CCCN 221 – Computer Architecture

LAB#6 Task5

Task Date: As per BB **Submission Date: As per BB**

Student Name: Amin Yahya Selhabi **Student ID: 2140632**

Note: Student must attach the code and the screenshot of the Final output using MIPS or Qtsmpm. ***Copy and cheating will be mark "0"***

Marks:

Exercises	1	2	3	Total
Allocated	1	1	1	3
Obtained				
CLO, PLO, SO	3.1, V3, S05	3.1, V3, S05	3.1, V3, S05	

1. Task

Write a program that prompts the user to enter an integer and determines whether it is divisible by 5 and 6 but not both. For example, if your input is 10, the output should be**.** *Read Lab6 sheet.*

(**Hint:** Use the **AND , OR ,X_OR** operators)

screenshot of the Final output using MIPS or Qtsmpm (without screenshot marks will be deducted).

Output Sample.

```
Is 10 divisible by 5 and 6? false
Is 10 divisible by 5 or 6? true
Is 10 divisible by 5 or 6, but not both? True
```

2. Task

Implement the following assembly program in Mars MIPS that performs the following tasks:

Write a program that prints the numbers from 1 to given number N. *Read Lab sheets.*

screenshot of the Final output using MIPS or Qtsmpm (without screenshot marks will be deducted).

3. Implement the following assembly program in Mars MIPS that performs the following tasks:

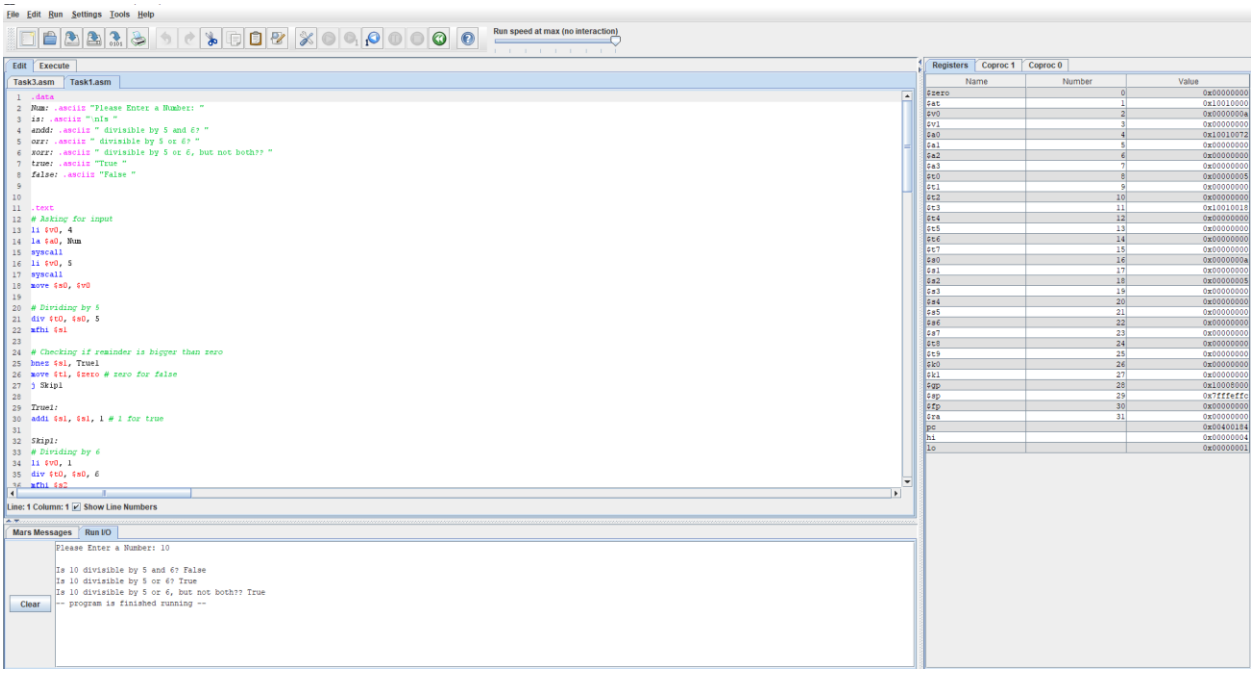
Write a program that asks user to enter Numbers and displays the Average. *Read Lab sheets.*

screenshot of the Final output using MIPS or Qtsmpm (without screenshot marks will be deducted)

Contents

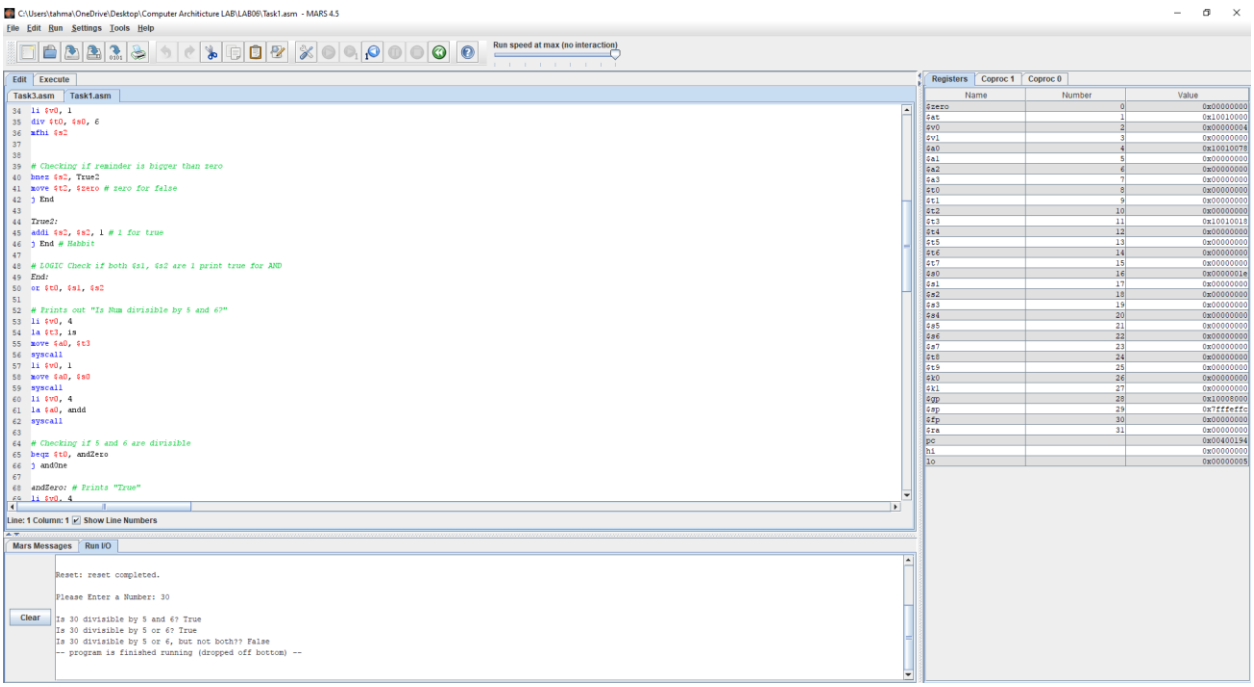
Code included in the end

Task 1:



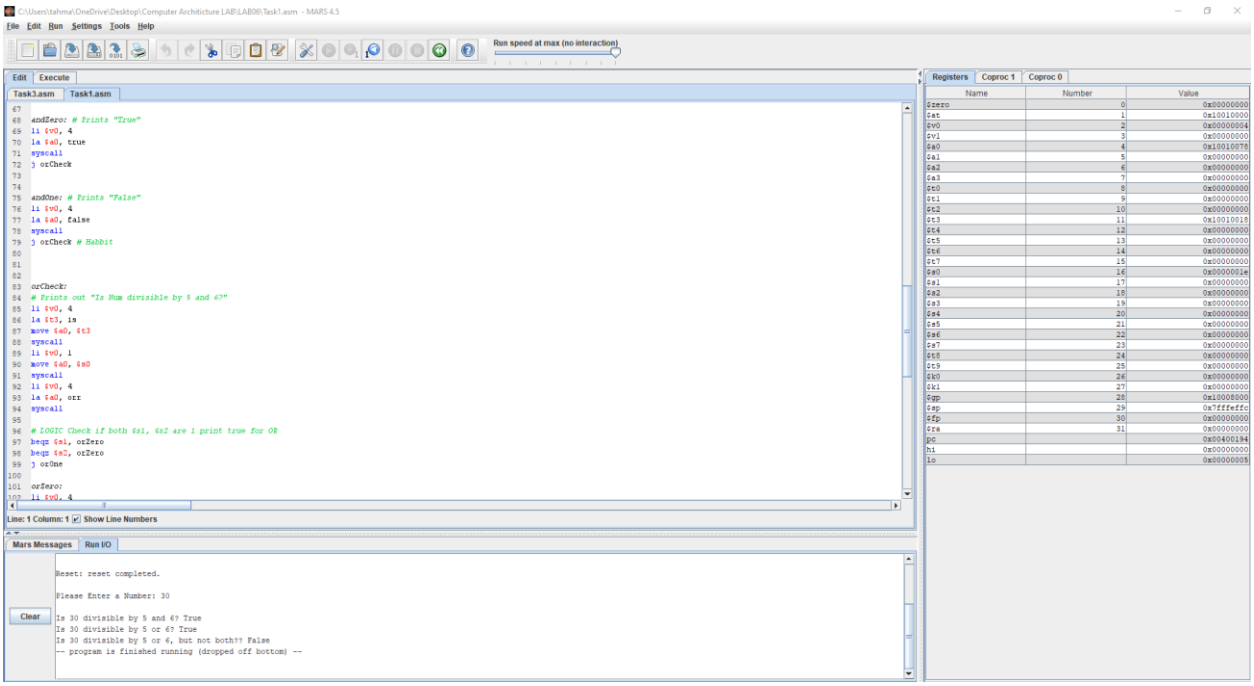
```
.data
Num: .asciiz "Please Enter a Number: "
is: .asciiz ".\nIs "
anddd: .asciiz " divisible by 5 and 6? "
orr: .asciiz " divisible by 5 or 6? "
xorr: .asciiz " divisible by 5 or 6, but not both?? "
true: .asciiz "True "
false: .asciiz "False "

.text
# Asking for input
li $v0, 4
la $a0, Num
syscall
li $v0, 5
syscall
move $s0, $v0

# Dividing by 5
div $t0, $s0, 5
mfhi $s1

# Checking if reminder is bigger than zero
bnez $s1, True1
move $t1, $zero # zero for false
j Skip1

True1:
addi $s1, $s1, 1 # 1 for true

Skip1:
# Dividing by 6
li $v0, 1
div $t0, $s0, 6
mfhi $s2
```

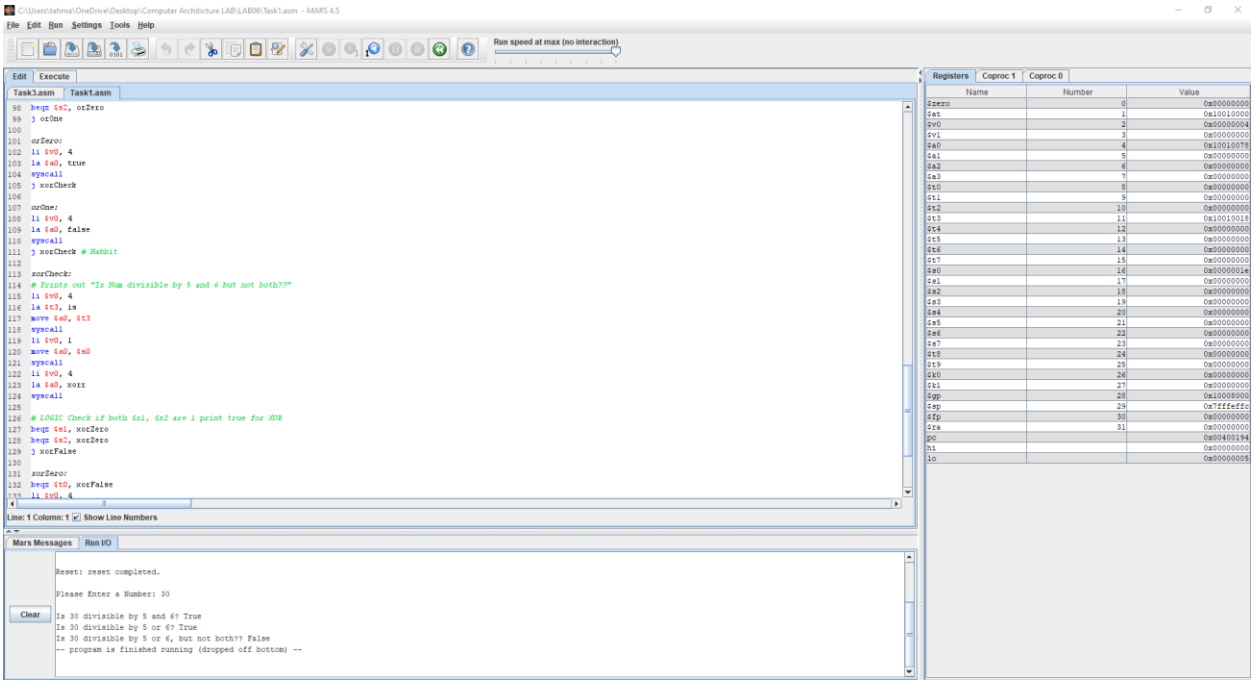
```
Please Enter a Number: 10

Is 10 divisible by 5 and 6? False
Is 10 divisible by 5 or 6? True
Is 10 divisible by 5 or 6, but not both?? True
-- program is finished running --
```

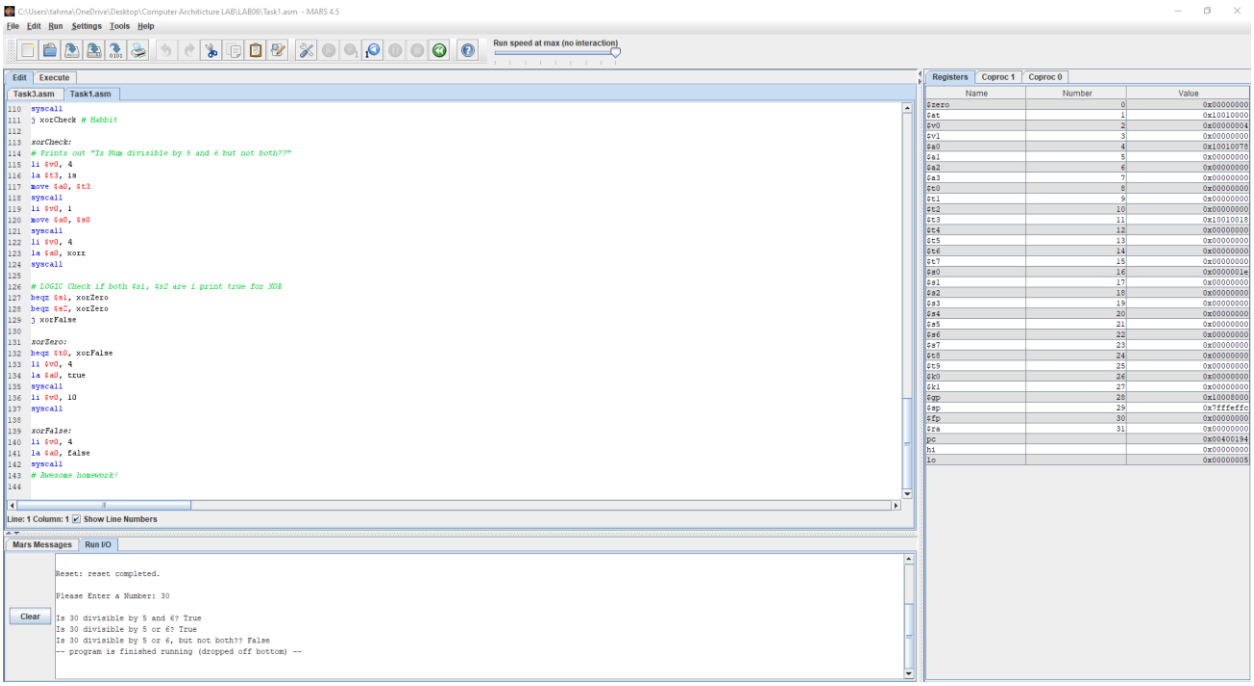


```
li $v0, 1
div $t0, $s0, 6
mfhi $s2


# Checking if reminder is bigger than zero
bnez $s0, True2
move $t2, $zero # zero for false
j End

True2:
addi $s2, $s2, 1 # 1 for true
j End # Habbit

# LOGIC Check if both $s1, $s2 are 1 print true for AND
End:
or $t0, $s1, $s2

# Prints out "Is Num divisible by 5 and 6?"
li $v0, 4
la $t3, is
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, anddd
syscall

# Checking if 5 and 6 are divisible
beqz $t0, andZero
j and0ne

andZero: # Prints "True"
li $v0, 4
```

```
Reset: reset completed.

Please Enter a Number: 30

Is 30 divisible by 5 and 6? True
Is 30 divisible by 5 or 6? True
Is 30 divisible by 5 or 6, but not both?? False
-- program is finished running (dropped off bottom) --
```


C:\Users\tahma\OneDrive\Desktop\Computer Architicture LAB\LAB06\Task1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Task3.asm Task1.asm
andZero: # Prints "True"
li $v0, 4
la $a0, true
syscall
j orCheck
andOne: # Prints "False"
li $v0, 4
la $a0, false
syscall
j orCheck # Habbit
orCheck:
Prints out "Is Num divisible by 5 and 6?"
li $v0, 4
la $t3, is
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, orr
syscall
LOGIC Check if both $s1, $s2 are 1 print true for OR
beqz $s1, orZero
beqz $s2, orZero
j orOne
orZero:
li $v0, 4
Line: 1 Column: 1 Show Line Numbers
Mars Messages Run I/O
Clear
Reset: reset completed.
Please Enter a Number: 30
Is 30 divisible by 5 and 6? True
Is 30 divisible by 5 or 6? True
Is 30 divisible by 5 or 6, but not both?? False
-- program is finished running (dropped off bottom) --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x00000004
$v1 3 0x00000000
$a0 4 0x10010078
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000000
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x10010018
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x0000001e
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400194
hi 0x00000000
lo 0x00000005


C:\Users\tahma\OneDrive\Desktop\Computer Architicture LAB\LAB06\Task1.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Task3.asm Task1.asm
beqz $s2, orZero
j orOne
orZero:
li $v0, 4
la $a0, true
syscall
j xorCheck
orOne:
li $v0, 4
la $a0, false
syscall
j xorCheck # Habbit
xorCheck:
Prints out "Is Num divisible by 5 and 6 but not both??"
li $v0, 4
la $t3, is
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, xorr
syscall
LOGIC Check if both $s1, $s2 are 1 print true for XOR
beqz $s1, xorZero
beqz $s2, xorZero
j xorFalse
xorZero:
beqz $t0, xorFalse
Line: 1 Column: 1 Show Line Numbers
Mars Messages Run I/O
Clear
Reset: reset completed.
Please Enter a Number: 30
Is 30 divisible by 5 and 6? True
Is 30 divisible by 5 or 6? True
Is 30 divisible by 5 or 6, but not both?? False
-- program is finished running (dropped off bottom) --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x00000004
$v1 3 0x00000000
$a0 4 0x10010078
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000000
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x10010018
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x0000001e
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400194
hi 0x00000000
lo 0x00000005

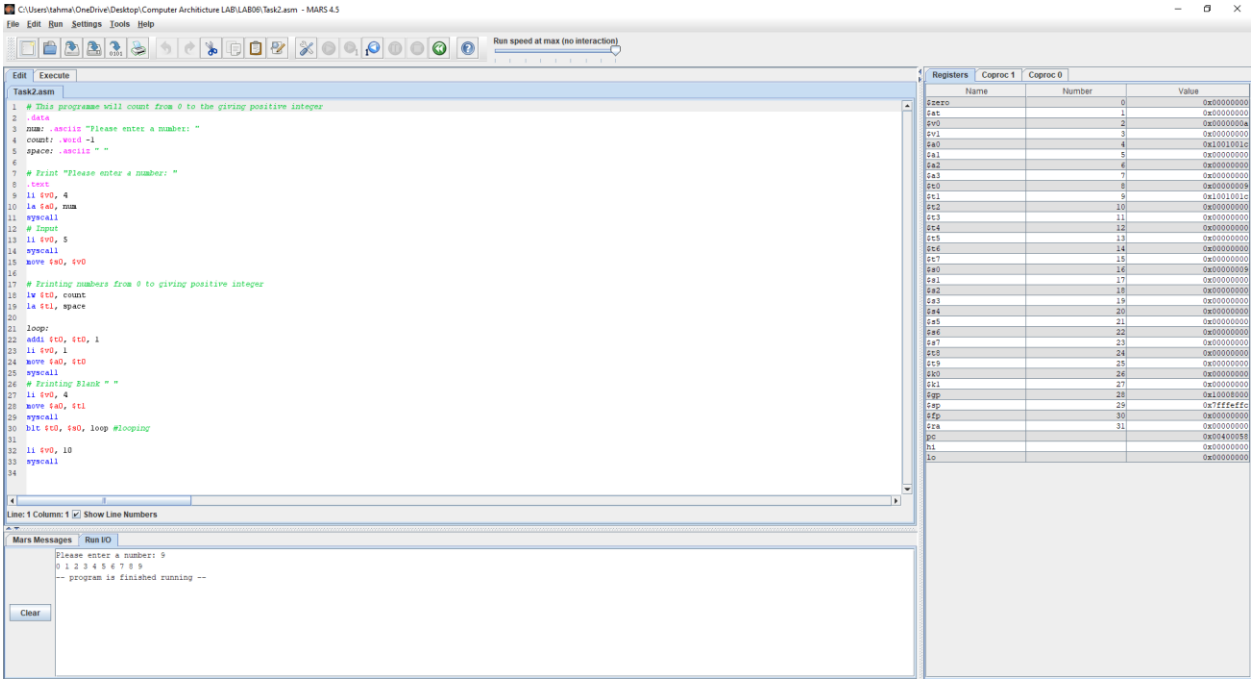


```
syscall
j xorCheck # Habbit

xorCheck:
# Prints out "Is Num divisible by 5 and 6 but not both??"
li $v0, 4
la $t3, is
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, xorr
syscall

# LOGIC Check if both $s1, $s2 are 1 print true for XOR
beqz $s1, xorZero
beqz $s2, xorZero
j xorFalse

xorZero:
beqz $t0, xorFalse
li $v0, 4
la $a0, true
syscall
li $v0, 10
syscall

xorFalse:
li $v0, 4
la $a0, false
syscall
# Awesome homework!
```

```
Reset: reset completed.

Please Enter a Number: 30

Is 30 divisible by 5 and 6? True
Is 30 divisible by 5 or 6? True
Is 30 divisible by 5 or 6, but not both?? False
-- program is finished running (dropped off bottom) --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010078
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x10010018
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x0000001e
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400194
hi		0x00000000
lo		0x00000005

Task 2:

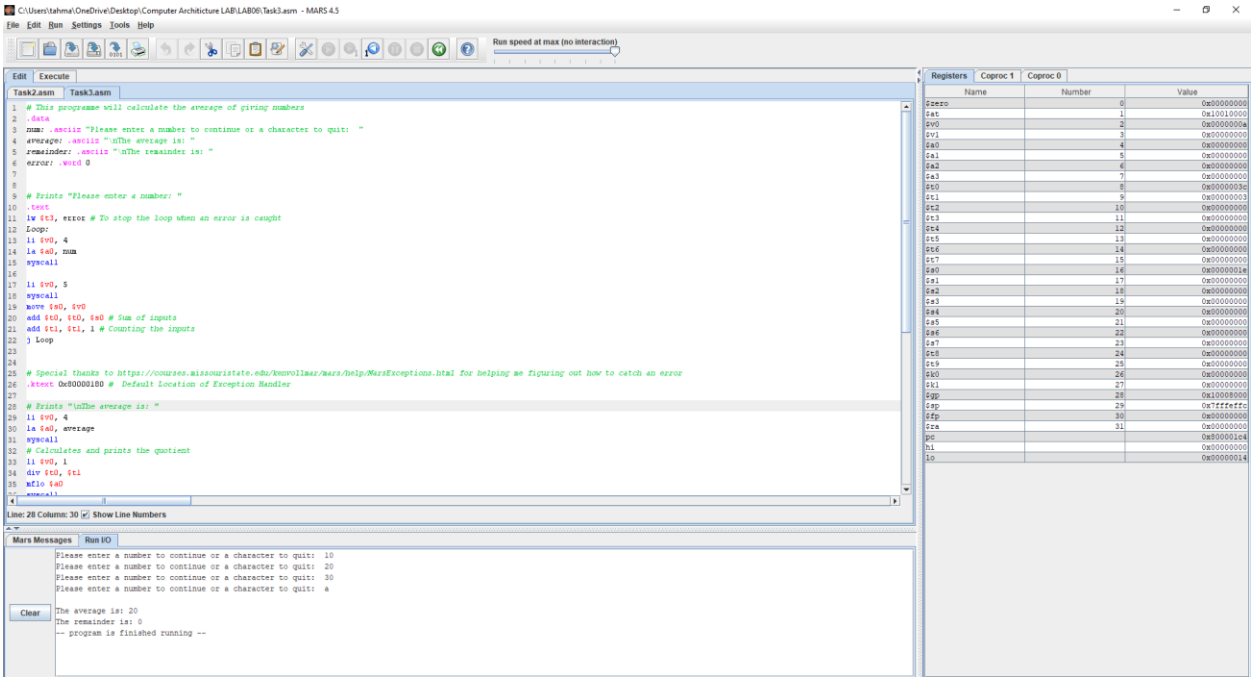


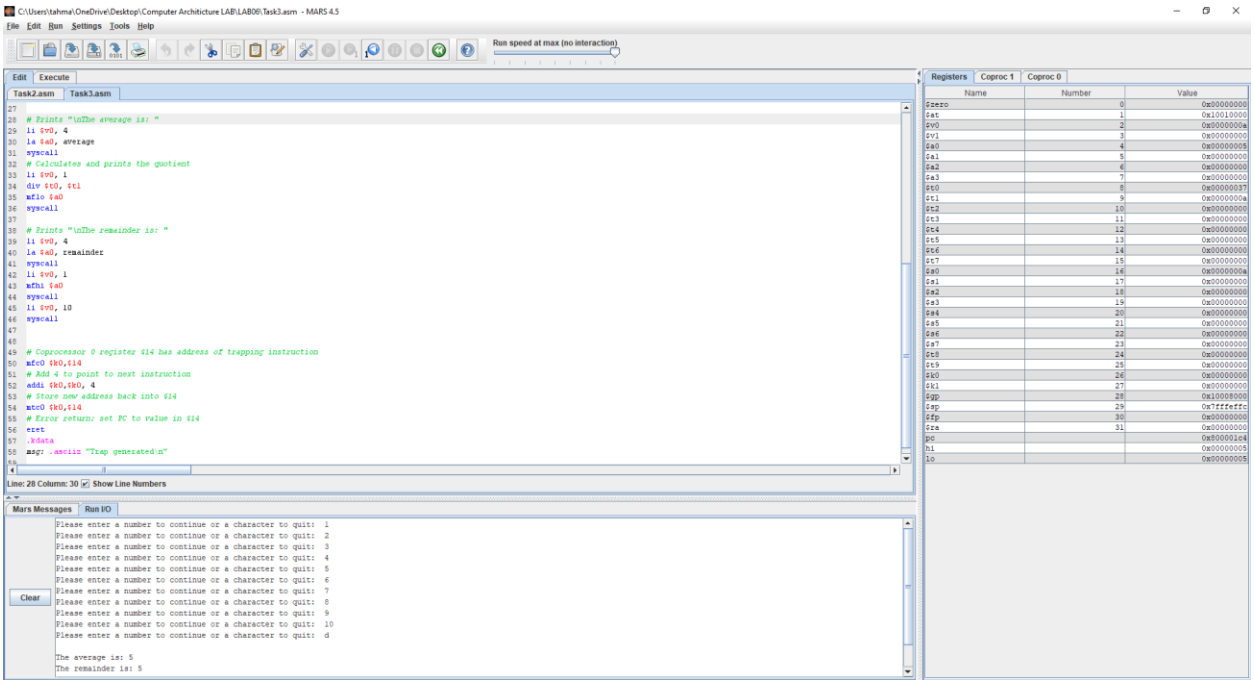
```
# This programme will count from 0 to the giving positive integer
.data
num: .asciiz "Please enter a number: "
count: .word -1
space: .asciiz " "

# Print "Please enter a number: "
.text
li $v0, 4
la $a0, num
syscall
# Input
li $v0, 5
syscall
move $s0, $v0

# Printing numbers from 0 to giving positive integer
lw $t0, count
la $t1, space

loop:
addi $t0, $t0, 1
li $v0, 1
move $a0, $t0
syscall
# Printing Blank " "
li $v0, 4
move $a0, $t1
syscall
blt $t0, $s0, loop #looping

li $v0, 10
syscall
```

```
Please enter a number: 9
0 1 2 3 4 5 6 7 8 9
-- program is finished running --
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x1001001c
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000009
$t1	9	0x1001001c
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000009
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400058
hi		0x00000000
lo		0x00000000

Task 3:


C:\Users\tahma\OneDrive\Desktop\Computer Architicture LAB\LAB06\Task3.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Task2.asm Task3.asm
This programme will calculate the average of giving numbers
.data
num: .asciiz "Please enter a number to continue or a character to quit: "
average: .asciiz "\nThe average is: "
remainder: .asciiz "\nThe remainder is: "
error: .word 0
Prints "Please enter a number: "
.text
lw $t3, error # To stop the loop when an error is caught
Loop:
li $v0, 4
la $a0, num
syscall
li $v0, 5
syscall
move $s0, $v0
add $t0, $t0, $s0 # Sum of inputs
add $t1, $t1, 1 # Counting the inputs
j Loop
Special thanks to https://courses.missouristate.edu/kenvollmar/mars/help/MarsExceptions.html for helping me figuring out how to catch an error
.ktext 0x80000180 # Default location of Exception Handler
Prints "\nThe average is: "
li $v0, 4
la $a0, average
syscall
Calculates and prints the quotient
li $v0, 1
div $t0, $t1
mflo $a0
Line: 28 Column: 30 Show Line Numbers
Mars Messages Run I/O
Clear
Please enter a number to continue or a character to quit: 10
Please enter a number to continue or a character to quit: 20
Please enter a number to continue or a character to quit: 30
Please enter a number to continue or a character to quit: a
The average is: 20
The remainder is: 0
-- program is finished running --
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x0000000a
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x0000003c
$t1 9 0x00000003
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x0000001e
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x800001e4
hi 0x00000000
lo 0x00000014


C:\Users\tahma\OneDrive\Desktop\Computer Architicture LAB\LAB06\Task3.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Task2.asm Task3.asm
Prints "\nThe average is: "
li $v0, 4
la $a0, average
syscall
Calculates and prints the quotient
li $v0, 1
div $t0, $t1
mflo $a0
syscall
Prints "\nThe remainder is: "
li $v0, 4
la $a0, remainder
syscall
li $v0, 1
mfhi $a0
syscall
li $v0, 10
syscall
Coprocessor 0 register $14 has address of trapping instruction
mfc0 $k0,$14
# Add 4 to point to next instruction
addi $k0,$k0, 4
# Store new address back into $14
mtc0 $k0,$14
Error return; set PC to value in $14
eret
.kdata
msg: .asciiz "Trap generated\n"
Line: 28 Column: 30 Show Line Numbers
Mars Messages Run I/O
Clear
Please enter a number to continue or a character to quit: 1
Please enter a number to continue or a character to quit: 2
Please enter a number to continue or a character to quit: 3
Please enter a number to continue or a character to quit: 4
Please enter a number to continue or a character to quit: 5
Please enter a number to continue or a character to quit: 6
Please enter a number to continue or a character to quit: 7
Please enter a number to continue or a character to quit: 8
Please enter a number to continue or a character to quit: 9
Please enter a number to continue or a character to quit: 10
Please enter a number to continue or a character to quit: d
The average is: 5
The remainder is: 5
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x10010000
$v0 2 0x0000000a
$v1 3 0x00000000
$a0 4 0x00000005
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000037
$t1 9 0x0000000a
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x0000000a
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x800001e4
hi 0x00000005
lo 0x00000005

Task 1 code:

```
.data
Num: .asciiz "Please Enter a Number: "
is: .asciiz "\nIs "
andd: .asciiz " divisible by 5 and 6? "
orr: .asciiz " divisible by 5 or 6? "
xorr: .asciiz " divisible by 5 or 6, but not both?? "
true: .asciiz "True "
false: .asciiz "False "



.text
# Asking for input
li $v0, 4
la $a0, Num
syscall
li $v0, 5
syscall
move $s0, $v0

# Dividing by 5
div $t0, $s0, 5
mfhi $s1

# Checking if reminder is bigger than zero
bnez $s1, True1
move $t1, $zero # zero for false
j Skip1
```

```
True1:
addi $s1, $s1, 1 # 1 for true

Skip1:
# Dividing by 6
li $v0, 1
div $t0, $s0, 6
mfhi $s2


# Checking if reminder is bigger than zero
bnez $s2, True2
move $t2, $zero # zero for false
j End

True2:
addi $s2, $s2, 1 # 1 for true
j End # Habbit

# LOGIC Check if both $s1, $s2 are 1 print true for AND
End:
or $t0, $s1, $s2

# Prints out "Is Num divisible by 5 and 6?"
li $v0, 4
la $t3, is
move $a0, $t3
syscall
li $v0, 1
```

```
move $a0, $s0
syscall
li $v0, 4
la $a0, andd
syscall

# Checking if 5 and 6 are divisible
beqz $t0, andZero
j andOne

andZero: # Prints "True"
li $v0, 4
la $a0, true
syscall
j orCheck



andOne: # Prints "False"
li $v0, 4
la $a0, false
syscall
j orCheck # Habbit



orCheck:
# Prints out "Is Num divisible by 5 and 6?"
li $v0, 4
la $t3, is
```

```
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, orr
syscall

# LOGIC Check if both $s1, $s2 are 1 print true for OR
beqz $s1, orZero
beqz $s2, orZero
j orOne

orZero:
li $v0, 4
la $a0, true
syscall
j xorCheck

orOne:
li $v0, 4
la $a0, false
syscall
j xorCheck # Habbit

xorCheck:
# Prints out "Is Num divisible by 5 and 6 but not both??"
li $v0, 4
```

```
la $t3, is
move $a0, $t3
syscall
li $v0, 1
move $a0, $s0
syscall
li $v0, 4
la $a0, xorr
syscall

# LOGIC Check if both $s1, $s2 are 1 print true for XOR
beqz $s1, xorZero
beqz $s2, xorZero
j xorFalse

xorZero:
beqz $t0, xorFalse
li $v0, 4
la $a0, true
syscall
li $v0, 10
syscall

xorFalse:
li $v0, 4
la $a0, false
syscall
# Awesome homework!
```

Task 2 code:

```
# This programme will count from 0 to the giving positive integer

.data

num: .asciiz "Please enter a number: "

count: .word -1

space: .asciiz " "


# Print "Please enter a number: "

.text

li $v0, 4

la $a0, num

syscall

# Input

li $v0, 5

syscall

move $s0, $v0


# Printing numbers from 0 to giving positive integer

lw $t0, count

la $t1, space


loop:

addi $t0, $t0, 1

li $v0, 1

move $a0, $t0

syscall

# Printing Blank " "

li $v0, 4

move $a0, $t1
```

```
syscall
blt $t0, $s0, loop #looping

li $v0, 10
syscall
```

Task 3 code:

```
# This programme will calculate the average of giving numbers
.data
num: .asciiz "Please enter a number to continue or a character to quit:  "
average: .asciiz "\nThe average is: "
remainder: .asciiz "\nThe remainder is: "
error: .word 0


# Prints "Please enter a number: "
.text
lw $t3, error # To stop the loop when an error is caught
Loop:
li $v0, 4
la $a0, num
syscall

li $v0, 5
syscall
move $s0, $v0
add $t0, $t0, $s0 # Sum of inputs
add $t1, $t1, 1 # Counting the inputs
j Loop
```

```
# Special thanks to https://courses.missouristate.edu/kenvollmar/mars/help/MarsExceptions.html for helping me figuring out how to catch an error
.ktext 0x80000180 #  Default Location of Exception Handler

# Prints "\nThe average is: "
li $v0, 4
la $a0, average
syscall
# Calculates and prints the quotient
li $v0, 1
div $t0, $t1
mflo $a0
syscall

# Prints "\nThe remainder is: "
li $v0, 4
la $a0, remainder
syscall
li $v0, 1
mfhi $a0
syscall
li $v0, 10
syscall


# Coprocessor 0 register $14 has address of trapping instruction
mfc0 $k0,$14
# Add 4 to point to next instruction
addi $k0,$k0, 4
```

```
# Store new address back into $14

mtc0 $k0,$14

# Error return; set PC to value in $14

eret

.kdata

msg: .asciiz "Trap generated\n"
```

References

University, M. S. (n.d.). Retrieved from https://courses.missouristate.edu/kenvollmar/mars/help/MarsExceptions.html