Lab 11b Activity 1

ANSWERS TO THE QUESTIONS BELOW

```
# By submitting this assignment, I agree to the following:
#   "Aggies do not lie, cheat, or steal, or tolerate those who
do."
#   "I have not given or received any unauthorized aid on this
assignment."
#
# Name:          PRANAV VEERUBHOTLA
# Section:       504
# Assignment:    lab 10b-2
# Date:          27 OCTOBER 2019
#
'''This program uses functions to observe the operations of
Newtons method.'''
import matplotlib.pyplot as plt

#Part A:
#Define a function F that takes in an x-value and returns f(x)
def F(x):
    answer = 4 * x**6 + 3 * x**2
    return answer

#Part B:
#Define a function that approximates the derivative of F at a
particular value, x
def deriv(x):
    f_x = F(x)
    a = 0.00001
    f_xplusa = F(x+a)
    deriv_ans = (f_xplusa - f_x)/a
    return deriv_ans

#Part C:
#Define a function that computes a step of newtons method
def newton_step(xi):
    xi_plus1 = xi - (F(xi)/deriv(xi))
    return xi_plus1
```

```
#Part D:
'''Define a function that will take an initial guess for a root
of x. The function
should compute a sequence of root approximations, stopping when
the difference
between successive approximations is less than 10^-6.'''
def newton(x0):
    current_app = x0
    new_app = newton_step(x0)
    print(new_app)
    approx_seq = []
    while abs(current_app - new_app) > 10**(-6):
        approx_seq.append(new_app)
        current_app = new_app
        new_app = newton_step(current_app)
    return approx_seq

#Part E:
#Ask the user to enter their initial guess
guess = float(input('Please enter your first guess for the
root:'))
seq = newton(guess)
print('The series of approximations is as follows:\n', seq)
#Plot the function appropriately
#Gather some values for f(x)
points = []
x_vals = []
for i in range(-15,16):
    points += [F(i)]
    x_vals += [i]
#Plot f(x)
plt.plot(x_vals, points, label = 'F(x)', color = 'r')
plt.xlabel('x')
plt.ylabel('F(x)')
plt.title('Line plot of F(x) for Domain:[-15,15]')
plt.legend(loc = 'upper right', frameon = True)
plt.grid(linestyle = '-', linewidth = '0.5', color = 'black')

#Part F:
```

```
'''When the function is changed to f(x) = x**2 + 1 the function
will continue trying
to approximate the root, however because an actual root does not
exist, the function will
never exit the while loop.'''

#Part G:
'''The function used was f(x) = 4 * x**6 + 3 * x**2:
    a. Yes, for all the tests I ran, all root approximations
seem to converge towards 0
    b. The program tends to usually provide 20-25 approximations
for the test cases I used
    c. Newtons method should take fewer iterations.
    d.  i.   The bisection method evaluates the function F once
per every iteration
        ii.  Newtons method evaluates the function F twice per
every iteration
        iii. No, it does not change my answer for part C,
because Newtons method evaluates F twice every iteration in
order to converge in fewer iterations.'''
```