MIPS Assembly Project – Text Analyzer

Simulator: MARS (MIPS Assembler and Runtime Simulator)
Duration: 1 Week

Project Description

You will write a MIPS assembly program that reads a line of text from the user and analyzes it to compute several statistics.

Your program must:

1. Read a line of text (up to 500 characters) from the user.
2. Count and display the following:
   - Total number of characters (excluding the null terminator)
   - Number of vowels (a, e, i, o, u; case-insensitive)
   - Number of consonants (alphabetic but not vowels)
   - Number of digits (0–9)
   - Number of spaces
3. Print the results in a formatted way.
4. Use at least one user-defined function that is called with `jal` and returns using `jr $ra`.
   For example, you may create a function like `countVowels`, `countDigits`, or `countSpaces`.

Bonus (Optional)

- Create separate functions for each type of count (vowels, digits, etc.).
- Convert all lowercase letters to uppercase and print the modified string.
- Add an option to analyze multiple lines until the user enters an empty string.

Technical Requirements

- Use `.data` and `.text` sections properly.
- Define a string buffer with `.space 200` for user input.
- Use `lb` to access each character of the string.
- Follow register conventions properly:
  - `$a0–$a3`: arguments
  - `$v0–$v1`: return values
  - `$t0–$t9`: temporaries
  - `$s0–$s7`: saved registers (for main variables)

- You must include at least one function that is called via `jal`.

Example Output

Enter text: Hello World 123

Characters: 15
Vowels: 3
Consonants: 7
Digits: 3
Spaces: 2

Learning Outcomes

After completing this project, you will be able to:
- Manipulate strings and characters in memory.
- Translate high-level control flow (loops, if statements) into assembly.
- Understand and implement function calls (`jal, jr $ra`) in MIPS.
- Realize that everything done in C can be achieved with assembly.