Q9a

```
import math

def func( t, y ):
    return (2/t) * y + t*t*math.exp(t)

# Function for euler formula
def euler( t0, y, h, t ):
    temp = -0

    # Iterating till the point at which we
    # need approximation
    while t0 < t:
        temp = y
        y = y + h * func(t0, y)
        t0 = t0 + h

    # Printing approximation
    print("Approximate solution at t = ", t, " is ", "%.6f"% y)
    return y



def exactfunc(t):
  return t*t*(math.exp(t)-math.exp(1))


t0 = 1
y0 = 0
h = 0.1
t = 1.1
approx = euler(t0, y0, h, 1.1)
exact = exactfunc(1.1)
print("The exact value at t = " ,t, " is ", "%.6f"% exactfunc(1.1))
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.1  is  0.271828
The exact value at t =  1.1  is  0.345920
0.07409169369383534
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.2
approx = euler(t0, y0, h, 1.2)
exact = exactfunc(1.2)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.2  is  0.684756
The exact value at t =  1.2  is  0.866643
0.1818869580441973
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.3
approx = euler(t0, y0, h, 1.3)
exact = exactfunc(1.3)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.3  is  1.276978
The exact value at t =  1.3  is  1.607215
0.330236733972034
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.4
approx = euler(t0, y0, h, 1.4)
exact = exactfunc(1.4)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.4  is  2.093548
The exact value at t =  1.4  is  2.620360
0.5268118633981391
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.5
approx = euler(t0, y0, h, 1.5)
exact = exactfunc(1.5)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.5  is  3.187445
The exact value at t =  1.5  is  3.967666
0.7802211717688712
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.6
approx = euler(t0, y0, h, 1.6)
exact = exactfunc(1.6)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
```

```
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.6  is  4.620818
The exact value at t =  1.6  is  5.720962
1.1001436793168287
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.7
approx = euler(t0, y0, h, 1.7)
exact = exactfunc(1.7)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.7  is  6.466396
The exact value at t =  1.7  is  7.963873
1.4974771001353657
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.8
approx = euler(t0, y0, h, 1.8)
exact = exactfunc(1.8)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.8  is  8.809120
The exact value at t =  1.8  is  10.793625
1.984504971547219
```

```
t0 = 1
y0 = 0
h = 0.1
t = 1.9
approx = euler(t0, y0, h, 1.9)
exact = exactfunc(1.9)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  1.9  is  11.747997
The exact value at t =  1.9  is  14.323082
2.5750849919285184
```

```
t0 = 1
y0 = 0
h = 0.1
t = 2
approx = euler(t0, y0, h, 2)
```

```
exact = exactfunc(2)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  2  is  15.398236
The exact value at t =  2  is  18.683097
3.2848614291071687
```

Q9b

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

# xi -> corresponds to the new data point
# n -> represents the number of known data points
def interpolate(f: list, xi: int, n: int) -> float:
    result = 0.0
    for i in range(n):
        term = f[i].y
        for j in range(n):
            if j != i:
                term = term * (xi - f[j].x) / (f[i].x - f[j].x)
        result += term
    return result



#(i)
# x is the same as t
#(t0, y0) = (1,0)
#(t1, y1) = (1.1, 0.271828)
d1 = [Point(1,0), Point(1.1, 0.271828)]
approx = interpolate(d1, 1.04, 2)
print(approx)
exact = exactfunc(1.04)
error = abs(exact-approx)
error
```

```
0.10873120000000001
0.011256297061343915
```

```
#(ii)
d1 = [Point(1.5,3.187445), Point(1.6, 4.620818)]
approx = interpolate(d1, 1.55, 2)
print(approx)
exact = exactfunc(1.55)
error = abs(exact-approx)
error
```

```
3.9041315
0.8845035208014025
```

```
#(iii)
d1 = [Point(1.9,11.747997), Point(2, 15.398236)]
approx = interpolate(d1, 1.97, 2)
print(approx)
exact = exactfunc(1.97)
error = abs(exact-approx)
error
```

```
14.303164300000002
2.976134135557661
```

Q11a

```
def func( t, y ):
    return -y + t + 1
```

```
def exactfunc(t):
  return math.exp(-t)+t
```

```
t0 = 0
y0 = 1
h = 0.2
t = 5
approx = euler(t0, y0, h, 5)
exact = exactfunc(5)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  5  is  5.003778
The exact value at t =  5  is  5.006738
0.0029600538127887432
```

```
t0 = 0
y0 = 1
h = 0.1
t = 5
approx = euler(t0, y0, h, 5)
exact = exactfunc(5)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  5  is  5.104638
The exact value at t =  5  is  5.006738
0.09790045068750253
```

```
t0 = 0
y0 = 1
h = 0.05
t = 5
approx = euler(t0, y0, h, 5)
exact = exactfunc(5)
print("The exact value at t = " ,t, " is ", "%.6f"% exact)
error = abs(exact-approx)
error
```

```
Approximate solution at t =  5  is  5.055625
The exact value at t =  5  is  5.006738
0.04888655576022316
```