TASK #01:

.data

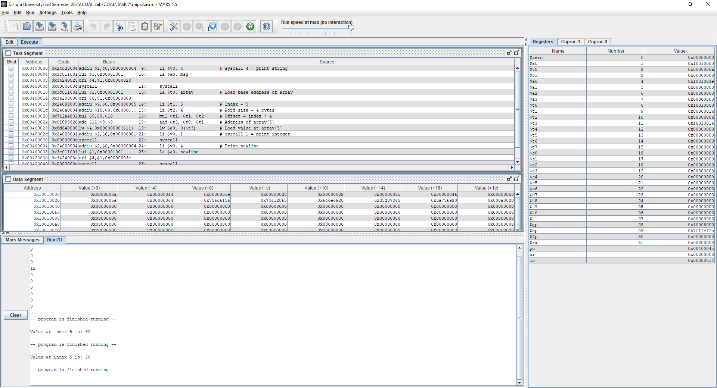
prompt: .asciiz "Enter a number (0 to finish): "

output\_msg: .asciiz "Total Sum: "

avg\_msg: .asciiz ", Average: "

newline: .asciiz "\n"

.text

.globl main

main:

li $t0, 0 # sum

li $t1, 0 # count

loop:

li $v0, 4

la $a0, prompt

syscall

li $v0, 5 # read int

syscall

move $t3, $v0 # input -> $t3

beq $t3, $zero, end\_loop

add $t0, $t0, $t3 # sum += input

addi $t1, $t1, 1 # count += 1

j loop

end\_loop:

beqz $t1, no\_input # if count == 0, avoid division by 0

# Compute average

div $t0, $t1

mflo $t2 # $t2 = average

# Print "Total Sum: "

li $v0, 4

la $a0, output\_msg

syscall

# Print sum

li $v0, 1

move $a0, $t0

syscall

# Print ", Average: "

li $v0, 4

la $a0, avg\_msg

syscall

# Print average

li $v0, 1

move $a0, $t2

syscall

# Newline

li $v0, 4

la $a0, newline

syscall

li $v0, 10

syscall

no\_input:

# Exit silently if no values were entered

li $v0, 10

syscall

TASK #02:

.data

prompt: .asciiz "Enter a number (0 to finish): "

output\_msg: .asciiz "Total Sum: "

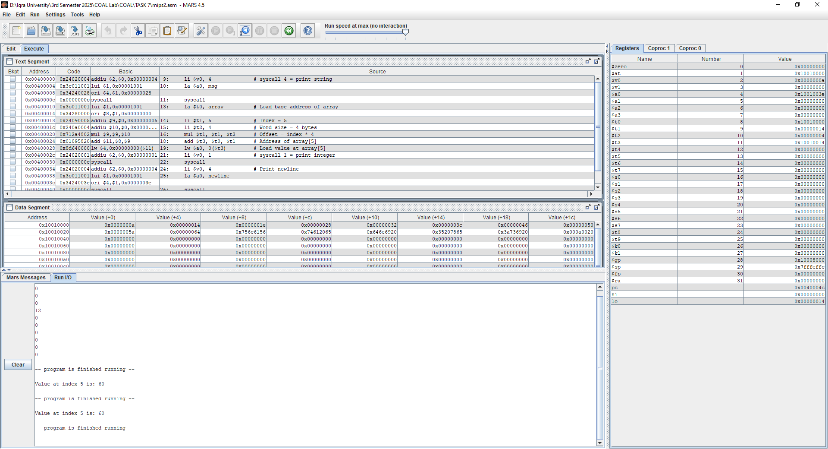
avg\_msg: .asciiz ", Average: "

newline: .asciiz "\n"

.text

.globl main

main:

 li $t0, 0 # sum

li $t1, 0 # count

loop:

li $v0, 4

la $a0, prompt

syscall

li $v0, 5 # read int

syscall

move $t3, $v0 # input -> $t3

beq $t3, $zero, end\_loop

add $t0, $t0, $t3 # sum += input

addi $t1, $t1, 1 # count += 1

j loop

end\_loop:

beqz $t1, no\_input # if count == 0, avoid division by 0

# Compute average

div $t0, $t1

mflo $t2 # $t2 = average

# Print "Total Sum: "

li $v0, 4

la $a0, output\_msg

syscall

# Print sum

li $v0, 1

move $a0, $t0

syscall

# Print ", Average: "

li $v0, 4

la $a0, avg\_msg

syscall

# Print average

li $v0, 1

move $a0, $t2

syscall

# Newline

li $v0, 4

la $a0, newline

syscall

li $v0, 10

syscall

no\_input:

# Exit silently if no values were entered

li $v0, 10

syscall

TASK #03:

.data

prompt: .asciiz "Enter a number (0 to finish): "

invalid: .asciiz "Invalid input! Please enter a non-negative number.\n"

num\_msg: .asciiz "Numbers entered: "

count\_msg: .asciiz "Count of Numbers: "

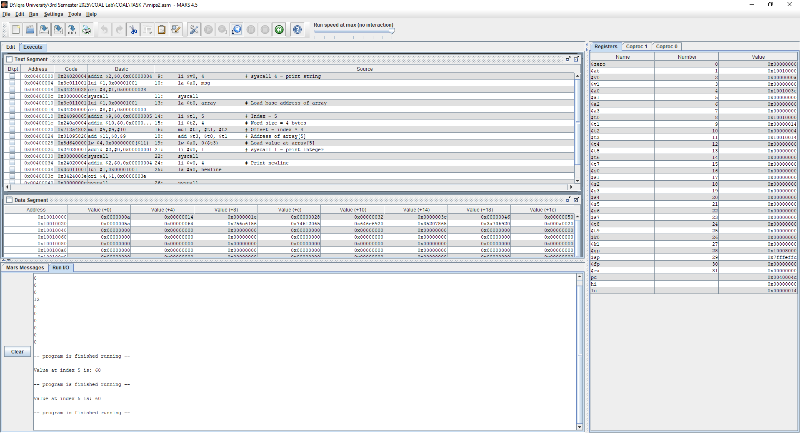
avg\_msg: .asciiz ", Average: "

space: .asciiz " "

newline: .asciiz "\n"

inputs: .space 400 # store up to 100 numbers (4 bytes each)

.text

.globl main

main:

li $t0, 0 # sum

li $t1, 0 # count

la $s0, inputs # array pointer

input\_loop:

# Prompt user

li $v0, 4

la $a0, prompt

syscall

# Read integer input

li $v0, 5

syscall

move $t3, $v0

bltz $t3, invalid\_input # if < 0 ? show error

beqz $t3, done\_input # if == 0 ? end

# Store input

sw $t3, 0($s0)

addi $s0, $s0, 4

add $t0, $t0, $t3 # sum += input

addi $t1, $t1, 1 # count++

j input\_loop

invalid\_input:

li $v0, 4

la $a0, invalid

syscall

j input\_loop

done\_input:

beqz $t1, exit\_program # skip output if nothing valid

# Print "Numbers entered: "

li $v0, 4

la $a0, num\_msg

syscall

la $s0, inputs # reset pointer

li $t4, 0 # index = 0

print\_numbers:

beq $t4, $t1, after\_numbers

lw $a0, 0($s0)

li $v0, 1

syscall

# Print space

li $v0, 4

la $a0, space

syscall

addi $s0, $s0, 4

addi $t4, $t4, 1

j print\_numbers

after\_numbers:

# Newline

li $v0, 4

la $a0, newline

syscall

# Average = sum / count

div $t0, $t1

mflo $t2

# Print "Count of Numbers: "

li $v0, 4

la $a0, count\_msg

syscall

li $v0, 1

move $a0, $t1

syscall

# Print ", Average: "

li $v0, 4

la $a0, avg\_msg

syscall

li $v0, 1

move $a0, $t2

syscall

# Final newline

li $v0, 4

la $a0, newline

syscall

exit\_program:

li $v0, 10

syscall