

Computer Architecture Lab – Week 4's report

Full name	Student ID	Works
Trần Quốc Việt	1953097	1 + combine
Lý Kim Phong	1952916	3 (2 + 5 done in lab)
Nguyễn Đình Khương Duy	1952207	4 (2 + 5 done in lab)

Question 1.

```
.text
.globl main

main:
    addi $t1, $zero, 0 #sum register
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msg
    syscall

loop :
    li $v0, 5 # read_int syscall code = 5
    syscall
    move $t0, $v0 #t0 = input

    bltz $t0, EXIT #compare if input is greater than 0

    add $t1, $t1, $t0

    li $v0, 4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msg
    syscall
    j loop

EXIT :
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg3 # load the address of msg
    syscall

    li $v0, 1
    move $a0, $t1
    syscall

.data
msg1: .asciiz "Enter input : "
msg2: .asciiz "please input an another integer number : "
msg3: .asciiz "Sum of previous inputs is : "
```

Question 2.

```
.text
.globl main
main:
    addi $t1, $zero, 0 #a
    addi $t2, $t2, 10 #b
    addi $t3, $t3, 5 #c

    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msgsyscall
    syscall

    li $v0, 5 # read_int syscall code = 5
    syscall
    move $t0, $v0 #t0 = input

    beq $t0, 0, case0 #check if input == 0
    beq $t0, 1, case1 #check if input == 1
    beq $t0, 2, case2 #check if input == 2

    li $v0, 4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msgsyscall
    syscall
    j EXIT

case0 :
    add $t1, $t2, $t3
    j END
case1 :
    sub $t1, $t2, $t3
    j END
case2 :
    sub $t1, $t3, $t2
    j END
END :
    li $v0, 1
    move $a0, $t1
    syscall

EXIT :

.data
msg1: .asciiz "Enter input : "
msg2: .asciiz "please input an another integer number\n"
```

Question 3.

```
.text
```

```

        .globl main
main:
    # Print string msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg1      # load the address of msg1
    syscall

    # Get input A from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0,$v0

    # Initialize registers
    la    $t3,arr
    add   $t1, $t3, $zero
    add   $t4, $zero, $zero
    add   $t2, $zero, $zero

    # Main loop body
loop:
    beq $t4, 10, exit1    #
    lw   $t2, 0($t1)
    beq  $t2, $t0, found
    addi $t1, $t1, 4
    addi $t4, $t4, 1
    j    loop

found:
    li $v0,1
    move $a0, $t4
    syscall
    j    exit

exit1:
    li $v0,4          # print_string syscall code = 4
    la $a0, msg2      # load the address of msg
    syscall

exit:

    .data
arr: .word 1, 12, 0, -3, 99, 48, -17, -9, 20, 15
msg1: .asciiz "Enter input: "
msg2: .asciiz "Input is not found in the array "

```

Question 4.

```

.text
        .globl      main
main:
    # Print string msg1

```

```

    li    $v0,4      # print_string syscall code = 4
    la    $a0, msg1   # load the address of msg1
    syscall

    # Get input A from user and save
    li    $v0,5      # read_int syscall code = 5
    syscall
    move  $a1,$v0

# Initialize registers
    la    $a0, arr
    add $a2, $zero, $a0 #clone array
    addi $t3, $zero, 0 #counter
    jal swap
    j Loop

swap:
    sll $t1, $a1, 2
    add $t1, $a2, $t1
    lw $t0, 0($t1)
    lw $t2, 4($t1)
    sw $t2, 0($t1)
    sw $t0, 4($t1)

    jr $ra

    li $v0,4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msg
    syscall

Loop:
    lw $t0, 0($a2)
    li $v0, 1
    move $a0, $t0
    syscall
    li $v0,4      # print_string syscall code = 4
    la $a0, msg3   # load the address of msg1
    syscall
    addi $a2, $a2, 4
    addi $t3, $t3, 1
    beq $t3, 10, End
    j Loop

End:

.data
arr: .word 1, 3, 8, 5, 12, 99, 15, 25, 0, 23
msg1: .asciiz "Enter input k: "
msg2: .asciiz "Array after swapped : "
msg3: .asciiz " | "

```

Question 5.

```
.text
.globl main
main:
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msg
    syscall

    li $v0, 5 # read_int syscall code = 5
    syscall
    move $a0, $v0 # a0 = input
    jal fact
    j EXIT

fact:
    addi $sp, $sp, -8 # adjust stack for 2 items
    sw $ra, 4($sp) # save return address
    sw $a0, 0($sp) # save argument
    slti $t0, $a0, 1 # test for n < 1
    beq $t0, $zero, L1
    addi $v1, $zero, 1 # if so, result is 1
    addi $sp, $sp, 8 # pop 2 items from stack
    jr $ra # and return

L1:
    addi $a0, $a0, -1 # else decrement n
    jal fact # recursive call
    lw $a0, 0($sp) # restore original n
    lw $ra, 4($sp) # and return address
    addi $sp, $sp, 8 # pop 2 items from stack
    mul $v1, $a0, $v1 # multiply to get result
    jr $ra # and return

EXIT :
    li $v0, 1
    move $a0, $v1
    syscall

.data
msg1: .asciiz "Enter input : "
```