09/04/2024

F TO C REPORT

Prepared For :

Dr.lamiaa

Description of the Problem:

The problem requires writing a MIPS assembly program to convert Fahrenheit temperatures to Celsius. The formula to perform this conversion is:

$$Celsius = (Fahrenheit - 32) * (5/9)$$

Sample Input:
The program expects the user to input a floating-point value representing the temperature in Fahrenheit like 85

ample Output:
Upon receiving the input Fahrenheit temperature, the program calculates the equivalent temperature in Celsius and outputs it : 29.444447

Problem Statement:
The program should prompt the user to input a temperature value in Fahrenheit. Upon receiving the input, it should calculate the equivalent temperature in Celsius using the provided formula. Finally, the program should display the converted temperature in Celsius to the user.

Updates to Make the Example Work:

1. **Comments for Improved Understanding:**
Throughout the code, comments are added to explain the purpose and functionality of each section. This includes comments for input/output operations, procedure calls, and mathematical operations involved in the Fahrenheit to Celsius conversion.

2. **Definition of Constants in .data Section:**
Constants such as const9, const5, and const32 are declared in the .data section, each with their corresponding floating-point values. This approach provides clarity and allows easy modification of constants if needed.

3. **Corrected Syntax for Loading Constants:**
The original code used the lwc1 instruction to load constants into floating-point registers. However, the correct instruction for loading floating-point constants is l.s. This correction ensures the proper loading of constants into registers.

4. **Proper Passing of Arguments:**
Before calling the f2c procedure, the Fahrenheit temperature input by the user is moved from $f0 to $f12, the designated argument register. This ensures that the procedure receives the correct input for conversion.

6. Ensured Proper Program Termination:
To guarantee the correct termination of the program, the code sets $v0 to 10 before invoking a syscall. This syscall signals the termination of the program, ensuring a graceful exit.