Name:
Date:
Section:
Assignment:

Problem 6.1 Write a program to use Newton's method that will find a root of f(x) given initial guess x_0. Use a desired accuracy $\in = 1 \times 10^{-3}$ and maximum steps of 100.
a) Find all three roots using different guesses (x_0) and list the guesses that gave you the unique roots that you get from the program neatly. Keep your guesses within the given range.
b) Plot f(x) for the range $x \in [-10, 20]$ and overlay the three points on the same plot. Include this plot in your results. Don't forget to title and label the plot appropriately before saving it.

a) Guesses: x = -1, 2, 17
Roots: x = -0.7131, 2.3303, 16.9179

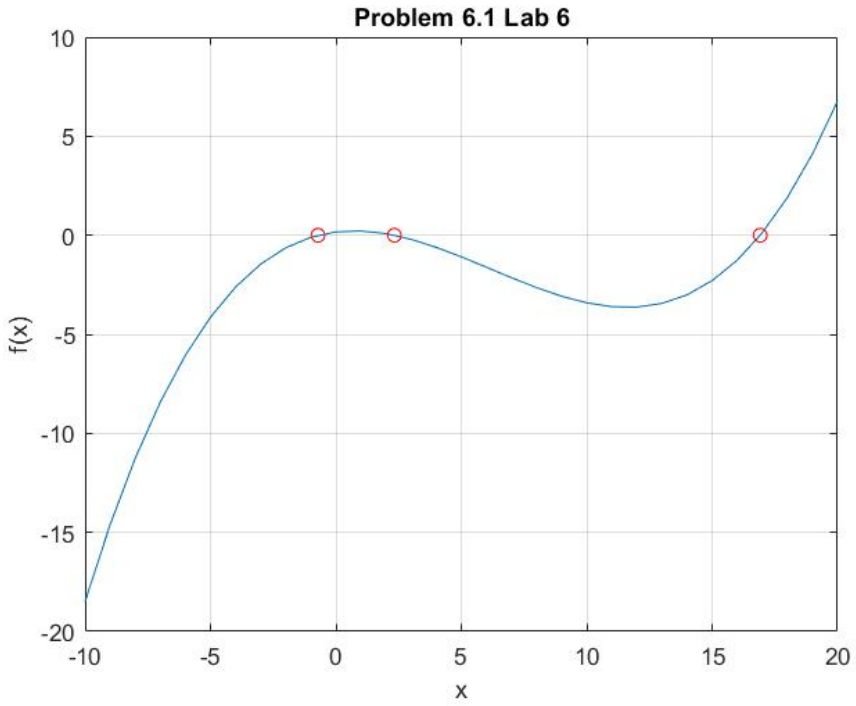
```
>> NewtonRhapson(-1)
iteration 1: -0.73949
iteration 2: -0.71312

ans =

   -0.7131


>> NewtonRhapson(2)
iteration 1: 2.3664
iteration 2: 2.3303

ans =

    2.3303

>> NewtonRhapson(17)
iteration 1: 16.9187
iteration 2: 16.9179

ans =


   16.9179
```

MATLAB Code for Newton Raphson

```
function [Root] = NewtonRhapson(x);

x_n = x;
Value_xn = f(x_n);
count = 0;

while abs(Value_xn) >= (10^(-3)) && count <= 100
    Value_xn = f(x_n);
    Deriv_xn = f_derivative(x_n);
    x_np1 = x_n - (Value_xn/Deriv_xn);
    x_n = x_np1;
    count = count + 1;
```

```
    disp(['iteration ', num2str(count), ': ', num2str(x_n)])
    Value_xn = f(x_n);
end

Root = x_n;
NumIterations = count;
```

b)



Problem 6.2 Write a script takes two initial guesses and uses the Secant method to find a root of the function g(x) with a desired accuracy $|g(x)| < 1\times10^{-3}$ and a maximum number of iterations of 100.

a) Find all roots of the function that exist on the interval [-3.5, -1.5] by choosing different initial guesses. List the guesses that gave you each of the unique roots.

b) Plot g(x) for the given interval and overlay the three points on the same plot. Include this plot in your results. Don't forget to title and label the plot appropriately before saving it.

a) Guesses: x = [-3.3, -3.27]; [-2.65, -2.62]; [-1.98, -1.95]
Roots: x = -3.2810, -2.6335, -1.9610

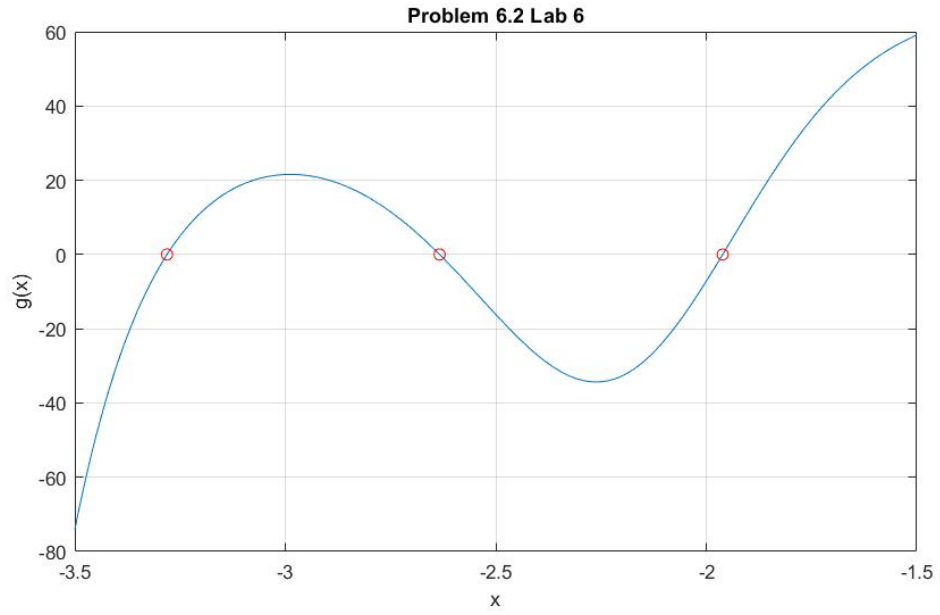
```
>> Secant(-3.3, -3.27)
iteration 1: -3.2804
iteration 2: -3.281
iteration 3: -3.281

ans =

   -3.2810

>> Secant(-2.65, -2.62)
iteration 1: -2.6337
iteration 2: -2.6335
```

```
ans =

   -2.6335

>> Secant(-1.98, -1.95)
iteration 1: -1.961
iteration 2: -1.961

ans =

   -1.9610
```

MATLAB Code:

```
function [Root] = Secant(x_0, x_1);

x_n = x_1;
x_nm1 = x_0;
Value_xn = g(x_n);
count = 0;

while abs(Value_xn) >= (10^(-3)) && count <= 100
    Value_xn = g(x_n);
    Value_xnm1 = g(x_nm1);
    x_np1 = x_n - Value_xn*((x_n-x_nm1)/(Value_xn-Value_xnm1));
    x_nm1 = x_n;
    x_n = x_np1;
    count = count + 1;
    disp(['iteration ', num2str(count), ': ', num2str(x_n)])
    Value_xn = g(x_n);
end

Root = x_n;
```

b)



Problem 6.3 Run the code that you wrote for Problem 5.5 for function f(x) with the initial guess x_0= 11.61 using the same ϵ and maxSteps as before. What happened? Now use an initial guess of x_0 = 11.62. Explain why the program returned such drastically different results for small changes in initial guesses. Observe the plot near our initial guesses to get a better understanding.

For x = 11.61, the Newton Raphson method returned -0.7129 as the root and for x = 11.62, the program returned 16.9182 for the root. This difference is due to the behaviour of the graph at this location. Calculating the derivative of f(x) at x = 11.61 and x = 11.62 yields -0.0019 and 1.009E-04, respectively. This is important because the slope changes between these two points. Because the slope is negative at x = 11.61, the function returns the closest root to the left. Due to similar reasoning, the function returns the closest root to the right for x = 11.62.

```
>> NewtonRhapson(11.61)
iteration 1: -1950.2327
iteration 2: -1298.1057
iteration 3: -863.3595
iteration 4: -573.5361
iteration 5: -380.3317
iteration 6: -251.5455
iteration 7: -165.7127
iteration 8: -108.5274
iteration 9: -70.4574
iteration 10: -45.1548
iteration 11: -28.3962
iteration 12: -17.3753
iteration 13: -10.2285
iteration 14: -5.715
iteration 15: -3.0005
iteration 16: -1.5189
iteration 17: -0.87568
iteration 18: -0.72206
iteration 19: -0.71288

ans =

   -0.7129

>> NewtonRhapson(11.62)
iteration 1: 36142.3096
iteration 2: 24096.9331
iteration 3: 16066.6823
iteration 4: 10713.1822
iteration 5: 7144.1828
iteration 6: 4764.8507
iteration 7: 3178.6307
iteration 8: 2121.1528
iteration 9: 1416.1707
iteration 10: 946.1873
iteration 11: 632.8721
iteration 12: 424.0059
iteration 13: 284.7778
iteration 14: 191.9834
```

```
iteration 15: 130.1575
iteration 16: 88.9968
iteration 17: 61.6439
iteration 18: 43.5458
iteration 19: 31.6991
iteration 20: 24.1551
iteration 21: 19.6949
iteration 22: 17.5522
iteration 23: 16.9629
iteration 24: 16.9182

ans =

   16.9182
```
