CMPT-439 Numerical Computation
Project 1

The goal of this project is to demonstrate how the bisection method can be used for solving nonlinear equations and develop students' skills in utilizing numerical solutions in software.

---

1. (**40/100**) Design the Bisection function (you may use the m-language and work in Matlab or any high level language) utilizing the bisection method for solving nonlinear equations $f(x)=0$. This function must use the following calling arguments: brackets x_1, x_2; δ (pre-determined tolerance value), and Flag (a stopping criterion for your iterative process based on the type of error used).
   Your function shall accept $f(x)$ as a calling argument (for example, like **anonymous function** in Matlab) too.
   Your function shall return a root, which was found, and the number of iterations needed to find it.
   Your function shall give the user 4 options to stop the iterative process and have 3 corresponding branches, accordingly (**an option to be chosen must be an argument** in your function):

   a) An absolute approximate error is used to stop the process. You may predict the number of iterations in advance in such a case.
   b) An absolute relative approximate error is used to stop the process.
   c) Estimation of a true absolute error is used to stop the process.
   d) Conjunction of an absolute approximate error and an estimated true absolute error is used to stop the process.

   You need to comment on every single line of your Bisection function explaining what this line of your code stands for.

2. (**10/100**) Ask a generative AI tool (ChatGPT or Copilot or Meta AI) to create a function in the same language, which you used and utilizing the bisection method while meeting the same requirements as your function had to meet (in particular, utilizing 4 stopping criteria). Evaluate differences between your code and generative AI code. **Is an AI-generated code correct? Does it meet requirements?**

3. (**30/100**) The equation $2\sin(x)-e^x/4-1=0$ has one root located in the interval [-7, -5] and another root located in the interval [-5, -3]. Find both roots using the bisection method and a **function**, **which you designed**. Use **all four options** to stop calculations for each of the roots and then estimate, which root is better (if any) in terms of closeness $2\sin(x)-e^x/4-1$ to 0 after a corresponding root was plugged in. Use $\delta=10^{-6}$ as your tolerance threshold for all four stopping criteria.

4. (**15/100**) Solve the same equation $2\sin(x)-e^x/4-1=0$ using the function created by generative AI. Use **all four options** to stop calculations for each of the roots and then estimate,

which root is better (if any) in terms of closeness $2\sin(x) - e^x/4 - 1$ to 0 after a corresponding root was plugged in. Use $\delta = 10^{-6}$ as your tolerance threshold for all four stopping criteria.

5. (**<u>15 extra credit points</u>**) Find all roots of the equation $\cos(2x) + \sin(3x) = 0$ in the interval [-1, 3]. Find all these three roots using a function, which you designed in Task 1, using conjunction of an absolute approximate error and an estimated true absolute error as a stopping criterion. You will need to find brackets for your roots and justify your choice of brackets.

6. (**5/100**) Write a brief technical report presenting your results (**all roots found**) and **the corresponding number of iterations** in the **table**. Make a conclusion about the best stopping criterion or about their equivalence to each other and justify your conclusion. Make a conclusion about the quality of the AI-generated implementation of the bisection method.

7. Turn in your source code, an AI-generated code, screen shots of all test run(s) and your brief technical report **in a single zip file**.