**­­­­Computer Architecture Lab Report Week 10.1**

**Full name: Nguyen Huu Phuc**

**Student ID: 20215234**

Assignment 1

.eqv SEVENSEG\_ LEFT 0xFFFF0011

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.data

studentID: .word 0x1258

# 7-segment display values for digits 0-9

SEGMENT\_VALUES: .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

.text

main:

lw $t0, studentID # Load the student ID into register $t0

andi $t1, $t0, 0xFF # Extract the last digit

move $t2, $t1 # Copy the student ID to another register

srl $t2, $t2, 4 # Shift right to get the next digit

andi $t1, $t0, 0xF

# Calculate the address offset for SEGMENT\_VALUES array

li $t3, 4 # Each element in SEGMENT\_VALUES is a word (4 bytes)

mul $t1, $t1, $t3 # Multiply the last digit by 4 to get the offset

mul $t2, $t2, $t3 # Multiply the next digit by 4 to get the offset

# Display the last digit on the right 7-segment display

lw $a0, SEGMENT\_VALUES($t1) # Load the segment value for the digit

jal SHOW\_7SEG\_RIGHT

# Display the next digit on the left 7-segment display

lw $a0, SEGMENT\_VALUES($t2) # Load the segment value for the digit

jal SHOW\_7SEG\_LEFT

j exit

exit:

li $v0, 10

syscall

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

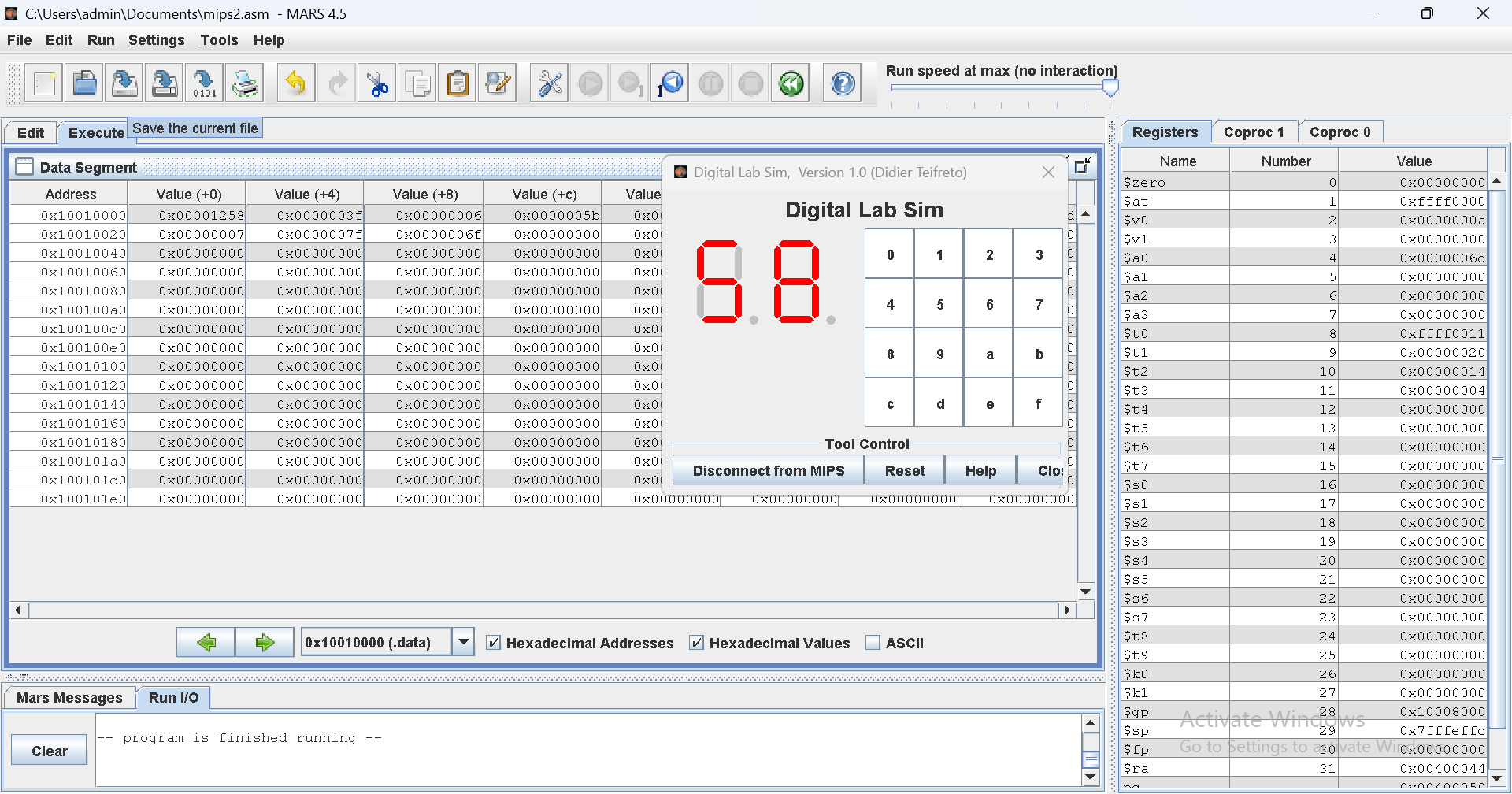
jr $ra

SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra



Assignment 2

.eqv SEVENSEG\_LEFT 0xFFFF0011

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.data

prompt: .asciiz "Enter an integer: "

# 7-segment display values for digits 0-9

SEGMENT\_VALUES: .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

.text

main: # Display prompt

li $v0, 4 # System call code for print\_str

la $a0, prompt # Load address of the prompt string

syscall

# Read integer input

li $v0, 5 # System call code for read\_int

syscall

move $t0, $v0 # Store the input into register $t0

li $t4, 10 # Decimal constant for 10 # Extract the last digit

divu $t0, $t4 # Divide the input by 10 # Copy the student ID to another register

mfhi $t1

mflo $t0 # Remainder is the last digit # Shift right to get the next digit

divu $t0, $t4 # Divide the input by 10 again

mfhi $t2 # Remainder is the next digit

# Calculate the address offset for SEGMENT\_VALUES array

li $t3, 4 # Each element in SEGMENT\_VALUES is a word (4 bytes)

mul $t1, $t1, $t3 # Multiply the last digit by 4 to get the offset

mul $t2, $t2, $t3 # Multiply the next digit by 4 to get the offset

# Display the last digit on the right 7-segment display

lw $a0, SEGMENT\_VALUES($t1) # Load the segment value for the digit

jal SHOW\_7SEG\_RIGHT

# Display the next digit on the left 7-segment display

lw $a0, SEGMENT\_VALUES($t2) # Load the segment value for the digit

jal SHOW\_7SEG\_LEFT

j exit

exit:

li $v0, 10

syscall

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

jr $ra

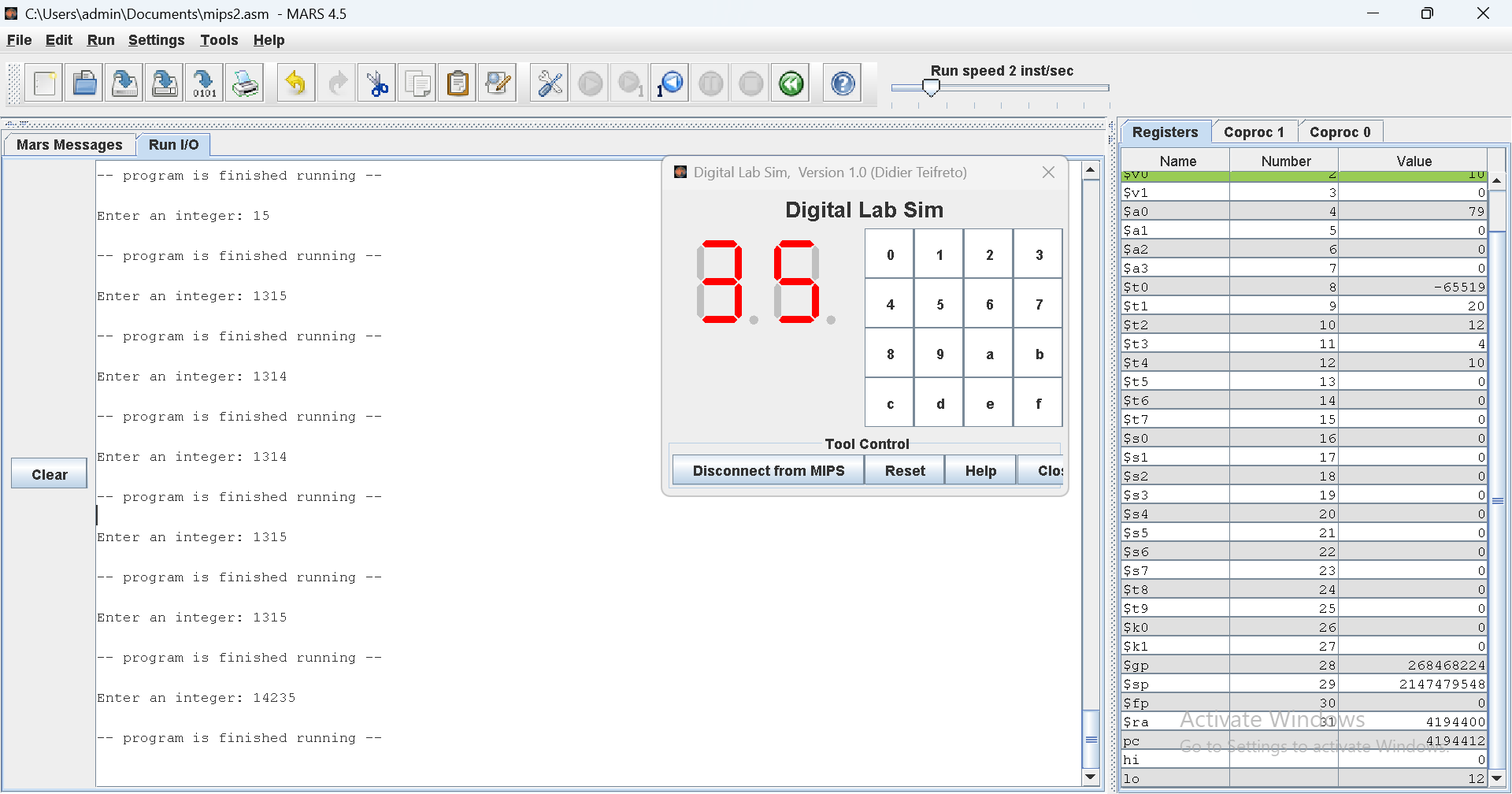
SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra

Results in console:



Assignment 3

.eqv SEVENSEG\_LEFT 0xFFFF0011

.eqv SEVENSEG\_RIGHT 0xFFFF0010

.data

prompt: .asciiz "Enter a character: "

# 7-segment display values for digits 0-9

SEGMENT\_VALUES: .word 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F

.text

main: # Display prompt

li $v0, 4 # System call code for print\_str

la $a0, prompt # Load address of the prompt string

syscall

# Read character input

li $v0, 12 # System call code for read\_char

syscall

move $t0, $v0 # Store the character input into register $t0

li $t4, 10 # Decimal constant for 10 # Extract the last digit

divu $t0, $t4 # Divide the input by 10 # Copy the student ID to another register

mfhi $t1

mflo $t0 # Remainder is the last digit # Shift right to get the next digit

divu $t0, $t4 # Divide the input by 10 again

mfhi $t2 # Remainder is the next digit

# Calculate the address offset for SEGMENT\_VALUES array

li $t3, 4 # Each element in SEGMENT\_VALUES is a word (4 bytes)

mul $t1, $t1, $t3 # Multiply the last digit by 4 to get the offset

mul $t2, $t2, $t3 # Multiply the next digit by 4 to get the offset

# Display the last digit on the right 7-segment display

lw $a0, SEGMENT\_VALUES($t1) # Load the segment value for the digit

jal SHOW\_7SEG\_RIGHT

# Display the next digit on the left 7-segment display

lw $a0, SEGMENT\_VALUES($t2) # Load the segment value for the digit

jal SHOW\_7SEG\_LEFT

j exit

exit:

li $v0, 10

syscall

SHOW\_7SEG\_LEFT:

li $t0, SEVENSEG\_LEFT

sb $a0, 0($t0)

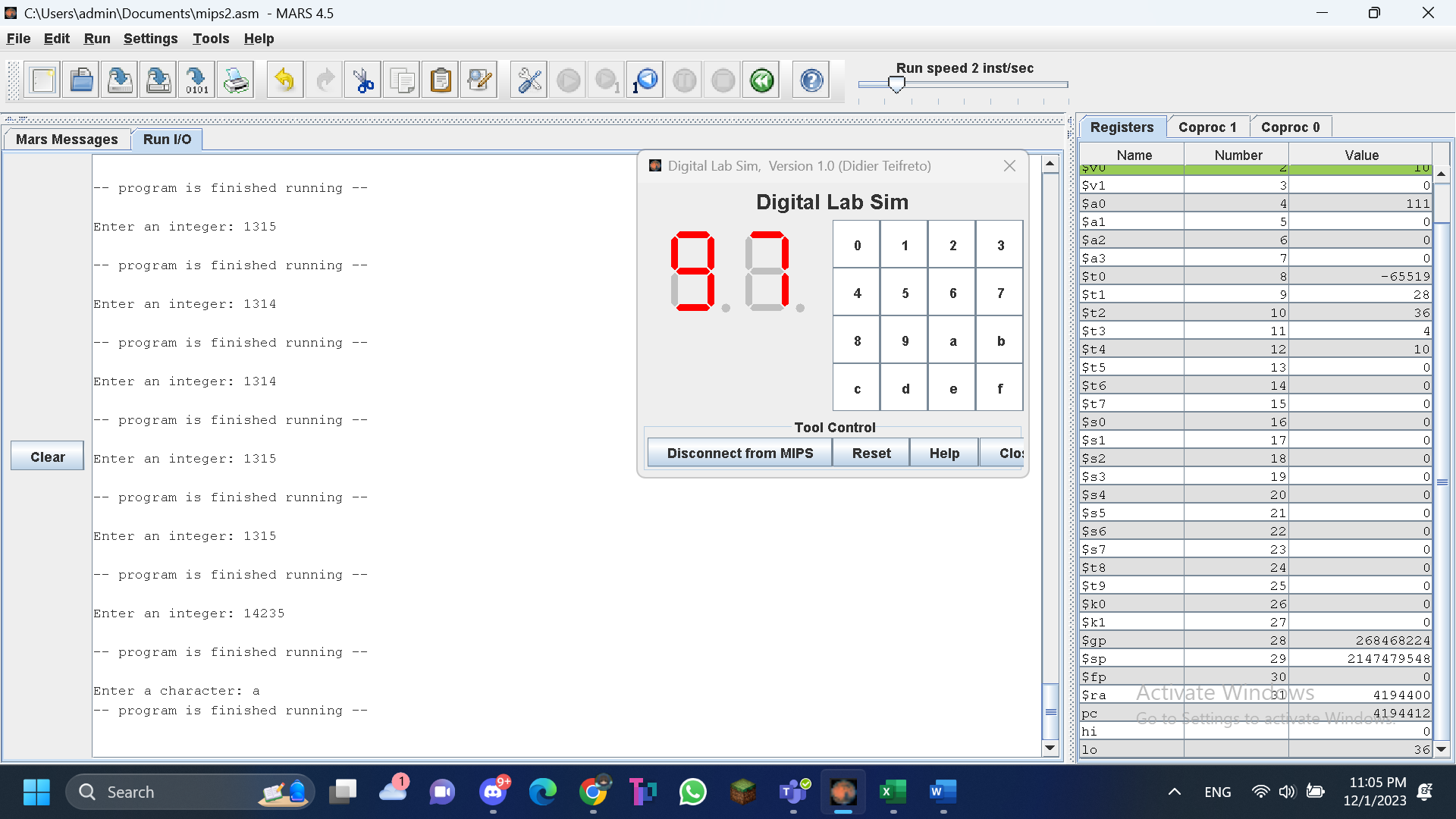
jr $ra

SHOW\_7SEG\_RIGHT:

li $t0, SEVENSEG\_RIGHT

sb $a0, 0($t0)

jr $ra



Assignment 4

Source code:

.eqv MONITOR\_SCREEN 0x10010000

.eqv RED 0x00FF0000

.eqv GREEN 0x0000FF00

.text

li $k0, MONITOR\_SCREEN

li $t1, 0

li $t0, RED

sw $t0, 0($k0)

sw $t0, 8($k0)

sw $t0, 16($k0)

sw $t0, 24($k0)

sw $t0, 36($k0)

sw $t0, 44($k0)

sw $t0, 52($k0)

sw $t0, 60($k0)

sw $t0, 64($k0)

sw $t0, 72($k0)

sw $t0, 80($k0)

sw $t0, 88($k0)

sw $t0, 100($k0)

sw $t0, 108($k0)

sw $t0, 116($k0)

sw $t0, 124($k0)

sw $t0, 128($k0)

sw $t0, 136($k0)

sw $t0, 144($k0)

sw $t0, 152($k0)

sw $t0, 164($k0)

sw $t0, 172($k0)

sw $t0, 180($k0)

sw $t0, 188($k0)

sw $t0, 192($k0)

sw $t0, 200($k0)

sw $t0, 208($k0)

sw $t0, 216($k0)

sw $t0, 228($k0)

sw $t0, 236($k0)

sw $t0, 244($k0)

sw $t0, 252($k0)

li $t0, GREEN

sw $t0, 4($k0)

sw $t0, 12($k0)

sw $t0, 20($k0)

sw $t0, 28($k0)

sw $t0, 32($k0)

sw $t0, 40($k0)

sw $t0, 48($k0)

sw $t0, 56($k0)

sw $t0, 68($k0)

sw $t0, 76($k0)

sw $t0, 84($k0)

sw $t0, 92($k0)

sw $t0, 96($k0)

sw $t0, 104($k0)

sw $t0, 112($k0)

sw $t0, 120($k0)

sw $t0, 132($k0)

sw $t0, 140($k0)

sw $t0, 148($k0)

sw $t0, 156($k0)

sw $t0, 160($k0)

sw $t0, 168($k0)

sw $t0, 176($k0)

sw $t0, 184($k0)

sw $t0, 196($k0)

sw $t0, 204($k0)

sw $t0, 212($k0)

sw $t0, 220($k0)

sw $t0, 224($k0)

sw $t0, 232($k0)

sw $t0, 240($k0)

sw $t0, 248($k0)

Results on console:

A screenshot of a computer

Description automatically generated

Assignment 5