Honors project – Math 151

The program defines a number of functions whose description are given below:

* Function open\_calculator(): Opens a regular non-scientific calculator capable of performing basic addition, subtraction, multiplication and division.
* Function open\_integral\_calculator(): Opens a calculator which performs integration by parts

The Function open\_integral calculator() also has some subfunctions which are listed below:

* Function on\_text\_field\_click(number): A global variable is defined which keeps track of which text field the user activates (u or v). This function updates that variable if the user switches from one text field to type into another text field.
* Function on\_button\_click(character): This function is responsible for adding on to the string in a given text field when the user clicks a button to type into the calculator.
* Function integrate(u,v): Uses sympy to evaluate the integral.
* Function evaluate\_integral(u,v): runs the integrate function and performs the necessary modifications to print out the evaluated integral.

Libraries used:

Tkinter: Tkinter is the standard GUI (Graphical User Interface) toolkit for Python, providing a set of modules that allow developers to create desktop applications with graphical interfaces. Developed as part of the standard library, Tkinter leverages the Tk GUI toolkit, originating from the Tcl language, and offers a simple and intuitive way to design and implement graphical applications. It provides a variety of widgets, such as buttons, labels, and entry fields, which can be arranged using geometry managers like **pack**, **grid**, or **place**. Tkinter enables developers to design interactive applications by associating Python functions with events triggered by user actions. Its simplicity and accessibility make it a popular choice for beginners learning GUI programming in Python, while its versatility allows for the development of sophisticated desktop applications for various purposes.

Sympy: SymPy is a Python library for symbolic mathematics, designed to perform algebraic manipulation, calculus, and other mathematical operations symbolically. As an open-source project, SymPy provides a comprehensive set of tools for working with mathematical expressions, equations, and algebraic structures. It allows users to represent mathematical entities symbolically, offering precise and exact computations without numerical approximations. SymPy supports a wide range of mathematical functionalities, including simplification, equation solving, calculus operations, linear algebra, and more. Its integration with Python facilitates seamless incorporation into scientific computing workflows, and its extensibility allows users to define custom mathematical functions and operations. SymPy is a valuable resource for researchers, educators, and engineers seeking a powerful and flexible symbolic mathematics library within the Python ecosystem. SymPy's **integrate** function is a key feature that allows users to perform symbolic integration, enabling the calculation of antiderivatives and definite integrals of various mathematical expressions. With **integrate**, users can handle a wide range of functions, from elementary to more complex expressions involving algebraic, trigonometric, exponential, and logarithmic functions. The function supports indefinite integration, providing antiderivatives, as well as definite integration, allowing users to compute the area under curves and solve various mathematical problems. SymPy's symbolic approach ensures precise results without relying on numerical approximations. This makes **integrate** a powerful tool for both theoretical exploration of mathematical concepts and practical applications in engineering, physics, and other scientific disciplines where symbolic integration is crucial for obtaining exact solutions.