Q1a Sum over first N natural numbers

In [121…

```python
#Sum over first N natural numbers#

def Sum(N):                                                           # N is the number of terms required#
    if N < 1 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number.")        #checking if N is not a natural number#

    else:
        S = 0
        for i in range(1,int(N)+1):
            S = S + i
        print("The sum of first" , int(N) , "natural numbers is =", S)

Sum(16)
Sum(100)
Sum(2.5)
Sum(-8)
Sum(36)
Sum(0)
```

```
The sum of first 16 natural numbers is = 136
The sum of first 100 natural numbers is = 5050
The entered value is not a natural number.
The entered value is not a natural number.
The sum of first 36 natural numbers is = 666
The entered value is not a natural number.
```

Q1b Sum over first N odd natural numbers

In [122…

```python
# Sum over first N odd natural numbers#

def odd_sum(N):                                                        # N is the number of terms required#
    if N < 1 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number.")         #checking if N is not a natural number#
    else:
        S = 0
        j = 1
        for i in range(1,int(N)+1):
            S = S + j
            i = i + 1
            j = j + 2
        print("The sum of first" , int(N) , "odd natural numbers is =", S)

odd_sum(4)
odd_sum(0)
odd_sum(-10)
odd_sum(5.9)
odd_sum(9)
```

```
The sum of first 4 odd natural numbers is = 16
The entered value is not a natural number.
The entered value is not a natural number.
The entered value is not a natural number.
The sum of first 9 odd natural numbers is = 81
```

Q2a Sum over the first N terms on an AP with common difference = 1.5

In [123…

```python
# Sum over the first N terms on an AP with common difference = 1.5 #

def Sum_AP(a_1,N):                                                          # N is the number of terms required ar
    if N < 1 or N > 0 and N - int(N)!= 0 :                                  #checking if N is not a natural numbe
        print("The entered value is not a natural number.")

    else:
        S = 0
        j = a_1                                                              #taking first term of the AP#
        for i in range(1,int(N)+1):
            S = S + j
            i = i + 1
            j = j + 1.5                                                     #given the common difference of the A
        print("The sum of first" , int(N) , " terms of an AP with initial term" , float(a_1) ,"and common diffe

Sum_AP(1.5,7)
Sum_AP(2,-5)
Sum_AP(3,0)
Sum_AP(1,100)
Sum_AP(5,7.5)
```

```
The sum of first 7  terms of an AP with initial term 1.5 and common difference of 1.5 is = 42.0
The entered value is not a natural number.
The entered value is not a natural number.
The sum of first 100  terms of an AP with initial term 1.0 and common difference of 1.5 is = 7525.0
The entered value is not a natural number.
```

Q2b Sum of first N terms of a GP with common ration = 0.5

In [124…

```python
# Sum of first N terms of a GP with common ration = 0.5 #

def Sum_GP(a_1,N):                                                                                        # N is the nu
    if N < 1 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number. Please enter a natural number.")        #checking if N

    else:
        S = 0
        j = a_1                                          #first term of GP#
        for i in range(1,int(N)+1):
            S = S + j
            i = i + 1
            j = j*0.5                                   #common ratio of GP taken to be 0.5#
        print("The sum of first" , int(N) , " terms of an GP with initial term" , float(a_1) ,"and with common

Sum_GP(1,4)
Sum_GP(2,0)
Sum_GP(2,-5)
Sum_GP(1.5,7.5)
Sum_GP(1.5,25)
```

```
The sum of first 4  terms of an GP with initial term 1.0 and with common ratio of 0.5 is = 1.875
The entered value is not a natural number. Please enter a natural number.
The entered value is not a natural number. Please enter a natural number.
The entered value is not a natural number. Please enter a natural number.
The sum of first 25  terms of an GP with initial term 1.5 and with common ratio of 0.5 is = 2.999999910593033
```

Q2c Sum over the first N terms on an HP with common difference = 1.5

In [125…

```python
# Sum over the first N terms on an HP with common difference = 1.5 #

def Sum_HP(a_1,N):                                               # N is the number of terms required and a_1 is the
    if N < 1 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number. Please enter a natural number.")

    else:
        S = 0
        j = a_1                                          #The first term of HP#
        for i in range(0,int(N)):
            S = S + 1/j
            i = i + 1
            j = j + 1.5                                 #the common difference of HP is given as 1.5#
        print("The sum of first" , int(N) , " terms of an HP with initial term" , float(a_1) ,"and with common

Sum_HP(2,0)
Sum_HP(3,-5)
Sum_HP(1,20)
Sum_HP(2,7.5)
Sum_HP(1.5,4)
```

```
The entered value is not a natural number. Please enter a natural number.
The entered value is not a natural number. Please enter a natural number.
The sum of first 20  terms of an HP with initial term 1.0 and with common difference of 1.5 is equal = 2.881578
6000970864
The entered value is not a natural number. Please enter a natural number.
The sum of first 4  terms of an HP with initial term 1.5 and with common difference of 1.5 is equal = 1.3888888
88888889
```

Q3 Factorial of Natural number N

In [126…

```python
#To find  Factorial of Natural number N#



def Fact(N):
    if N < 0 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number.")          #checking if N is not a natural number#
    else:
        fact = 1
        for i in range(1,int(N)+1):
            fact = fact * i
            i = i + 1
        print("The factorial of", int(N) ," is equal to",fact)


Fact(0)
Fact(4)
Fact(-0.5)
Fact(-5)
Fact(3.4)
Fact(12)
```

```
The factorial of 0  is equal to 1
The factorial of 4  is equal to 24
The entered value is not a natural number.
The entered value is not a natural number.
The entered value is not a natural number.
The factorial of 12  is equal to 479001600
```

Q4 To calculate sin(x) and exp(−x) accurate up to 4 place in decimal and to plot modulus of error versus iteration numbers

In [127…

```python
# importing "math" for mathematical operations

import math

import matplotlib.pyplot as plt

def Fact(N):
    if N < 0 or N > 0 and N - int(N)!= 0 :
        print("The entered value is not a natural number.")          #checking if N is not a natural number#
    else:
        fact = 1
        for i in range(1,int(N)+1):
            fact = fact * i
            i = i + 1
        return fact

#function to calculate sin(x) using Taylor Expansion#

def sin(x):
    sin_x = 0
    i = 0
    while abs((sin_x) - math.sin(x)) > 0.00001 :
        temp = ((-1)**i)*(x**(2*i+1))/(Fact(2*i+1))
        sin_x += temp
        i = i+1

    print ("sin("+ str(x) + ") is " + str(math.floor(sin_x*10**4)/10**4))


#function to calculate e^(-x) using Taylor expansion#

def eminusx(x):
    e_minusx =0
    m = 0
    while abs((e_minusx)- math.exp(-x)) > 0.00001:
        temp_2 = (((-1)**m)*(x**m))/Fact(m)
        e_minusx += temp_2
        m = m + 1
    print ("e^[-("+ str(x) + ")] is " + str(math.floor(e_minusx*10**4)/10**4))

sin(1)
sin(25)
sin(0)
sin(-5)
sin(2.5)

print('\n')
eminusx(2)
eminusx(4)
eminusx(6.7)
eminusx(-5)
eminusx(0)
print('\n')


#for graph


#graph for sin(x)

#plotting the error for sin(25 radians) against the iterative numbers


singraph = []
i = 0
sin_a = 0

while abs((sin_a) - math.sin(25)) > 0.00001 :
    temp3 = ((-1)**i)*(25**(2*i+1))/(Fact(2*i+1))
    sin_a += temp3
    i = i+1
    singraph.append(float(abs((sin_a) - (math.sin(25)))))

print ("Modulus of error for sin(25) is " + str(singraph))
print('\n')

#graph for e^(-x)

#plotting the errorr for e^(-4) against the iterative numbers

expgraph = []
j = 0
e_minusb = 0

while abs((e_minusb)- math.exp(-4)) > 0.00001:
    temp2 = (((-1)**j)*((4)**j))/Fact(j)
    e_minusb += temp2
    j = j + 1
    expgraph.append(float(abs((e_minusb)- math.exp(-4))))

print("Modulus of error for e^(-4) is " + str(expgraph))


plt.plot(range(i), singraph, '-o')


plt.xlabel("No. Iteration")
plt.ylabel("|S_n(x)-sin(x)| for x=25 radians")
pt.title("Error in sin(x) V/s No. Iteration")
plt.show()


plt.plot(range(j), expgraph, '-o')


plt.xlabel("No. Iteration")
plt.ylabel("|e_n(x)-e^{-x}| for x=-4")
pt.title("Error in e^-x V/s No. Iteration")
plt.show()
```

```
sin(1) is 0.8414
sin(25) is -0.1324
sin(0) is 0.0
sin(-5) is 0.9589
sin(2.5) is 0.5984


e^[-(2)] is 0.1353
e^[-(4)] is 0.0183
e^[-(6.7)] is 0.0012
e^[-(-5)] is 148.4131
e^[-(0)] is 1.0


Modulus of error for sin(25) is [25.132351750097772, 2579.0343149165687, 78801.17401841676, 1132213.8309419006,
9380069.198227521, 50348811.64932602, 188949589.1822186, 523248032.34023565, 1113235472.996286, 1877414207.8087
611, 2572957341.008273, 2924043089.842807, 2801999025.6270676, 2295972943.2741737, 1627958683.035279, 100909214
1.097418, 551662371.2917258, 268061637.31601775, 116568621.97815621, 45640495.47046641, 16177003.557209888, 521
6095.49721851, 1536776.6790530798, 415371.8455397712, 103375.10508545494, 23768.75536190441, 5064.667272202846,
1002.9721710015428, 185.08379767099564, 31.904799763351686, 5.149264006174836, 0.7797656548477133, 0.1110140418
2033806, 0.01488765660063257, 0.00188313656072861, 0.00022586660342646667, 2.4918590180250666e-05, 3.32298567637
3902e-06]


Modulus of error for e^(-4) is [0.9816843611112658, 3.018315638888734, 4.981684361111266, 5.6849823055554, 4.98
1684361111266, 3.5516489722220674, 2.1372399166668217, 1.1135537341268291, 0.5118430912699963, 0.21055549779525
945, 0.0784039378308428, 0.026672220578648945, 0.008353165557848305, 0.0024238763303046917, 0.00065527849488188
07, 0.00016582945850120503, 3.944752984456554e-05, 8.852938001499555e-06]
```