Task No 01:

Write a MIPS assembly program that reads an integer from the user and then determines whether it's even or odd.

Solution:

.data

prompt: .asciiz "Enter an integer: "

even\_message: .asciiz "Your number is even."

odd\_message: .asciiz "Your number is odd."

.text

main:

# Read an integer from the user.

li $v0, 4

la $a0, prompt

syscall

li $v0,5 # 1 INPUT

syscall

# Store the user's input in register $t0.

move $t0, $v0

# Check if the user's input is even.

li $t1, 2

div $t0, $t1

mfhi $t2

beqz $t2,even

j odd

even:

li $v0, 4

la $a0, even\_message

syscall

j exit

odd:

# Print the odd message.

li $v0, 4

la $a0, odd\_message

syscall

exit:

# Exit the program.

li $v0, 10

syscall

Output:

A black text on a white background

Description automatically generated

Task No 02:

Write a MIPS assembly program that determines the largest of three integers entered by the user. The program should prompt the user to enter three integers and then print the largest among them.

Solution:

Output:

Task No 03:

Write a MIPS assembly program that uses loop to calculate the sum of integers from 1 to N. (N is an input by user)

Solution:

.data

prompt: .asciiz "Enter Value"

sum\_message: .asciiz "The sum of integers from 1 to N is: "

.text

main:

li $v0, 4

la $a0, prompt

syscall

li $v0,5 # 1 INPUT

syscall

# Store the user's input in register $t0.

move $t0, $v0

li $t1, 0

# Start the loop.

loop:

# Add the current loop iteration to the sum variable.

add $t1, $t1, $t0

# Decrement the loop counter.

sub $t0, $t0, 1

# Check if the loop counter is equal to zero.

beq $t0, $zero, end

# Jump back to the beginning of the loop.

j loop

end:

# Print the sum message.

li $v0, 4

la $a0, sum\_message

syscall

# Print the sum of integers from 1 to N.

move $a0, $t1

li $v0, 1

syscall

# Exit the program.

li $v0, 10

syscall

Output:

A black text on a white background

Description automatically generated