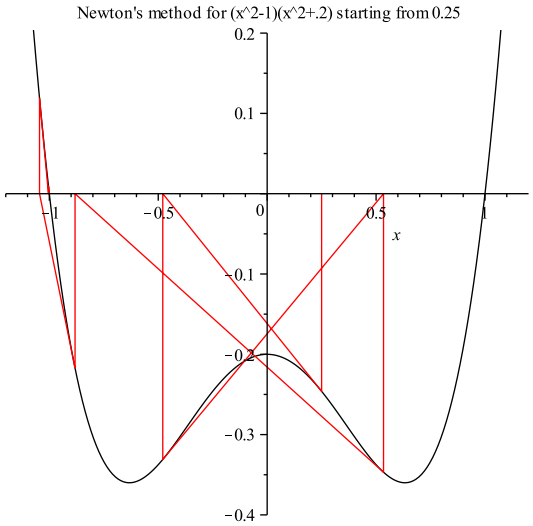
26. (*expires 5/10*) Write a `Maple` procedure to implement Newton's method. As imput, it should take as input a function f, an initial guess x_0, and a number of iterates n. Then it should calculate

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

optionally printing out i, x_i, and $f(x_i)$, as i ranges from 1 to n. It should return x_n.

27. (*expires 5/10*) Adapt your answer to the previous problem to graphically illustrate the process of Newton's method for any given function and initial guess.

Specifically, your procedure should take a function as input a funcion f, an initial value x_0, a number of iterates n, and an x-range for the plot. Then it should print out the values of i, x_i and $f(x_i)$ for $0 \le i \le n$ (where x_{i+1} is given by one step of Newton's method from x_i), and produce a graph of $f(x)$, together with the x_i indicated as in the graph below.




```
1, 0.25, -0.24609375
2, -0.4791666667, -0.3309640767
3, 0.5341996093, -0.3468597849
4, -0.8818848457, -0.2173269360
5, -1.044991129, 0.1188729407
6, -1.003895062, 0.009427277846
7, -1.000032517, 0.00007804622955
8, -1.000000002, 4.800000016*10^-9
```

28. (*expires 5/10*) Consider the following function on positive integers n. If n is even, then let $c(n) = n/2$. If n is odd, let $c(n) = 3n + 1$.

The Collatz conjecture, proposed in 1937 by Luthar Collatz, is that if for any initial n, $c(n)$ is applied repeatedly, the process eventually arrives at with $n = 1$. For example, if we begin with 11, we get the sequence

$$11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1$$

and it takes 14 steps to arrive at 1. The sequence beginning with 27 takes 111 steps to arrive at 1.

Write a *recursive* `Maple` procedure `Collatz` which, given an initial n, calls itself to determine how many steps are reqired to arrive at 1. Along the way, it should print out the value of the sequence. For example, `Collatz(3)` should print out the value 3 and return 1+`Collatz(10)`. `Collatz(1)` should print 1 and return 0.