**COAL PROJECT**



**REPORT**

**Combined Calculator Code**

**Member:**

* **Fuzail Ahmed** 😊 01-134231-025

# Introduction

This projects aims to provide a semi-combined calculator program developed in Irvine32 assembly language. This program offers a range of mathematical operations, including addition, subtraction, multiplication, division, finding remainder, checking for primality, calculating factorial, and continuous calculations. Each operation is executed based on user input, providing a versatile calculator experience.

# Program Structure

The code follows a structured approach, divided into two main sections:

1. **Data Section**: This section contains predefined messages and variables utilized throughout the program. These messages serve as prompts and notifications for the user, while variables store input values and calculated results.
2. **Code Section**: The code section comprises the main program (**main**) and various procedures (**addition, subtraction, multiplication, division, remainder, primeornot, factorial, continuous**) responsible for executing different mathematical operations. These procedures are called based on the user's input selection.

# Functionality Overview

Upon execution, the main program (**main**) displays a welcoming message to the user, followed by a menu of options representing different mathematical operations. The user is prompted to choose an operation by entering the corresponding character. Once selected, the program calls the respective procedure to perform the desired operation. After completion, the user is given the option to continue with another calculation or end the program.

# Detailed Explanation of Operations

1. **Addition (addition):** This procedure prompts the user to input two integers, adds them together, and displays the result. The user is then asked if they want to perform another addition or exit the program.
2. **Subtraction (subtraction):** Similar to addition, this operation prompts the user to input two integers, subtracts the second from the first, and displays the result. The user is given the option to perform another subtraction or exit.
3. **Multiplication (multiplication):** This operation takes two integer inputs, multiplies them, and displays the result. The user is then prompted to continue with another multiplication or exit.
4. **Division (division):** The division operation prompts the user to input two integers, performs division, calculates the decimal value up to 3 decimal places, and displays the result. The user can choose to perform another division or exit.
5. **Remainder (remaind):** Similar to division, this operation takes two integer inputs, divides them, and instead of displaying the decimal value it displays both the quotient and remainder. The user is prompted to continue with another operation or exit.
6. **Prime or Not Check (primeornot):** This operation takes an integer input from the user and checks whether it's a prime number or not. The result is displayed accordingly. The user can choose to perform another check or exit.
7. **Factorial (factorial):** Upon inputting an integer, this operation calculates its factorial and displays the result. The user is then prompted to continue with another calculation or exit.
8. **Continuous Calculations (Continuous):** This operation allows users to perform multiple calculations continuously. It takes an initial value and an operator, performs the operation with the following value, and continues based on user input.

# Conclusion

The combined calculator program demonstrates efficient implementation of mathematical operations in assembly language. Its modular design, user-friendly interface, and versatile functionality make it a valuable tool for arithmetic calculations. The program's structured approach allows for easy extension and modification, enabling potential enhancements in the future.

Overall, the code exemplifies effective programming techniques in assembly language, catering to the needs of users requiring quick and simple calculations.