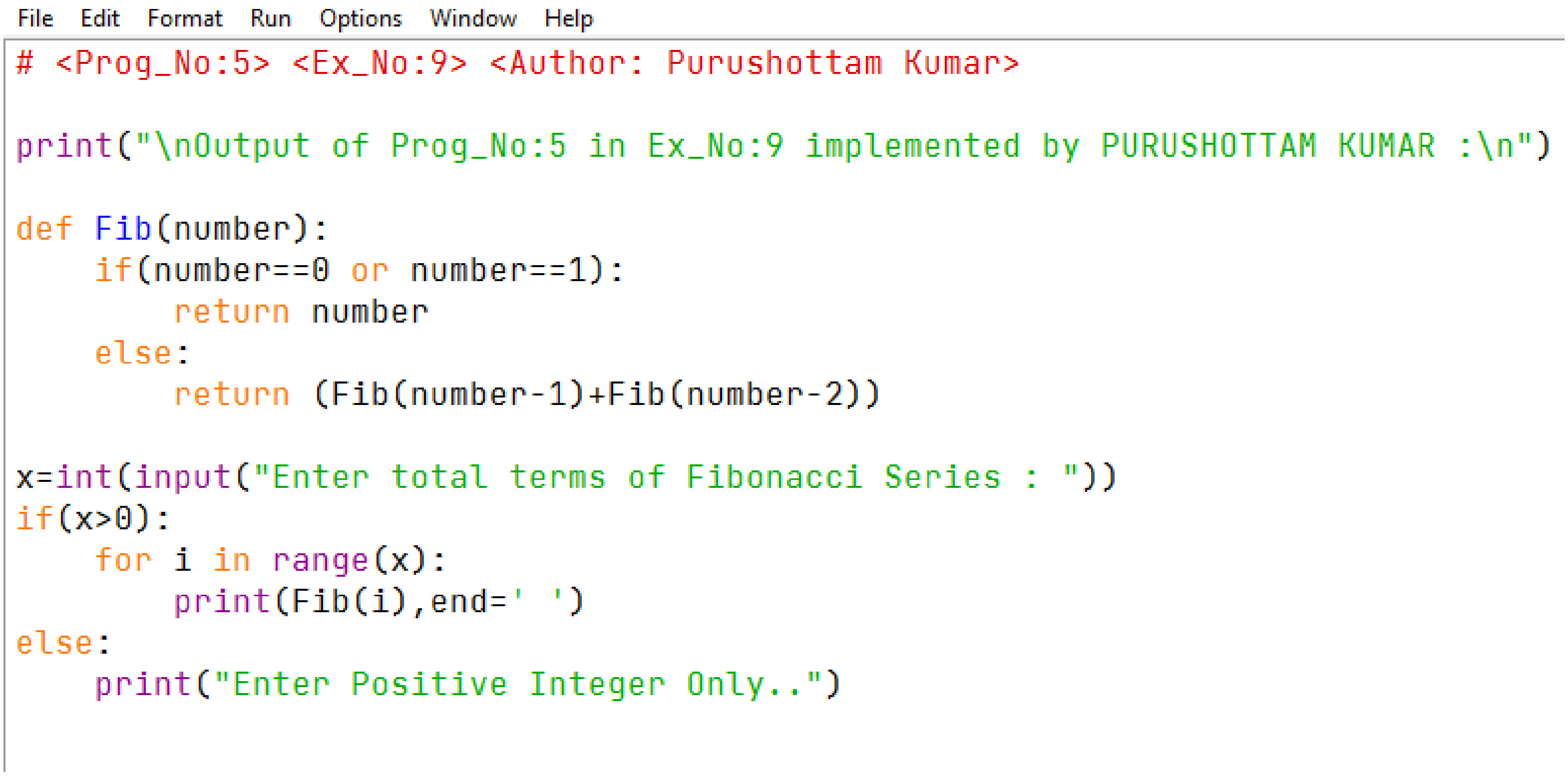
```python
# <Prog_No:4> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:4 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def Factorial(number):
    if(number==0 or number==1):
        return 1
    else:
        return number*Factorial(number-1)

x=int(input("Enter Number : "))
print("Factorial of ",x," is : ",Factorial(x))
```

OUTPUT



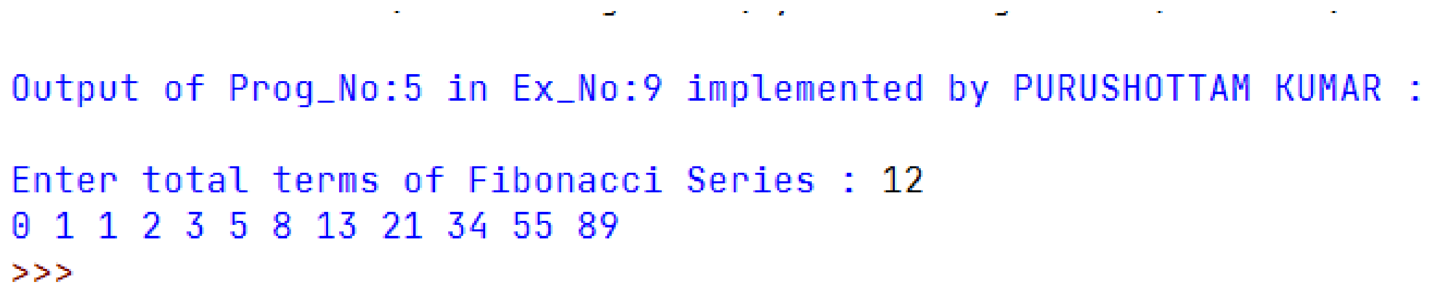
```
Output of Prog_No:4 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter Number : 5
Factorial of  5  is :  120
>>>
```

5. Write a program to generate the Fibonacci series of a given numbers using function and recursion.



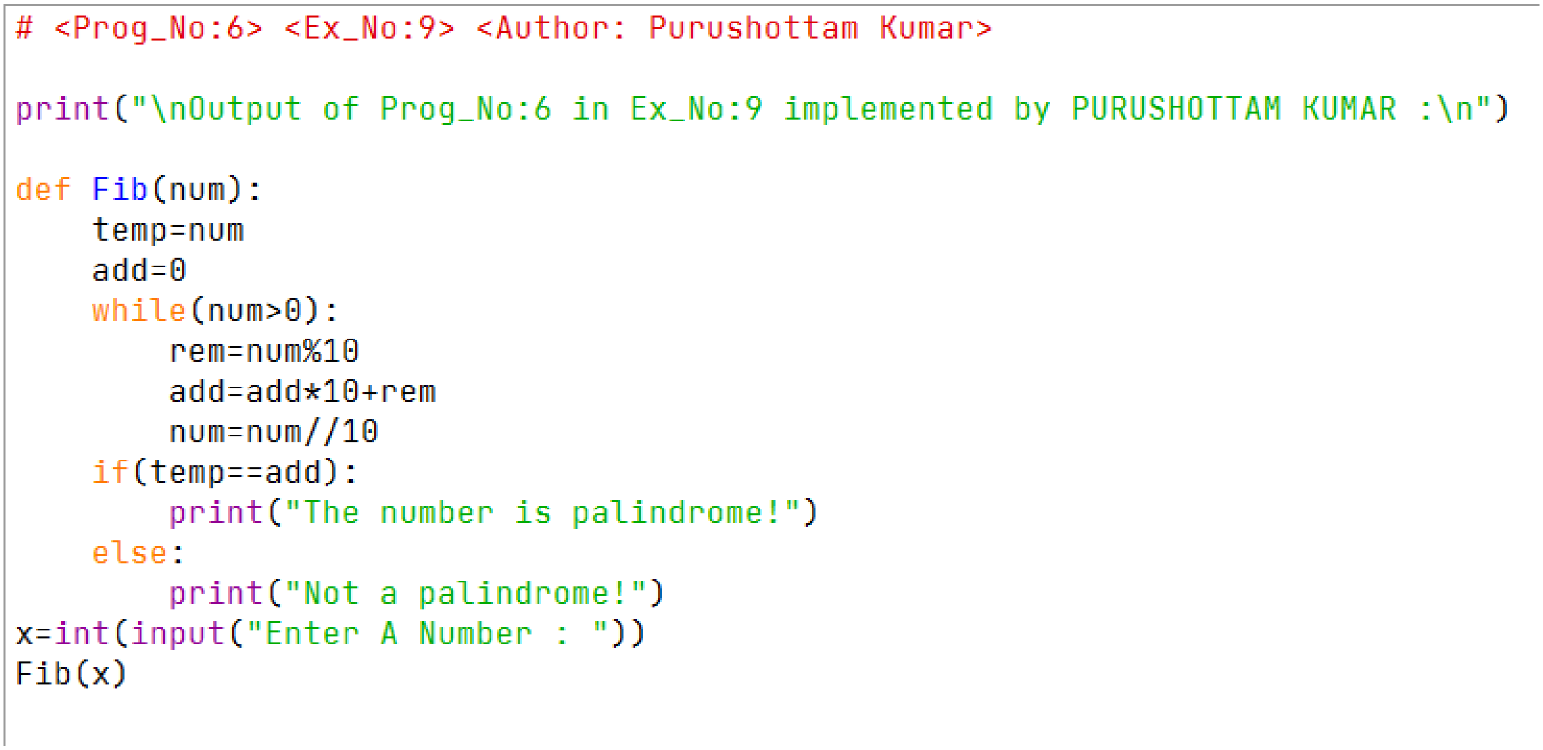
```python
# <Prog_No:5> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:5 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def Fib(number):
    if(number==0 or number==1):
        return number
    else:
        return (Fib(number-1)+Fib(number-2))

x=int(input("Enter total terms of Fibonacci Series : "))
if(x>0):
    for i in range(x):
        print(Fib(i),end=' ')
else:
    print("Enter Positive Integer Only..")
```

OUTPUT



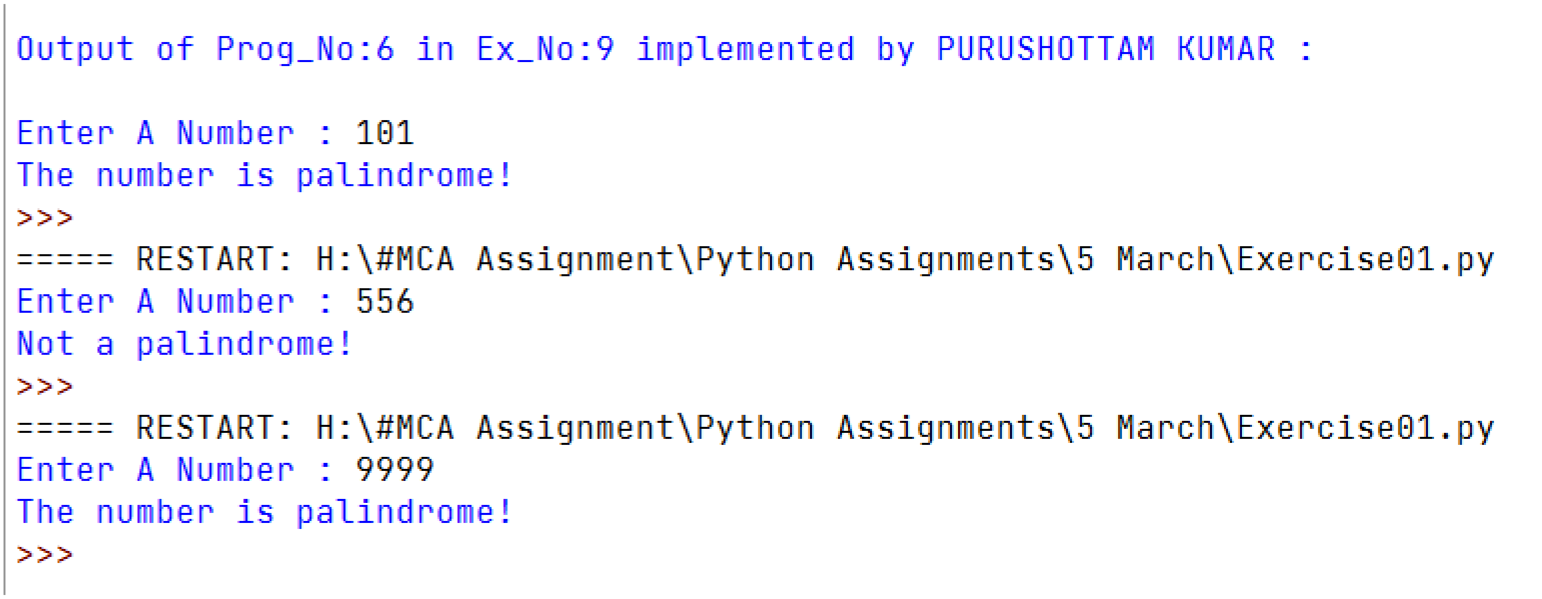
```
Output of Prog_No:5 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter total terms of Fibonacci Series : 12
0 1 1 2 3 5 8 13 21 34 55 89
>>>
```

6. Write a function that receives a number and checks whether it is a palindrome or not.



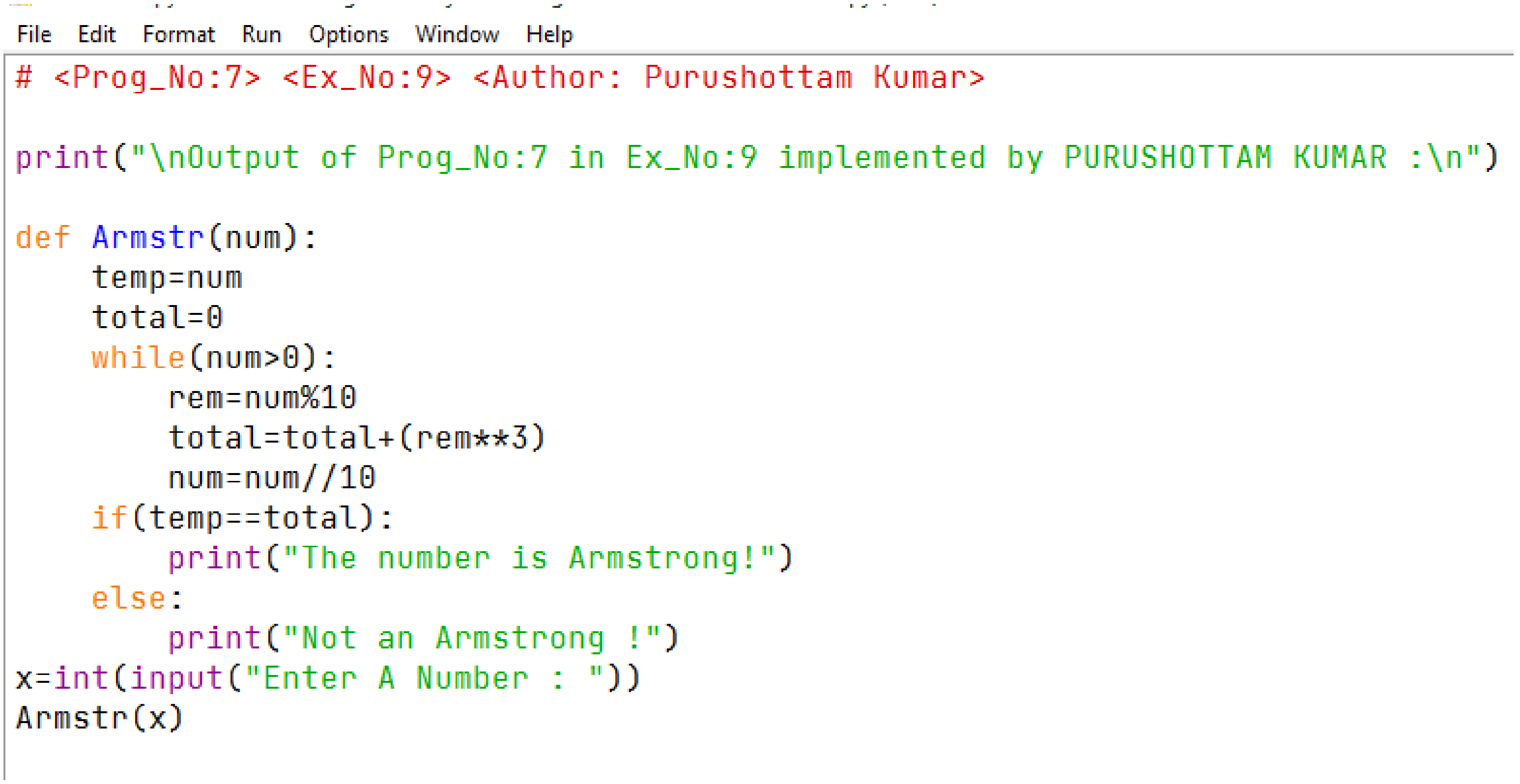
```
# <Prog_No:6> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:6 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def Fib(num):
    temp=num
    add=0
    while(num>0):
        rem=num%10
        add=add*10+rem
        num=num//10
    if(temp==add):
        print("The number is palindrome!")
    else:
        print("Not a palindrome!")
x=int(input("Enter A Number : "))
Fib(x)
```

OUTPUT



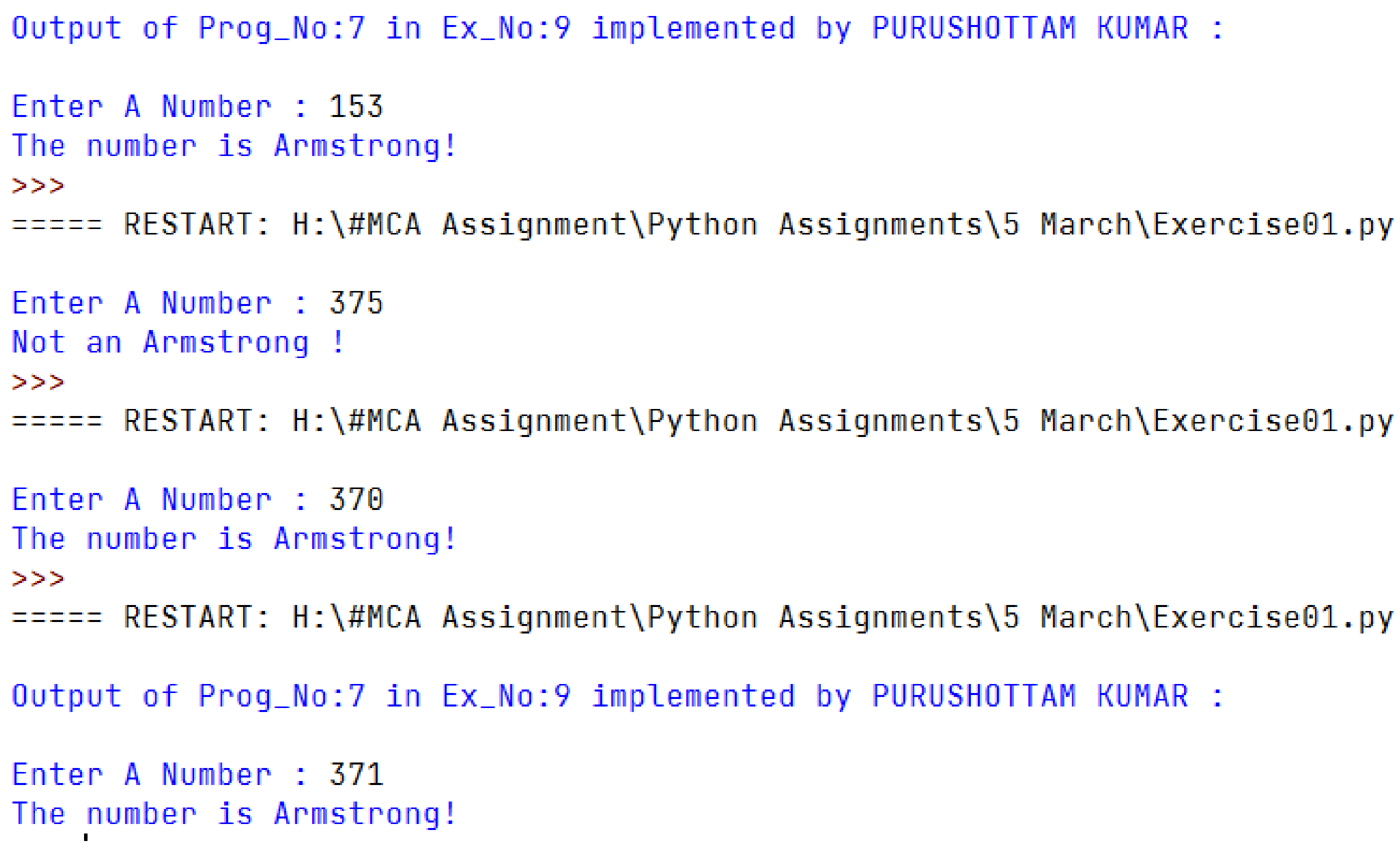
```
Output of Prog_No:6 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter A Number : 101
The number is palindrome!
>>> 
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py
Enter A Number : 556
Not a palindrome!
>>> 
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py
Enter A Number : 9999
The number is palindrome!
>>> 
```

7. Write a function that receives a number and checks whether it is an Armstrong number or not.



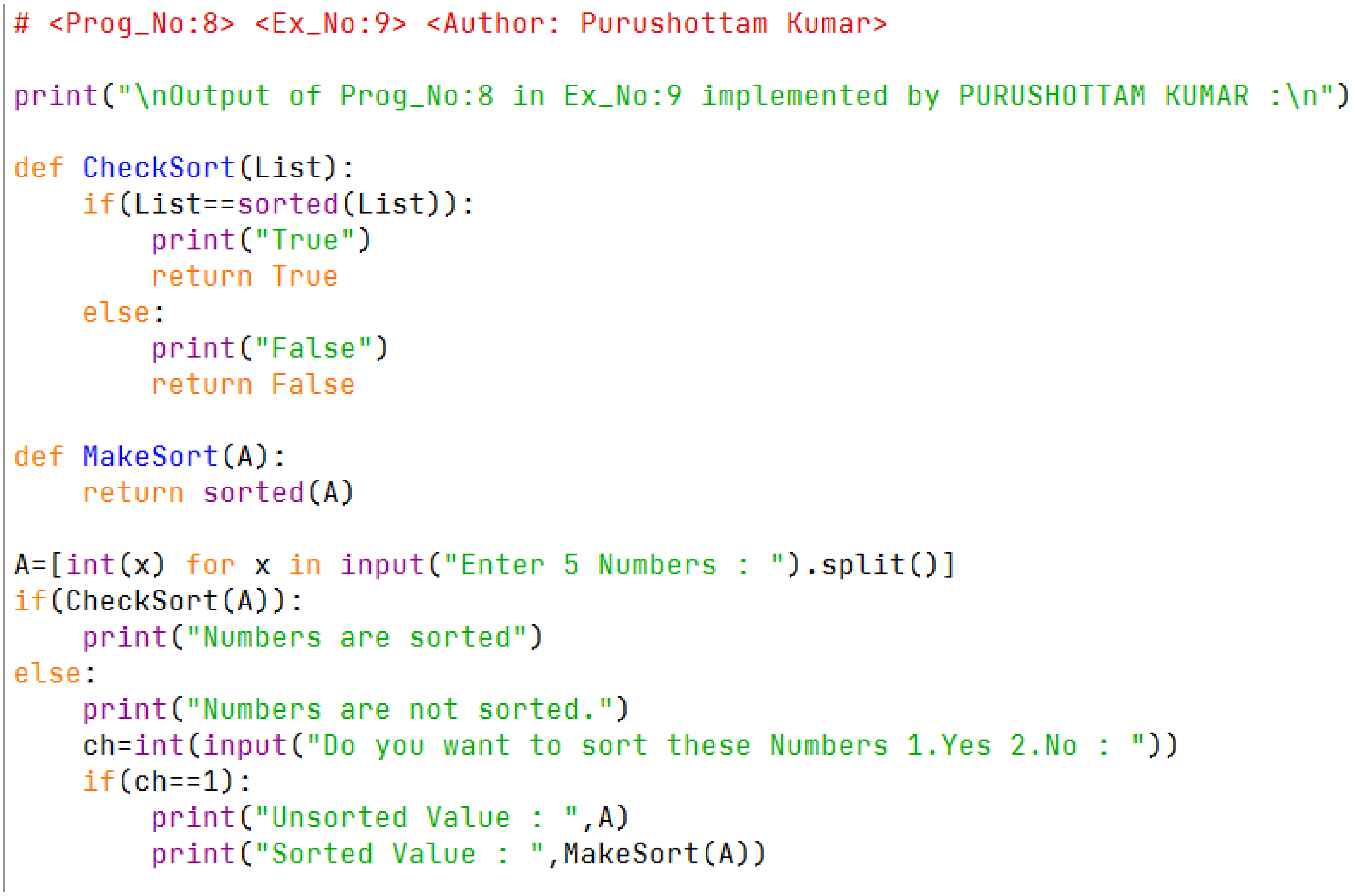
```
# <Prog_No:7> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:7 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def Armstr(num):
    temp=num
    total=0
    while(num>0):
        rem=num%10
        total=total+(rem**3)
        num=num//10
    if(temp==total):
        print("The number is Armstrong!")
    else:
        print("Not an Armstrong !")
x=int(input("Enter A Number : "))
Armstr(x)
```

OUTPUT



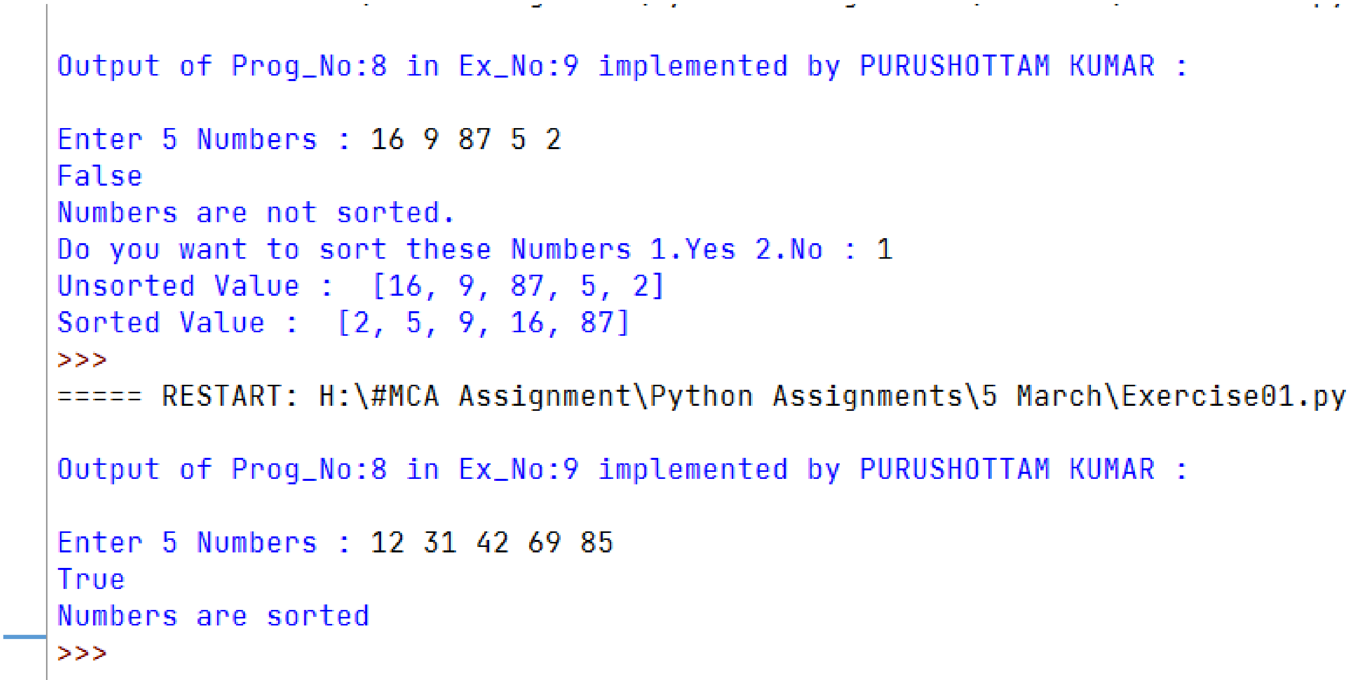
```
Output of Prog_No:7 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter A Number : 153
The number is Armstrong!
>>>
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py

Enter A Number : 375
Not an Armstrong !
>>>
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py

Enter A Number : 370
The number is Armstrong!
>>>
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py

Output of Prog_No:7 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter A Number : 371
The number is Armstrong!
```

8. Write a function that accepts 5 integers, and returns TRUE if they are sorted, otherwise it returns false and prompts the user WHETHER TO SORT THE NUMBERS OR NOT. If the user enters YES, then another function must sort the 5 numbers and return the sorted series.



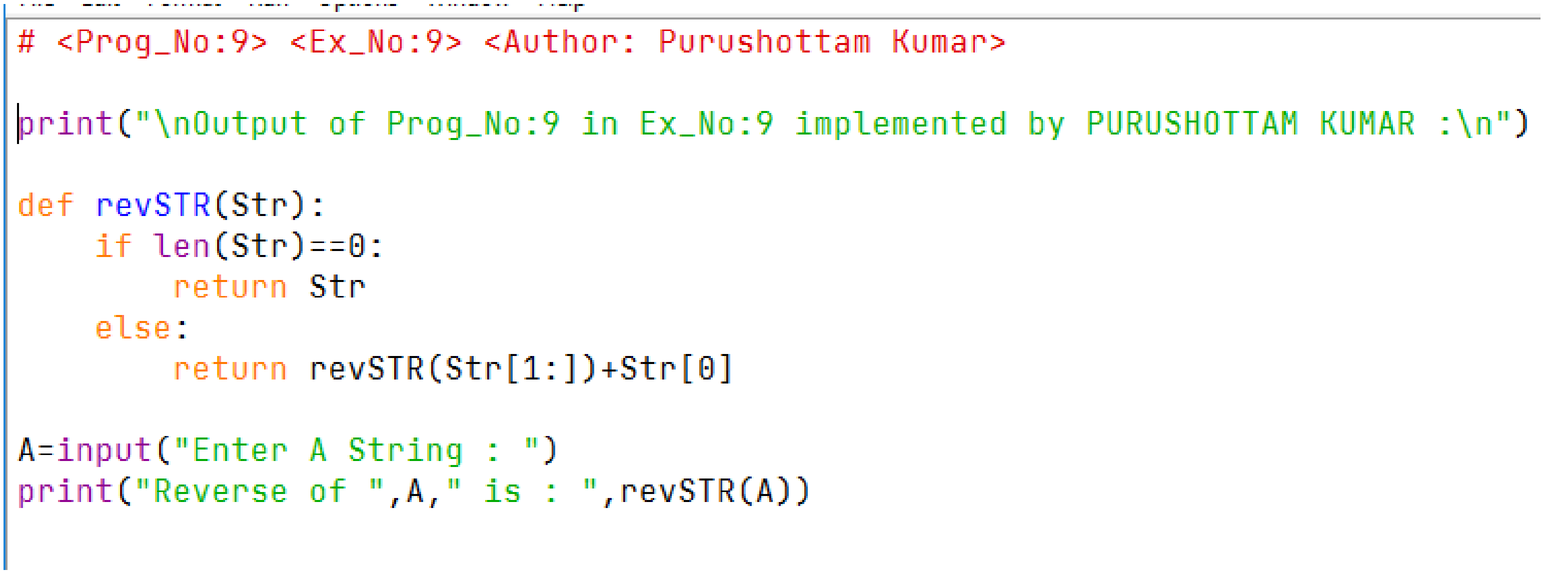
```python
# <Prog_No:8> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:8 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def CheckSort(List):
    if(List==sorted(List)):
        print("True")
        return True
    else:
        print("False")
        return False

def MakeSort(A):
    return sorted(A)

A=[int(x) for x in input("Enter 5 Numbers : ").split()]
if(CheckSort(A)):
    print("Numbers are sorted")
else:
    print("Numbers are not sorted.")
    ch=int(input("Do you want to sort these Numbers 1.Yes 2.No : "))
    if(ch==1):
        print("Unsorted Value : ",A)
        print("Sorted Value : ",MakeSort(A))
```

OUTPUT



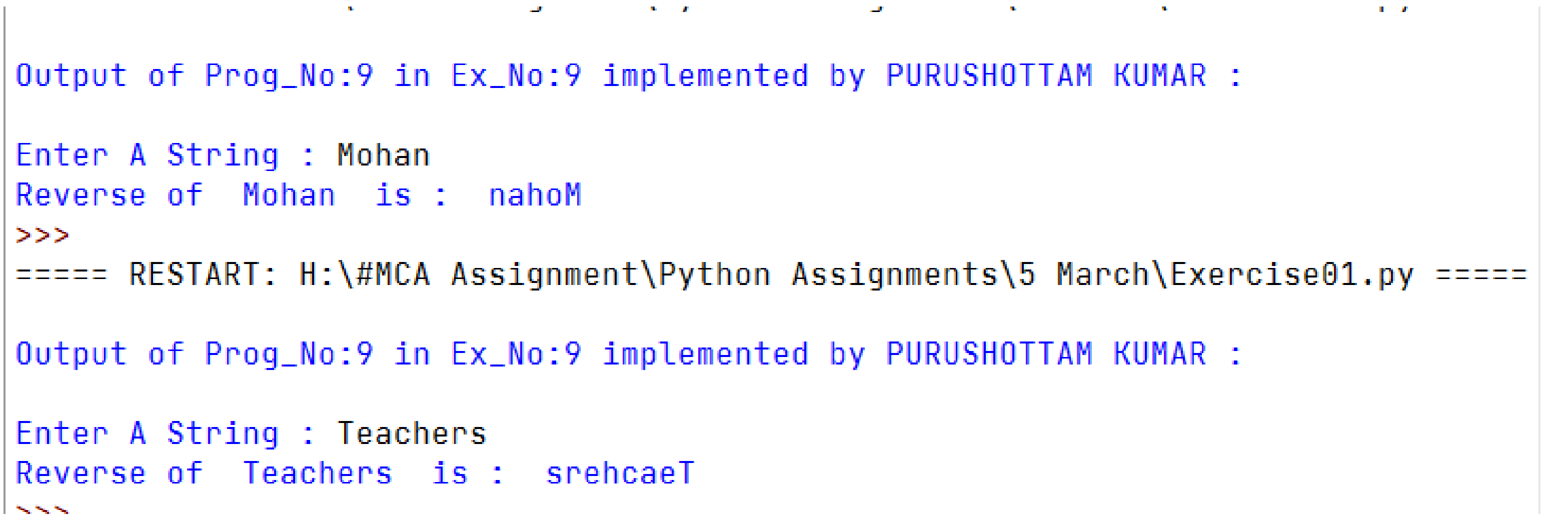
```
Output of Prog_No:8 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter 5 Numbers : 16 9 87 5 2
False
Numbers are not sorted.
Do you want to sort these Numbers 1.Yes 2.No : 1
Unsorted Value :  [16, 9, 87, 5, 2]
Sorted Value :  [2, 5, 9, 16, 87]
>>> 
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py

Output of Prog_No:8 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter 5 Numbers : 12 31 42 69 85
True
Numbers are sorted
>>> 
```

9. Write a program to reverse a string using function and recursion.



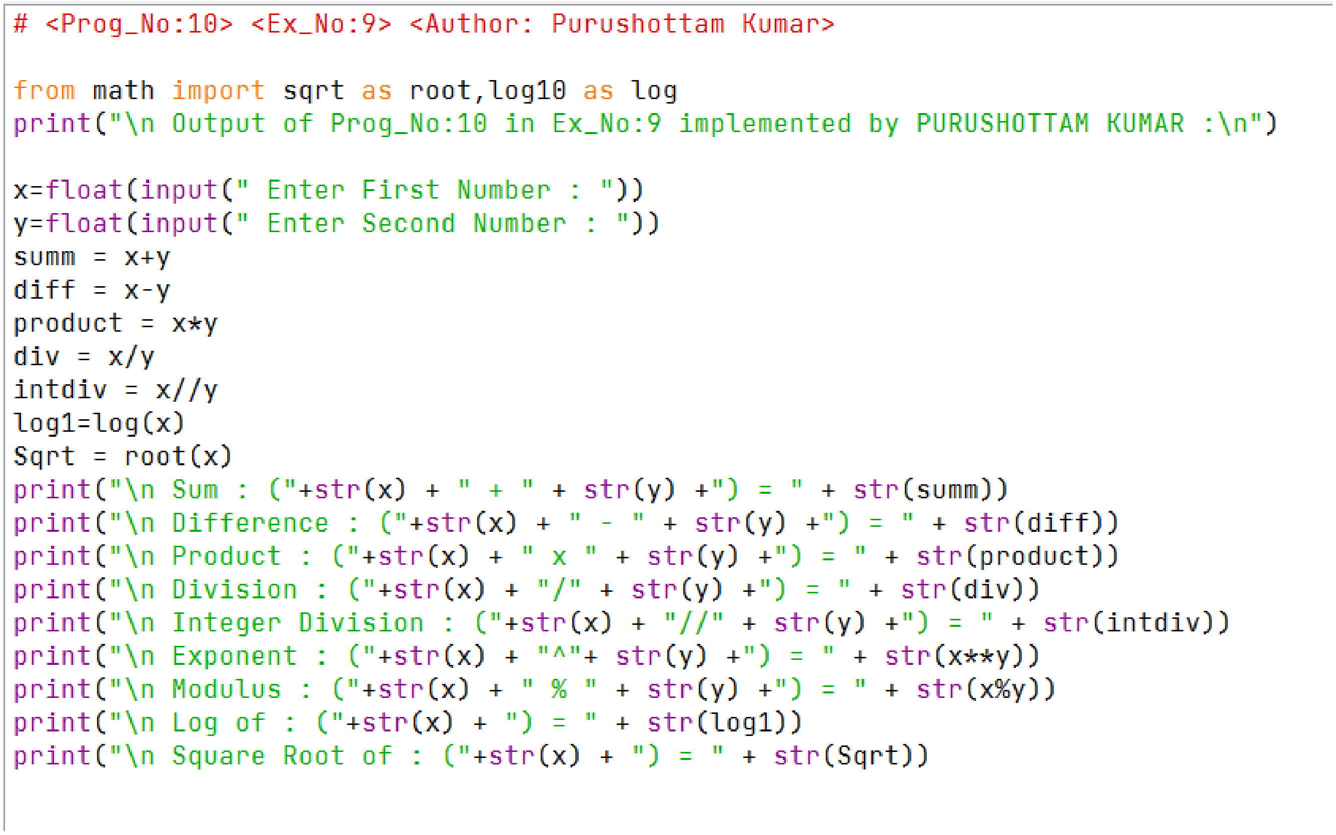
```
# <Prog_No:9> <Ex_No:9> <Author: Purushottam Kumar>

print("\nOutput of Prog_No:9 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

def revSTR(Str):
    if len(Str)==0:
        return Str
    else:
        return revSTR(Str[1:])+Str[0]

A=input("Enter A String : ")
print("Reverse of ",A," is : ",revSTR(A))
```

OUTPUT



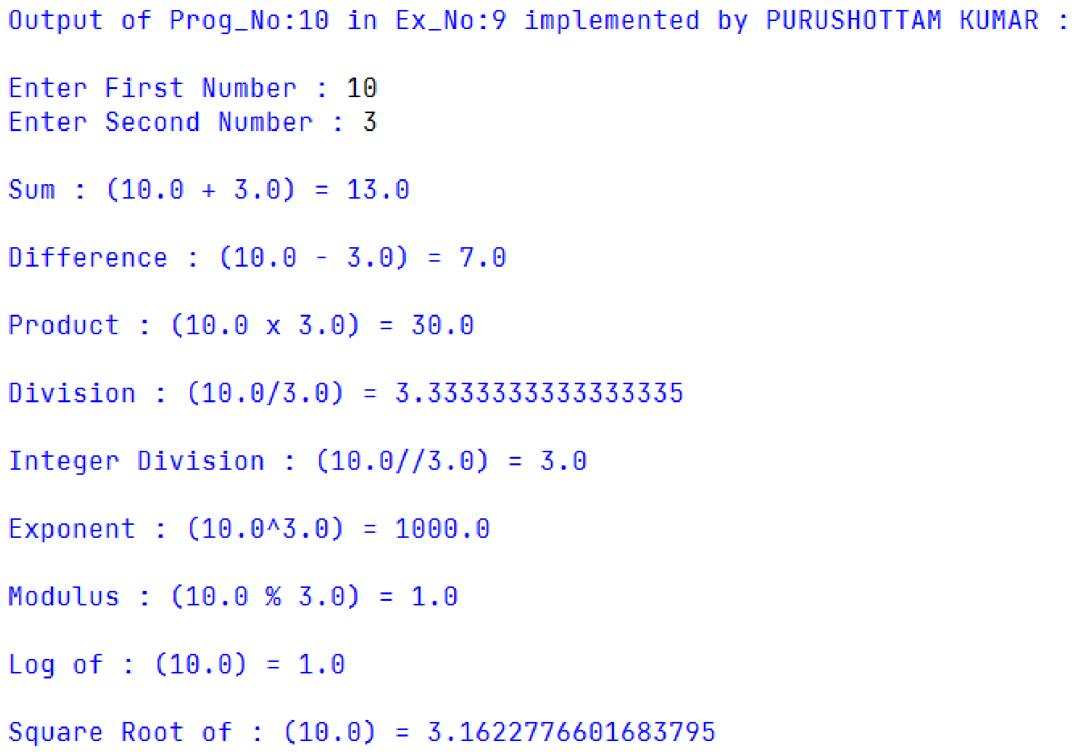
```
Output of Prog_No:9 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter A String : Mohan
Reverse of  Mohan  is :  nahoM
>>>
===== RESTART: H:\#MCA Assignment\Python Assignments\5 March\Exercise01.py =====

Output of Prog_No:9 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

Enter A String : Teachers
Reverse of  Teachers  is :  srehcaeT
>>>
```

10. Write a menu driven program using functions to perform basic arithmetic operations such as addition, subtraction, multiplication, division, modulo, to the power of, log and square root.



```python
# <Prog_No:10> <Ex_No:9> <Author: Purushottam Kumar>

from math import sqrt as root,log10 as log
print("\n Output of Prog_No:10 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :\n")

x=float(input(" Enter First Number : "))
y=float(input(" Enter Second Number : "))
summ = x+y
diff = x-y
product = x*y
div = x/y
intdiv = x//y
log1=log(x)
Sqrt = root(x)
print("\n Sum : ("+str(x) + " + " + str(y) +") = " + str(summ))
print("\n Difference : ("+str(x) + " - " + str(y) +") = " + str(diff))
print("\n Product : ("+str(x) + " x " + str(y) +") = " + str(product))
print("\n Division : ("+str(x) + "/" + str(y) +") = " + str(div))
print("\n Integer Division : ("+str(x) + "//" + str(y) +") = " + str(intdiv))
print("\n Exponent : ("+str(x) + "^"+ str(y) +") = " + str(x**y))
print("\n Modulus : ("+str(x) + " % " + str(y) +") = " + str(x%y))
print("\n Log of : ("+str(x) + ") = " + str(log1))
print("\n Square Root of : ("+str(x) + ") = " + str(Sqrt))
```

OUTPUT



```
 Output of Prog_No:10 in Ex_No:9 implemented by PURUSHOTTAM KUMAR :

 Enter First Number : 10
 Enter Second Number : 3

 Sum : (10.0 + 3.0) = 13.0

 Difference : (10.0 - 3.0) = 7.0

 Product : (10.0 x 3.0) = 30.0

 Division : (10.0/3.0) = 3.3333333333333335

 Integer Division : (10.0//3.0) = 3.0

 Exponent : (10.0^3.0) = 1000.0

 Modulus : (10.0 % 3.0) = 1.0

 Log of : (10.0) = 1.0

 Square Root of : (10.0) = 3.1622776601683795
```