Overview:

To implement the design of the MIPS system in C, we created a method that simulates through the registers since we did not have them by default. In order to achieve this, we set integers to the normal names of the MIPS registers, these are tied to a global array that will hold the contents of the register. With this, we can simulate MIPS by doing the operations through access to the values stored in registers. To determine what operation is being called, a value is read in from the input data and stored into a memory address. After this value is stored and the opcode is checked, we know what operation we are using. The 32-bit string will be broken down to get the values needed for the operation on the bases of basic instruction formatting. Once these values are obtained, they are sent to a function with the operation that was found through the opcode and the values that are stored in the register.

Group Work:

* As a group
  + Choosing a method to implement the task given and discussing how to best divide the work.
* Kaleb Ross
  + In charge of using the hexadecimal data that was given to us and parsing it into the MIPS instruction set. Created the print instruction function to print out respective MIPS instructions and registers. Created half of the control flow instructions.
* Joseph Tierney
  + In charge of creating the arrays for the register data to be stored and creating function for the ALU operations.
* Joy Adeyemo
  + In charge of creating functions for the load and store operations. Also created some of the control flow instructions.