**Name:**

Note: Please post your homework to ICS232 D2L on or before the due date.

**Chapter 4 – MARIE: An Introduction to a Simple Computer**

**Essential Terms and Concepts**

31. How does a microprogram operation differ from a regular assembly language instruction?

Microprogrammed control uses software of microinstructions to carry out microoperations and is slower where regular assembly language instruction is much faster but harder to change.

39. Compare CISC machines to RISC machines.

CISC has a larger instruction set that can perform multiple operations at once, however, it can have larger code sizes.

RISC has a smaller and simpler instruction set but may need more instructions.

**Exercises**

34. Write the following code segment in MARIE assembly language.

Sum = 0;

for (X = 1; X <= 10; X++)

Sum = Sum + X;

A screenshot of a computer

Description automatically generated

39. MARIE saves the return address for a subroutine in memory, at a location designated by the jump-and-store instruction. In some architectures, this address is stored in a register, and in many it is stored on a stack. Which of these methods would best handle recursion?

Explain your answer. (Hint: Recursion implies many subroutine calls.)

I would think that a Stack would be the best because a stack would allow each recursion to have its own space, Stacks are also LIFO so when a recursion is done it would pop off of the stack.

X1. Write a C program that has two functions named findMax and countOdd. Each function will take two arguments, an integer array and the number of elements in the array. findMax returns the largest element. countOdd returns the count of the number of odd elements in the array. The main function should call findMax and countOdd twice with different arrays and then print out the results of calling the functions. The two arrays used to test the program should be:

static int array1[] = {1, -1, 100, 32, 64, -97, 42};

static int array2[] = {-100, 1, -10, 50, -40, 98, 110, 66};

Submit the C code and the results of executing the program.

**Prepare for next class by reading Chapter 5 – A Closer Look at Instruction Set Architectures.**

**Continue working on Project 1**

**Continue working on Your Group Project**