```
# Matthew Pahayo
# 2/11/2021
# computational methods
# hw 1
# question 3
# Mod. newtons method

import numpy
# create variable
import sympy as sym
x = sym.Symbol('x')

# define function, initial guess, and tol
f = sym.exp(6*x)+3*sym.ln(2)**2*sym.exp(2*x)-sym.exp(4*x)*sym.ln(8)-sym.ln(2)**3
t_0 = -1
tol = 10**-5

# lambdify funtion and its derivative
fp = sym.diff(f)
fpp = sym.diff(fp)
f = sym.lambdify(x, f)
fp = sym.lambdify(x, fp)
fpp = sym.lambdify(x, fpp)

# # get next iteration for t and declare lists
t_1 = t_0 - f(t_0)*fp(t_0)/((fp(t_0))**2-f(t_0)*fpp(t_0))
tl = [t_0, t_1]
fl = []
dtl = []

# newtons method
n = 0
while (abs(tl[n+1]-tl[n])/abs(tl[n])) > tol:
    t_np1 = tl[n+1] -
f(tl[n+1])*fp(tl[n+1])/((fp(tl[n+1]))**2-f(tl[n+1])*fpp(tl[n+1]))
    tl.append(t_np1)
    n += 1

# define fl and dtl
for j in range(len(tl)):
    fl.append(f(tl[j]))
for k in range(len(tl)-1):
    dtl.append(tl[k+1]-tl[k])

data_fl = open("flm.txt", "a")
data_tl = open("tlm.txt", "a")
for i in range(len(tl)):
    data_tl.write(str(tl[i]) + "\n")
    data_fl.write(str(fl[i]) + "\n")
data_tl.close()
```

```
data_fl.close()

i = 0
data_dtl = open("dtlm.txt", "a")
for i in range(len(dtl)):
    data_dtl.write(str(dtl[i]) + "\n")
data_dtl.close()

print("t = " + str(t_np1))
print("total iterations = " + str(n))
print("final delta t = " + str(dtl[n-1]))

# keeps python console up; for non-IDE use
i = 1
while i == 1:
    i = input("Press enter to continue.\n")
```