

Lab #3 Report

Name: ZHU He (祝禾)

ID Number: 202283930036

Major: Software Engineering

Task 1

Assembly Code:

```
.data
promptA: .asciiz "Enter integer A: "
promptB: .asciiz "Enter integer B: "
result: .asciiz "Result of A + 2B - 5: "

.text
.globl main

main:
    # Print prompt for A
    li $v0, 4
    la $a0, promptA
    syscall

    # Read integer A
    li $v0, 5
    syscall
    move $t0, $v0 # Move read integer into $t0

    # Print prompt for B
    li $v0, 4
    la $a0, promptB
    syscall

    # Read integer B
    li $v0, 5
    syscall
    move $t1, $v0 # Move read integer into $t1

    # Compute 2B
    sll $t1, $t1, 1 # $t1 = $t1 * 2

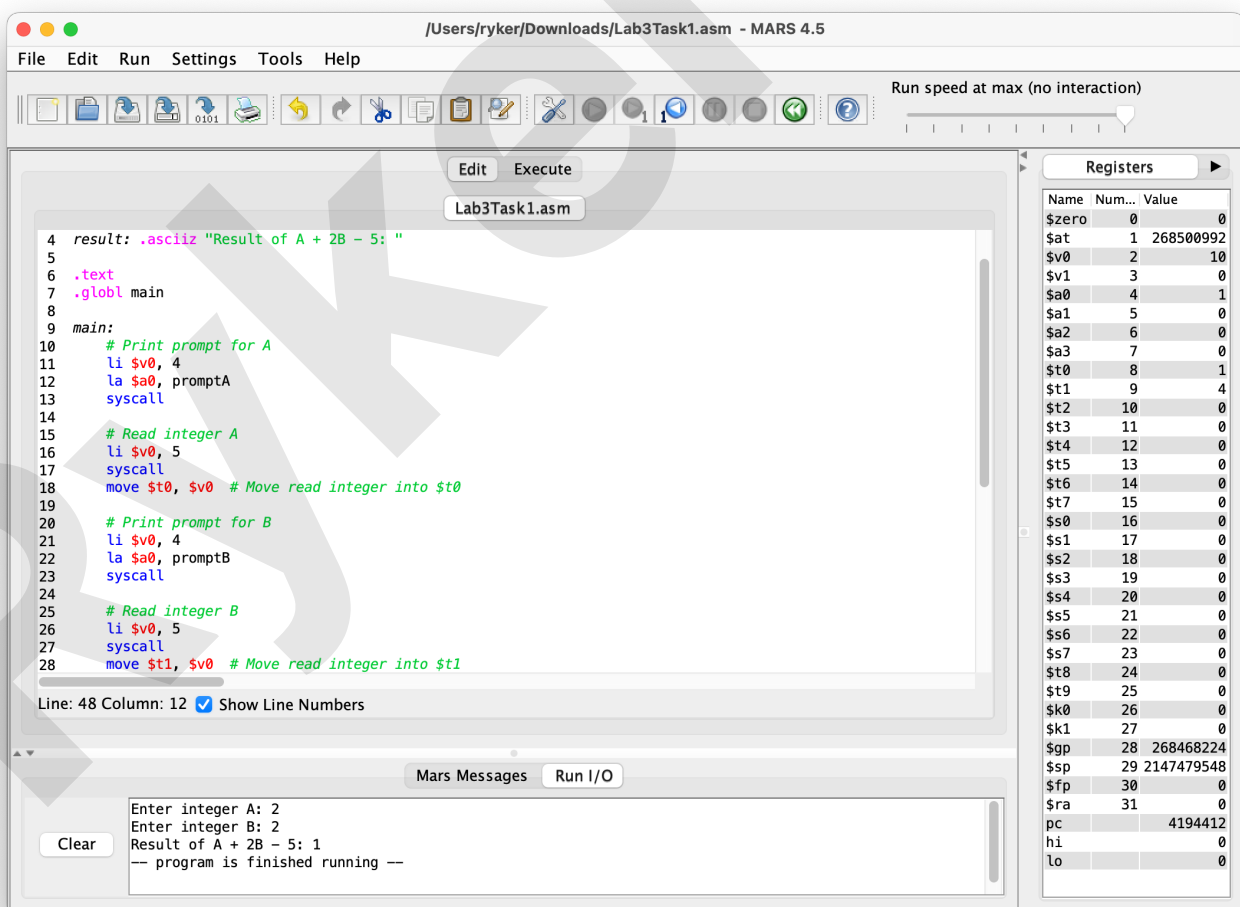
    # Compute A + 2B - 5
    add $t0, $t0, $t1 # $t0 = $t0 + $t1
    subi $t0, $t0, 5 # $t0 = $t0 - 5

    # Print result
    li $v0, 4
    la $a0, result
    syscall

    li $v0, 1
    move $a0, $t0
    syscall

    # Exit program
    li $v0, 10
    syscall
```

Screenshot:



Task 2

Assembly Code:

```
.data

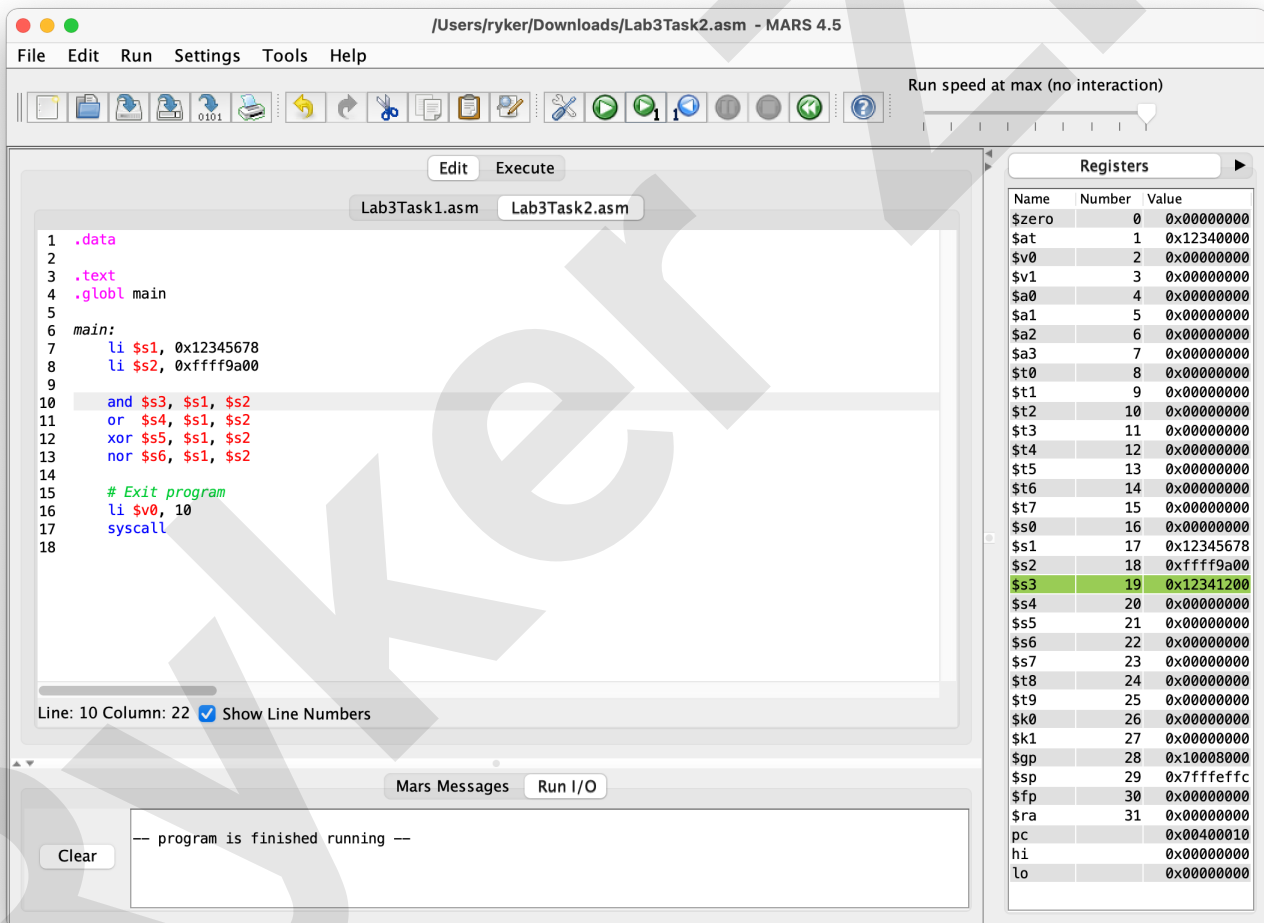
.text
.globl main

main:
    li $s1, 0x12345678
    li $s2, 0xffff9a00

    and $s3, $s1, $s2
    or $s4, $s1, $s2
    xor $s5, $s1, $s2
    nor $s6, $s1, $s2

    # Exit program
    li $v0, 10
    syscall
```

Screenshot:



Answers:

Register values after execution:

- \$s3 = 0x12341200
- \$s4 = 0xffffde78
- \$s5 = 0xedcbcc78
- \$s6 = 0x00002187

Task 3

Assembly Code:

```
.data

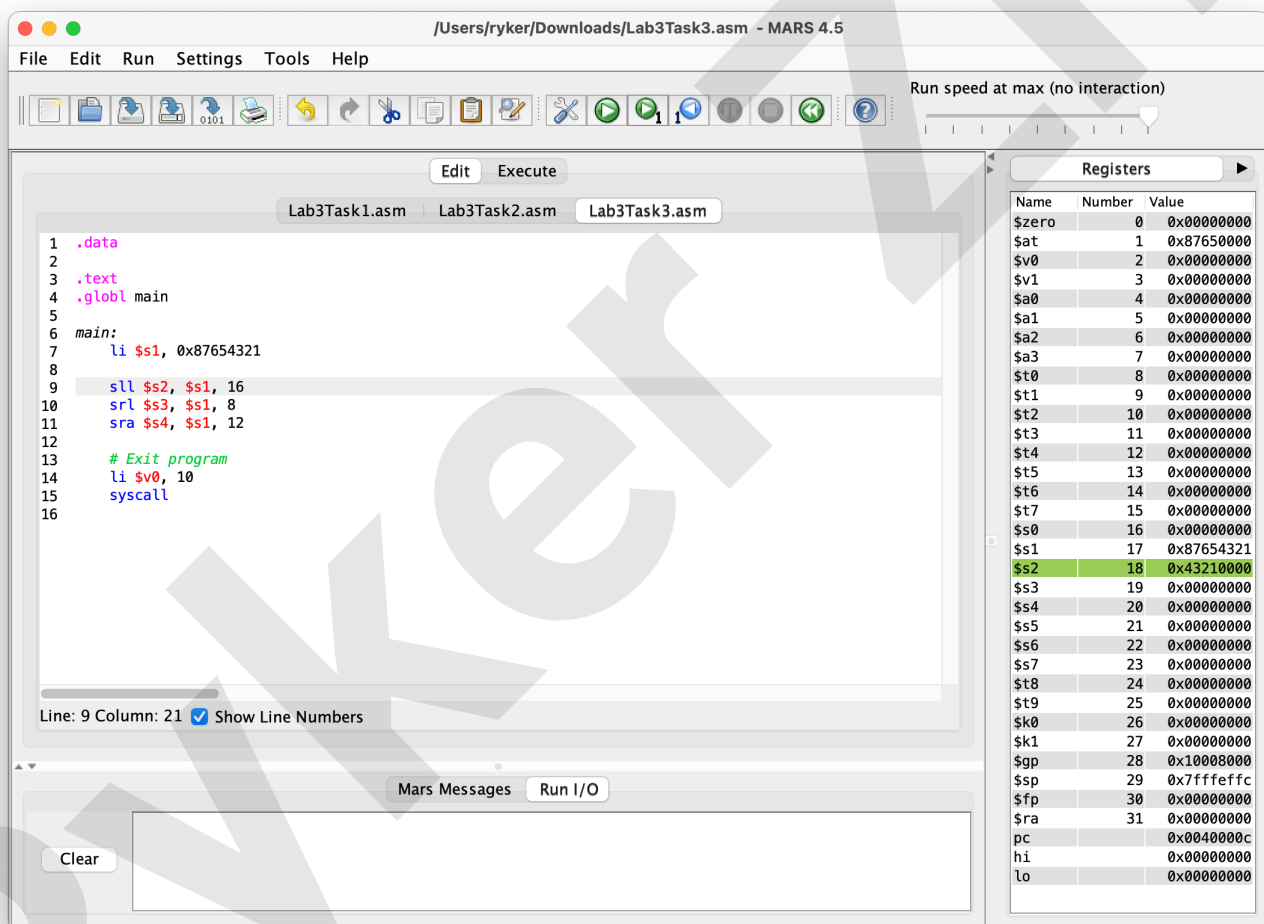
.text
.globl main

main:
    li $s1, 0x87654321

    sll $s2, $s1, 16
    srl $s3, $s1, 8
    sra $s4, $s1, 12

    # Exit program
    li $v0, 10
    syscall
```

Screenshot:



Answers:

Register values after execution:

- \$s2 = 0x43210000
- \$s3 = 0x00876543
- \$s4 = 0xffff87654

Assembly Code:

```
.data
promptChar: .ascii "\nEnter an alphabetic character: "
resultChar: .ascii "\nConverted character: "

.text
.globl main

main:
    # Print prompt
    li $v0, 4
    la $a0, promptChar
    syscall

    # Read character
    li $v0, 12
    syscall
    move $t0, $v0

    # Check if lower case
    li $t1, 'a'
    li $t2, 'z'
    bgt $t0, $t2, upper_case
    blt $t0, $t1, upper_case

    # Convert to upper case
    li $t3, 0x20
    sub $t0, $t0, $t3
```

```
j print_result

upper_case:
    # Check if upper case
    li $t1, 'A'
    li $t2, 'Z'
    bgt $t0, $t2, print_result
    blt $t0, $t1, print_result

    # Convert to lower case
    add $t0, $t0, $t3

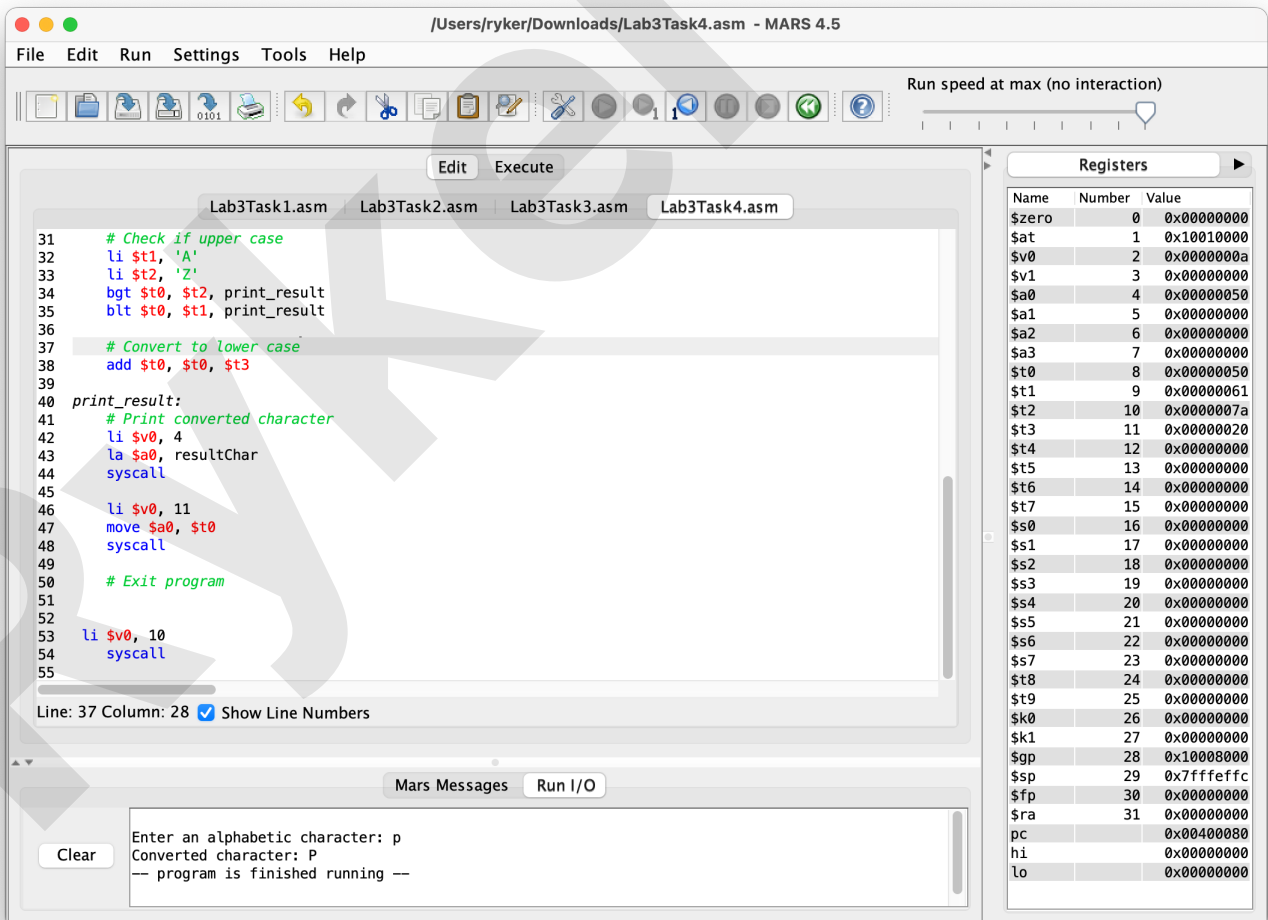
print_result:
    # Print converted character
    li $v0, 4
    la $a0, resultChar
    syscall

    li $v0, 11
    move $a0, $t0
    syscall

    # Exit program

li $v0, 10
syscall
```

Screenshot:



Task 5

Assembly Code:

```
.data
promptNum: .asciiz "Enter an unsigned number: "
resultNum: .asciiz "Number after swapping bits: "

.text
.globl main

main:
    # Print prompt
    li $v0, 4
    la $a0, promptNum
    syscall

    # Read number
    li $v0, 5
    syscall
    move $t0, $v0

    # Swap bits
    # Extract odd and even bits
```

```
li $t1, 0xAAAAAAAA # Mask for even bits
li $t2, 0x55555555 # Mask for odd bits
and $t3, $t0, $t1 # Even bits
and $t4, $t0, $t2 # Odd bits
srl $t3, $t3, 1 # Shift even bits right
sll $t4, $t4, 1 # Shift odd bits left
or $t0, $t3, $t4 # Combine bits

# Print result
li $v0, 4
la $a0, resultNum
syscall

li $v0, 1
move $a0, $t0
syscall

# Exit program
li $v0, 10
syscall
```

Screenshot:

