 **Northwestern Polytechnic University**

**EE488 - Computer Architecture**

**Homework Assignment #5**

**Due day: 11/24/2021**

**Instruction:**

1. **Push the answer sheet to GitHub in word file**
2. **Overdue homework submission could not be accepted.**
3. **Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**
4. Implement a subprogram that prompts the user for *3* numbers, finds the median (middle value) of the *3*, and returns that value to the calling program.

**# Utils.asm**

.text

PrintNewLine:

li $v0, 4

la $a0, \_\_PNL\_newline

syscall

jr $ra

.data

\_\_PNL\_newline: .asciiz "\n"

.text

PrintInt:

li $v0, 4

syscall

move $a0, $a1

li $v0, 1

syscall

jr $ra

.text

PromptInt:

addi $v0, $zero, 4

syscall

move $a0, $a1

addi $v0, $zero, 5

syscall

jr $ra

.text

Exit:

li $v0, 10

syscall

.text

PrintString:

addi $v0, $zero, 4

syscall

jr $ra

Median:

la $v0, ($s1)

jr $ra

**# median.asm:**

.data

prompt1: .asciiz "Enter first number: "

prompt2: .asciiz "Enter second number: "

prompt3: .asciiz "Enter third number: "

result: .asciiz "Median = "

.text

main:

la $a0, prompt1

jal PromptInt

move $s0, $v0

jal PrintNewLine

la $a0, prompt2

jal PromptInt

move $s1, $v0

jal PrintNewLine

la $a0, prompt3

jal PromptInt

move $s2, $v0

jal PrintNewLine

jal Median

move $s0, $v0

# Print the result

la $a0, result

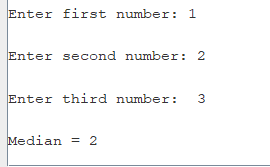
la $a1, ($s0)

jal PrintInt

jal PrintNewLine

jal Exit

.include "utils.asm"



1. Implement a recursive program that takes in a number and finds the square of that number through addition. For example if the number *3* is entered, you would add *3+3+3=9*. If *4* is entered you would add *4+4+4+4=16*. This program must be implemented using recursion to add the numbers together.

.data

num: .asciiz "Enter a number : "

sum: .asciiz "Square of given number through addition: "

.text

#Enter number prompt

li $v0,4

la $a0,num

syscall

#Read the number

li $v0,5

syscall

move $s0,$v0

move $t0,$v0

li $s1,0

#loop for recursive addition

loop:

blez $t0,exit

add $s1,$s1,$s0

addi $t0,$t0,-1

j loop

exit:

li $v0,4

la $a0,sum

syscall

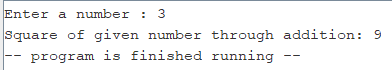
li $v0,1

move $a0,$s1

syscall

li $v0,10

syscall



1. Write a recursive program to calculate factorial numbers. Use the definition of factorial as *F(n) = n \* F(n-1)*

.data

prompt: .asciiz "Input an integer n:"

result: .asciiz "Factorial of n, Fact(n) = "

.text

main:

li $v0, 4

la $a0, prompt

syscall

# Read input n

li $v0, 5

syscall

# function call

move $a0, $v0

jal factorial # jump to factorial

move $t0, $v0

li $v0, 4

la $a0, result

syscall

# print the result

li $v0, 1

move $a0, $t0

syscall

li $v0, 10

syscall

.text

factorial:

# adjust stack pointer to store return address and argument

addi $sp, $sp, -8

sw $s0, 4($sp)

sw $ra, 0($sp)

bne $a0, 0, else

addi $v0, $zero, 1 # return 1

j factorial\_return

else:

move $s0, $a0

addi $a0, $a0, -1 # n -= 1

jal factorial

multu $s0, $v0 # return n\*Fact(n-1)

mflo $v0

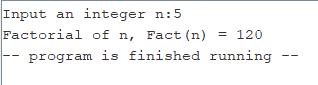
factorial\_return:

lw $s0, 4($sp)

lw $ra, 0($sp)

addi $sp, $sp, 8

jr $ra

**

1. The following pseudo code converts an input value of a single decimal number from

*1 ≤ n ≥15* into a single hexadecimal digit. Translate this pseudo code into MIPS assembly.

*main{*

*String a[16]*

*a[0] = "0x0"*

*a[1] = "0x1"*

*a[2] = "0x2"*

*a[3] = "0x3"*

*a[4] = "0x4"*

*a[5] = "0x5"*

*a[6] = "0x6"*

*a[7] = "0x7"*

*a[8] = "0x8"*

*a[9] = "0x9"*

*a[10] = "0xa"*

*a[11] = "0xb"*

*a[12] = "0xc"*

*a[13] = "0xd"*

*a[14] = "0xe"*

*a[15] = "0xf"*

*int i = prompt("Enter a number from 0 to 15 ")*

*print("your number is " + a[i])*

*}*

.data

prompt: .asciiz "Enter a number from 0 to 15: "

result: .asciiz "Hexadecimal number is "

a: .asciiz "0x0","0x1","0x2","0x3","0x4","0x5","0x6","0x7","0x8","0x9","0xa","0xb","0xc","0xd","0xe","0xf"

.text

.globl main

main:

li $v0,4

la $a0, prompt

syscall

li $v0,5

syscall

move $t0, $v0

li $v0,4

la $a0,result

syscall

la $t3, a

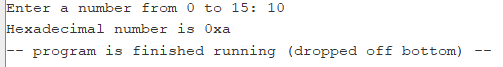
sll $t2, $t0, 2

add $t1, $t2, $t3

li $v0,4

la $a0, ($t1)

syscall



1. The following pseudo code program calculates the Fibonacci numbers from *1…n*, and stores them in an array. Translate this pseudo code into MIPS assembly, and use the PrintIntArray subprogram to print the results.

*main{*

*int size = PromptInt(“Enter a max Fibonacci number to calc: “)*

*int Fibonacci[size]*

*Fibonacci[0] = 0*

*Fibonacci[1] = 1*

*for (int i = 2; i < size; i++){*

*Fibonacci[i] = Fibonacci[i-1] + Fibonacci[i-2]*

*}*

*PrintIntArray(Fibonacci, size)*

*}*

.data

message: .asciiz "Enter a max Fibonacci number to calculate:"

.text

main:

la $a0,message

li $v0,4

syscall

li $v0,5

syscall

move $t0,$v0

li $a0,0

li $v0,1

syscall

li $a0,32

li $v0,11

syscall

li $a0,1

li $v0,1

syscall

li $a0,32

li $v0,11

syscall

jal printarray

li $v0,10

syscall

printarray:

addi $sp,$sp,-4

sw $t0,0($sp)

li $t1,2

blt $t1,$t0,loop

ret:

jr $ra

loop:

li $t2,0

li $t3,1

fibonacci:

add $t4,$t3,$t2

move $a0,$t4

li $v0,1

syscall

li $a0,32

li $v0,11

syscall

move $t2,$t3

move $t3,$t4

addi $t1,$t1,1

blt $t1,$t0,fibonacci

j ret

