CS 224

Section No. 4

Spring 2020

Lab No. 1

Mannan Abdul

21801066

Part 1.

1. beq $t0, $t1, next => 0x11090003
2. bne $t0, $t1, again => 0x1509fffa
3. j again => 0x08100060
4. la $t0, array => 1. lui $at, 0x00001001 => 0x3c011001
   2. ori $t0, $at, 0x00000088 => 0x34280088
5. lw $t1, array => 1. lui $at, 0x00001001 => 0x3c011001
   2. lw $t1, 0x00000088($at) => 0x8c290088
6. bge $t1, $t2, next => 1. slt $at, $t1, $t2 => 0x012a082a
   2. beq $at, $zero, next => 0x1020fff9

Part 2 & 3.

```
#Lab03, Mannan Abdul, Part 2 & 3
        .text
        li $v0, 4
        la $a0, msg
        syscall

        li $v0, 5
        syscall

        move $t0, $v0

        li $v0, 4
```

```
la $a0, msg2
syscall

li $v0, 5
syscall

move $t1, $v0

move $a0, $t0
move $a1, $t1

addi $v0, $zero, 0
jal recursiveMultiplication

move $t0, $v0

li $v0, 4
la $a0, msg3
syscall

li $v0, 1
add $a0, $zero, $t0
syscall
```

```
li $v0, 4
la $a0, msg4
syscall

li $v0, 5
syscall

move $a0, $v0
addi $a1, $zero, 0

addi $v0, $zero, 0
jal recursiveSummation

move $t1, $v0

li $v0, 4
la $a0, msg5
syscall

li $v0, 1
add $a0, $zero, $t1
syscall
```

```
        li $v0, 10
        syscall

recursiveMultiplication:
        add $v0, $v0, $a0
        blt $a1, 2, done
        addi $a1, $a1, -1
        j recursiveMultiplication
done:
        jr $ra

recursiveSummation:
        bgt $a1, $a0, done
        add $v0, $v0, $a1
        addi $a1, $a1, 1
        j recursiveSummation


        .data
msg: .asciiz "\nEnter a positive integer: "
msg2:.asciiz "\nEnter a positive integer you want to multiply the first integer with: "
msg3:.asciiz "\nThe product of the 2 integers is: "
msg4:.asciiz "\n\nEnter a positive integer n: "
```

```
msg5:.asciiz "\nThe sum of integers from 1 to n is: "
```

Part 4.

```
Delete_x:
        li $t4, -1
        move $t0, $a0
        lw $t1, 4($t0)
        lw $t2, ($t0)
        beq $t1, $a1, deleteHead
next:
        move $t3, $t0
        move $t0, $t2
        lw $t2, ($t0)
        lw $t1, 4($t0)
        beq $t1, $a1, deleteNode
        bnez $t2, next
        j complete
deleteHead:
        move $a0, $t2
        li $t4, 0
        beqz $a0, complete
        j next
deleteNode:
```

```
        sw $t2, ($t3)
        li $t4, 0
        beqz $t2, complete
        j next

complete:
        move $v0, $t4
        move $v1, $a0
        jr $ra
        nop
```