# Andrew Souza

# Comp-265 -- Spring 2024

# Building Our Skills in Assembly Assignment -- Problem 2

# My first step was to figure out how to perform this operation in a HLL

# I have some basic javascript proficiency, and the js console in a web

# browser is a convenient place to test some simple code.

# I found that in javascript console.log(~n) == console.log(n ^ -1)

.text

.globl main

main:

# Prompt user for input

li $v0, 4

la $a0, prompt1

syscall

# Take input of integer

li $v0, 5

syscall

move $s0, $v0

# Perform operation on integer

# Performs Not operation by xor-ing the user input against -1

addi $s1, $s1, -1

xor $s1, $s1, $s0

# Output results

li $v0, 4

la $a0, output1

syscall

li $v0, 1

move $a0, $s0

syscall

li $v0, 4

la $a0, outputXOR

syscall

li $v0, 1

move $a0, $s1

syscall

# Exit program

ori $v0, $zero, 10

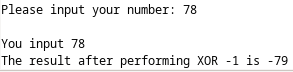
syscall

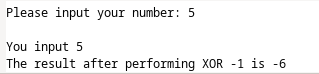
.data

prompt1: .asciiz "\nPlease input your number: "

output1: .asciiz "\nYou input "

outputXOR: .asciiz "\nThe result after performing XOR -1 is "





# Andrew Souza

# Comp-265 -- Spring 2024

# Building Our Skills in Assembly Assignment -- Problem 5

# I again started by finding a high level representation of the

# problem in the javascript console. I found that (n << 3) + (n + n)

# is equivalent to n \* 10

.text

.globl main

main:

# Get Value from input

li $v0, 4

la $a0, prompt

syscall

li $v0, 5

syscall

move $s0, $v0

# Operate on input

sll $s1, $s0, 3 # Shifts $s0 3 bits to left, stores result in $s1

add $s2 $s0, $s0 # Adds $s0 to $s0, stores result to $s2

add $s3 $s1, $s2 # Adds $s1 to $s2, stores result to $s3

# Output results

li $v0, 4

la $a0, output

syscall

li $v0, 1

move $a0, $s0

syscall

li $v0, 4

la $a0, outputAfter

syscall

li $v0, 1

move $a0, $s3

syscall

# Exit program

ori $v0, $zero, 10

syscall

.data

prompt: .asciiz "\nInput an integer value: "

output: .asciiz "\nYou input "

outputAfter: .asciiz "\nYour input \* 10 = "