**TASK NO:01 OF LAB NO 06**

.data

mask0:.word 0x00000000

mask1:.word 0xffffffff

spaceseq: .asciiz "\n"

.text

li $v0,5

syscall

move $t0,$v0

lw $t1, mask0

lw $t2, mask1

and $t5,$t0,$t2

and $t6,$t0,$t1

move $a0,$t5

li $v0,1

syscall

la $a0,spaceseq

li $v0,4

syscall

move $a0,$t6

li $v0,1

syscall

li $v0,10

syscall

**LAB NO 07**

**Task**

.data

equalPrompt: .asciiz"The number are equal"

not EqualPrompt: .asciiz "The number aren't equal"

.text

li $v0,5 # 1 INPUT

syscall

move $t0,$v0

li $v0,5 # 2 INPUT

syscall

move $t1,$v0

beq $t0, $t1, equalNumbers

j notEqualNumbers

equalNumbers:

la $a0,equalPrompt

li $v0,4

syscall

j exit

notEqualNumbers:

la $a0,notEqualPrompt

li $v0,4

syscall

j exit

exit:

li $v0,10

syscall

**Task no 01**

.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

**While Task**

.data

outPrompt: .asciiz "you are out of loop"

.text

li $t0,0

while:

beq $t0, 10,out

move $a0,$t0

li $v0,1

syscall

addi $t0,$t0,1

j while

out:

la $a0,outPrompt

li $v0,4

syscall

li $v0,10

syscall

**do While Task**

.data

outPrompt: .asciiz "\n you are out of loop"

.text

li $t0,1

dowhile:

beq $t0, 10,out

move $a0,$t0

li $v0,1

syscall

addi $t0,$t0,1

j dowhile

out:

la $a0,outPrompt

li $v0,4

syscall

li $v0,10

syscall

**Task**

.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