In [1]:
```python
# Question 3
import math as mt
import numpy as np

# method 1
def e_exp_x1(order,x):
    sum = 0
    for n in range (order+1):
        # (((-1)**n)*(x**n))/mt.factorial(n) is the general formula for n_th term
        sum += (((-1)**n)*(x**n))/mt.factorial(n)
    return sum

# method 2
def e_exp_x2(order,x):
    sum = 0
    for n in range (order+1):
        # (x**n)/mt.factorial(n) is the general formula for the n_th term in the denominator
        sum += (x**n)/mt.factorial(n)
    return 1/sum

#specify the order and x value of taylor series
x = 1
order = 3

#calculate sum of series with two equations
eq1 = e_exp_x1(order=order,x=x)
eq2 = e_exp_x2(order=order,x=x)

#get real value from math function
true_val = mt.e**(-x)

#calculate relative error for both equations
error1 = abs(eq1-true_val)/true_val
error2 = abs(eq2-true_val)/true_val

# print all output with description
print("True Value", true_val)
print("Equation 1: ", eq1, "relative error: ", error1)
print("Equation 2: ", eq2, "relative error: ", error2)

#conclusion: method 2 [1/series] is more subjected to error
#the series is a special type of taylor series which expand at x = 0 (known as machaurin series), so when tested x with large value (eg 2 or 100), the error increase rapidly
#the higher the order, the smaller the error for both equations
```

```
True Value 0.36787944117144233
Equation 1:  0.33333333333333337 relative error:  0.09390605718031818
Equation 2:  0.375 relative error:  0.01935568567214193
```

In [ ]:

In [ ]: