Analysis

# Premise

I’ve been interested in the idea of functional programming and have been intending to learn it. The functional programming paradigm is a very different from imperative or object-oriented programming, and I believe requires a unique framework to be developed on. In the past I’ve built various levels of imperative programming simulations, including a simple imperative virtual machine and a custom RISC CPU simulation, though neither of them had useful interfaces to interact with them. For my NEA I would like to build a more useful tool, geared towards allowing the user to visually see the processes of a program and learn about them without being overloaded with information. The application I would like to build is a functional programming IDE, kitted with a code editor featuring syntactic highlighting, and debug functionality for syntax errors and runtime issues, plus a visible (and hopefully useful) call stack widget and I/O queues. By implementing a more user-oriented type of solution, it is possible to bypass the intricacies of an accurate simulation of functional programming processes, and instead focus on delivering a higher-level interpretation of functional programming paradigm, much closer to what a user trying to code in a functional programming language would need to understand and visualise.

Firstly a user interface could be implemented through Windows Forms, for example the RichTextBox form would be appropriate for the code editor, with the use of rich text formatting for syntactic highlighting and showing errors with underlines etc., table forms for showing the stack and I/O queues,