

Computer Architecture Lab – Week 3's report

Full name	Student ID	Works
Trần Quốc Việt	1953097	Done in lab session
Lý Kim Phong	1952916	Done in lab session
Nguyễn Đình Khương Duy	1952207	Done in lab session

Question 1.

```
.text
    .globl main
main:

#Create temp register for array
    la $s1, array
    addi $s0, $zero, 0
    add $s0, $s1, $zero
    addi $t2, $zero, 0 #counter
    addi $t3, $zero, 0 #sum register

Loop :
    lw $t1, 0($s0)
    addi $s0, $s0, 4
    addi $t2, $t2, 1
    add $t3, $t3, $t1
    beq $t2, 10, Exit
    j Loop
Exit :
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msgsyscall
    syscall
    li $v0, 1
    move $a0, $t3
    syscall

.data
array: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
msg1: .asciiz "Sum of 10 array's elements is : "
```

Question 2.

```
.text
    .globl main
```

```

main:

#Create temp register for array
    la $s1, array
    addi $s0, $zero, 0
    add $s0, $s1, $zero
    addi $s2, $zero, 0
    addi $s2, $s1, 4
    addi $t2, $zero, 0 #counter
    addi $t3, $zero, 0 #sum register for even
    addi $t4, $zero, 0 #sum register for odd

Loop :
    lw $t0, 0($s0)
    lw $t1, 0($s2)
    add $t3, $t3, $t0
    add $t4, $t4, $t1
    addi $s0, $s0, 8
    addi $s2, $s2, 8
    addi $t2, $t2, 1
    beq $t2, 5, Exit
    j Loop
Exit :
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msgsyscall
    syscall
    li $v0, 1
    move $a0, $t3
    syscall

    li $v0, 4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msgsyscall
    syscall
    li $v0, 1
    move $a0, $t4
    syscall

.data
array: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
msg1: .asciiz "Sum of even array's elements is : "
msg2: .asciiz "\nSum of odd array's elements is : "

```

Question 3.

```

.text
    .globl main
main:

```

```

#Create temp register for array
    la $s1, array
    addi $s0, $zero, 0
    add $s0, $s1, $zero
    addi $t2, $zero, 0 #counter
    addi $t3, $zero, 0 #sum register

#Print string to require input array
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg1 # load the address of msg
    syscall

# Get input A from user and save
Loop :    li $v0, 5 # read_int syscall code = 5
        syscall
        move $t1, $v0
        sw $t1, 0($s0)
        add $t3, $t3, $t1
        addi $s0, $s0, 4
        addi $t2, $t2, 1
        beq $t2, 10, Exit
        j Loop
Exit :
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msg
    syscall

    li $v0, 1
    move $a0, $t3
    syscall

.data
array: .word 0:10
msg1: .asciiz "Enter 10 array's elements : "
msg2: .asciiz "Sum of 10 array's elements is : "

```

Question 4.

```

.text
    .globl main
main:

#Create temp register for array
    la $s1, array
    add $s0, $s1, $zero

#Print string to require input array
    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 $t1, $v0 # $t1 now store the index
        addi $t2, $zero, 4
        mul $t1, $t1, $t2
        add $s0, $s0, $t1
        lw $t2, 0($s0)

Exit :
        li $v0, 4 # print_string syscall code = 4
        la $a0, msg2 # load the address of msg
        syscall
        li $v0, 1
        move $a0, $t2
        syscall

.data
array: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
msg1: .asciiz "Enter the index : "
msg2: .asciiz "\nThe element is : "

```

Question 5.

```

.text
.globl main
main:

#Create temp register for array
        la $s1, array
        addi $s0, $zero, 0
        add $s0, $s1, $zero
        addi $t2, $zero, 0 #counter

#Print string to require input array
        li $v0, 4 # print_string syscall code = 4
        la $a0, msg1 # load the address of msg
        syscall

# Get input A from user and save
Loop :   li $v0, 5 # read_int syscall code = 5
        syscall
        move $t1, $v0
        sw $t1, 0($s0)
        addi $s0, $s0, 4
        addi $t2, $t2, 1
        beq $t2, 10, Exit
        j Loop

```

```

Exit :
    addi $t2, $zero, 0 #re-initialize counter
    addi $s2, $s1, 0
    subi $s0, $s0, 4

Loop2: #swap
    lw $t3, 0($s2)
    lw $t4, 0($s0)
    sw $t3, 0($s0)
    sw $t4, 0($s2)
    addi $s2, $s2, 4
    subi $s0, $s0, 4
    addi $t2, $t2, 1
    beq $t2, 5, Exit1
    j Loop2

Exit1:
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg2 # load the address of msgsyscall
    syscall
    addi $t2, $zero, 0

Loop3:
    lw $t5, 0($s1)
    li $v0, 1
    move $a0, $t5
    syscall
    li $v0, 4 # print_string syscall code = 4
    la $a0, msg3 # load the address of msgsyscall
    syscall
    addi $s1, $s1, 4
    addi $t2, $t2, 1
    beq $t2, 10, End
    j Loop3

End :

```

.data

```

msg: .asciiz "a"
array: .word 0:10
msg1: .asciiz "Enter 10 array's elements : "
msg2: .asciiz "Array after reversed : "
msg3: .asciiz " | "

```