2. (a) **[5 points]** Create a file `02.c` where you put the `newton` function that we have in the files uploaded in Blackboard. Slightly modify the function so that it also prints the iteration 0 corresponding to the starting point x_0.

   (b) **[10 points]** Newton's method typically has a *local* convergence property, in the sense that convergence to a zero z_0 is guaranteed only when the initial guess x_0 is *close enough* to z_0.

   If this is not the case, the sequence may still converge to the same root, or it may converge to *another* root, or it may *diverge*.

   To investigate this behavior numerically, implement in the same file `02.c` a `main` function where you perform Newton's method for the function $f(x) = \cos(x)$ for various starting points x_0.

   Choose $x_0 = 1.5$ and decrease its value by steps of length $s_1 = 0.5$ as long as $x_0 > -0.2$ (you can implement this with a loop at your choice, such as `for` or `while`).

   Use $\epsilon = 10^{-7}$ and max iterations read from terminal as in the previous problem.

   What happens when $x_0 = 0$? Give your answer with a print to terminal.

   (c) **[5 points]** In the same `main` function, perform another loop where $x_0 = 1.5$ and its value is now decreased by steps of length $s_2 = 0.4$ as long as $x_0 > -0.2$.

   To what true value does the sequence converge when $x_0 = 0.7$? Infer this true value from the numerically obtained one.

   To what true value does the sequence converge when $x_0 = 0.3$? Infer this true value from the numerically obtained one.

   To what true value does the sequence converge when $x_0 = -0.1$? Infer this true value from the numerically obtained one.

   Give your answers by printing to terminal.

The file must compile and the generated executable must run as expected.