Homework 1 (v1)

1) You are expected to implement the Bisection method in either C, C++, Java or Python. The program should work from the command line. The equation to solve will be hard coded inside the program (meaning that you'll have to modify the source and re-compile for every equation to solve). (**40 points**)

The program will have 4 inputs, that it will receive as command line arguments.

Input 1: the start of the root search interval as a real value: a
Input 2: the end of the root search interval as a real value: b
Input 3: the type of stopping criterion as a character array (DISTANCE_TO_ROOT, ABSOLUTE_ERROR, RELATIVE_ERROR)
Input 4: the epsilon value ϵ as a real value

Your program should support all 3 types of stopping criteria:
a) $|f(p_n)|<\epsilon$ (DISTANCE_TO_ROOT)
b) $|p_n - p_{n-1}|<\epsilon$ (ABSOLUTE_ERROR)
c) $\frac{|p_n - p_{n-1}|}{|p_n|}<\epsilon$ (RELATIVE_ERROR)

Your program's output will be to print for every iteration: the current iteration number, the absolute error, and the relative error. If the tolerance value is not reached after 100 iterations, your program should quit with an appropriate error message. If the root is successfully approximated, your program should print with "nice" formatting the approximate root, the number of iterations that have been executed, and the theoretically required number of iterations according to Theorem 2.1.

What to send? a) Your program's source files. b) The outputs of your program for every equation and interval in exercise 6 of section 2.1 using the relative_error stopping criterion and a tolerance of 10E-5.

For the following questions provide every step of your calculations. Do not send photos of your notes. Use equation editors for drafting your response.

2) Solve exercise 5 of section 2.2 and calculate the theoretical number of iterations required according to Corollary 2.5. (**30 points**)

3) Solve exercises 4 and 5 of section 2.3. (**30 points**)

Good luck.