Lab 5 Question 1

```
.data
buffer: .space 20
isPalin: .asciiz "\nIt's a Palindrome"
notPalin: .asciiz "\nNot a Palindrome"
newline: .asciiz "\n"
.text
main:
lb $t4, newline
la $a0, buffer     #a0 is buffer and a1 is length
li $a1, 20
li $v0, 8
syscall
add $t2, $a0, $zero        #initially base address in t2
slen:
        lb $t3, ($t2)
        addi $t2, $t2, 1
        beq $t3, $t4, next          #if current byte is \n
        bne $t3, $zero, slen #if current byte is \0
next:
        add $t1, $a0, $zero        #base address in t1
        addi $t2, $t2, -2 #last char, -2 because of \n and \0
testLoop:
        bge $t1, $t2, palin # if start>=end
        lb $t3, 0($t1)
        lb $t4, 0($t2)
        bne $t3, $t4, not_palin
        addi $t1, $t1, 1
        addi $t2, $t2, -1
        j testLoop
palin:
```

```
        li $v0, 4
        la $a0, isPalin
        syscall
        j exit
not_palin:
        li $v0, 4
        la $a0, notPalin
        syscall

exit:
        li $v0, 10
        syscall
```

Lab 5 Question 2

```
.data
str: .asciiz "hello"
char: .byte 'o'
ele: .asciiz "element found"
eleno: .asciiz "element not found"
.text
main:

la $t0, str
la $t1, char
addi $t2, $t0, 4

loop:
beq $t2, $t0, not_found
lb $t4, 0($t2)
lb $t5, 0($t1)
beq $t4, $t5, found
```

```
addi $t2, $t2, -1
j loop

found:
        la $a0, ele
        li $v0, 4
        syscall
        j exit

not_found:
        la $a0, eleno
        li $v0, 4
        syscall

exit:
li $v0, 10
syscall
```

Lab 5 Question 3

```
.data
strng: .space 6
rvstr: .space 6
newline: .asciiz "\n"
.text
main:

la $a0, strng
li $a1, 6
li $v0, 8
syscall
```

```
li $t0, 5
addi $t1, $a0, 4
la $t2, rvstr

loop:
        beq $t0, $zero, print
        lb $t3, 0($t1)
        sb $t3, 0($t2)
        addi $t1, $t1, -1
        addi $t2, $t2, 1
        addi $t0, $t0, -1
        j loop

print:
        li $t4, 0
        sb $t4, 0($t2)
        la $a0, strng
        li $v0, 4
        syscall

        la $a0, newline
        li $v0, 4
        syscall

        la $a0, rvstr
        li $v0, 4
        syscall

exit:
li $v0, 10
```

syscall

Lab 5 Question 4

```
.data
arr: .word 3 4 1 6 7 2
ma: .asciiz "The maximum element is: "
mi: .asciiz "The minimum element is: "
new: .asciiz "\n"
.text
main:

la $a0, arr
lw $t0, 0($a0)
lw $t1, 0($a0)
addi $a0, $a0, 4

move $t4, $zero
loop:
        beq $t4, 5, exit
        lw $t2, 0($a0)
        bge $t2, $t0, max
        ble $t2, $t1, min
        point:
                addi $t4, $t4, 1
                addi $a0, $a0, 4
                j loop

max:
        lw $t0, 0($a0)
```

```
        j point
min:
        lw $t1, 0($a0)
        j point

exit:
la $a0, ma
li $v0, 4
syscall

move $a0, $t0
li $v0, 1
syscall

la $a0, new
li $v0, 4
syscall

la $a0, mi
li $v0, 4
syscall

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

li $v0, 10
syscall
```

Lab 6 Question 2

```
.data
userInput1: .space 20
userInput2: .space 20
removed: .asciiz "Letters removed: "
size: .word 24
vowels: .asciiz "aeiou"
endline: .asciiz "\n"
text: .asciiz "Enter String to remove Vowels: "
.text
main:
la $a0, text
li $v0, 4
syscall
li $v0, 8
la $a0, userInput1
li $a1, 20
syscall

la $a0, userInput1
jal remove_vowels

move $a1, $v0
li $v0, 4
la $a0, removed
syscall

move $a0, $a1
li $v0, 1
syscall
```

```
li $v0, 4
la $a0, endline
syscall
la $a0, userInput1
syscall
li $v0, 10
syscall

remove_vowels:
li $v0, 0
la $a0, userInput1
check_letter:
        lb $a1, ($a0)
        beq $a1, $zero, exit
        la $t1, vowels
        check_vowels:
                lb $t2, ($t1)
                beq $t2, $zero, end_vowel_loop
                beq $t2, $a1, remove_letter
                addi $t1, $t1, 1
                j check_vowels
                remove_letter:
                        addi $v0, $v0, 1
                        move $t3, $a0
                        move $t4, $a0
                        addi $t4, $t4, 1
                        remove_loop:
                                lb $t5, ($t4)
                                lb $t6, ($t3)
                                beq $t6, $zero, end_remove_loop
```

```
                sb $t5, ($t3)
                addi $t4, $t4, 1
                addi $t3, $t3, 1
                j remove_loop
        end_remove_loop:
                addi $a0, $a0, -1
    end_vowel_loop:
        addi $a0, $a0, 1
    j check_letter
exit:
    jr $ra
```

Lab 6 Question 3 (Triangle) – using recursion

```
.data
print_str: .asciiz "Enter a number: "
line: .asciiz "\n"
star: .asciiz "*"
.text

main:
    la $a0, print_str
    li $v0,4
    syscall

    li $v0, 5
    syscall
    move $t2, $v0
    li $t1, 0
    jal print_star_line
```

```
li $v0, 10
syscall

print_star_line:
        addi $sp, $sp, -8
        sw $ra, 4($sp)
        sw $a0, 0($sp)
        li $t0, 0
        beq $t2, $zero, L1
        loop:
                bgt $t0, $t1, out
                la $a0, star
                li $v0, 4
                syscall
                addi $t0, $t0, 1
                j loop
        out:
                la $a0, line
                li $v0, 4
                syscall
        addi $t2, $t2, -1
        addi $t1, $t1, 1
        jal print_star_line
        L1:
                lw $ra, 4($sp)
                addi $sp, $sp, 8
                jr $ra
```

Lab 6 Question 3 using loops

```
.data
star: .asciiz "*"
newline: .asciiz "\n"
buffer: .space 20
text: .asciiz "Enter square/triangle followed by integer: "
.text
main:

la $a0, text
li $v0, 4
syscall

la $a0, buffer
li $a1, 20
li $v0, 8
syscall

li $v0, 5
syscall
move $a1, $v0

lb $t1, 0($a0)
beq $t1, 0x74, triangle   #74 is hex of ascii of t.. so if first char is t then triangle
squaure:
        jal print_square
        j exit

triangle:
        jal print_triangle
```

```
exit:
        li $v0, 10
        syscall

print_square:
        move $t0, $a1
        loop1:
                move $t1, $a1
                loop2:
                        la $a0, star
                        li $v0, 4
                        syscall
                        addi $t1, $t1, -1
                        bne $t1, $zero, loop2

                la $a0, newline
                li $v0, 4
                syscall
                addi $t0, $t0, -1
                bne $t0, $zero, loop1
                jr $ra

print_triangle:
        move $t3, $a1
        addi $t3, $t3, 1
        li $t0, 1
        loop3:
                move $t1, $t0
                loop4:
                        la $a0, star
                        li $v0, 4
```

```
        syscall
        addi $t1, $t1, -1
        bne $t1, $zero, loop4

    la $a0, newline
    li $v0, 4
    syscall
    addi $t0, $t0, 1
    bne $t0, $t3, loop3
    jr $ra
```

Lab 6 Question 4

```
.data
prompt: .asciiz "input an integer x: "
result: .asciiz "\nFact(x) = "
.text
main:

la $a0, prompt
li $v0, 4
syscall

li $v0, 5
syscall
move $a0, $v0
jal factorial
move $t0, $v0

li $v0, 4
la $a0, result
```

```
syscall

li $v0, 1
move $a0, $t0
syscall

li $v0, 10
syscall

factorial:
        addi $sp, $sp, -8
        sw $s0, 4($sp)
        sw $ra, 0($sp)
        bne $a0, 0, else
        addi $v0, $zero, 1
        j fact_return

        else:
                move $s0, $a0
                addi $a0, $a0, -1
                jal factorial
                multu $s0, $v0
                mflo $v0
        fact_return:
                lw $s0, 4($sp)
                lw $ra, 0($sp)
                addi $sp, $sp, 8
                jr $ra
```

Lab 8 Question 1

```
.data
print_str: .asciiz "Enter a number: "
.text

main:
        la $a0, print_str
        li $v0, 4
        syscall

        li $v0, 5
        syscall

        move $t2, $v0

        sll $v0, $v0, 2
        move $a0, $v0
        li $v0, 9
        syscall

        move $t0, $v0
        li $t1, 0



        inp_loop:
                beq $t1, $t2, exit1
                li $v0, 5
                syscall
                move $t3, $t1
                sll $t3, $t3, 2
                add $t3, $t3, $t0
```

```
        sw $v0, 0($t3)
        addi $t1, $t1, 1
        j inp_loop

exit1:
        li $t4, 0
        li $t5, 0
        out_loop:
                beq $t4, $t2, exit2
                move $t6, $t4
                sll $t6, $t6, 2
                add $t6, $t6, $t0
                lw $t7, 0($t6)
                add $t5, $t5, $t7
                addi $t4, $t4, 1
                j out_loop

exit2:
        move $a0, $t5
        li $v0, 1
        syscall
        li $v0, 10
        syscall
```

Lab 8 Question 2

```
.data
nameinput: .asciiz "Enter name: "
rollinput: .asciiz "Enter rollno: "
cgpainput: .asciiz "Enter CGPA: "
newline: .asciiz "\n"
input: .asciiz "Enter number of students: "
```

```
.text
main:
la $a0, input
jal print

li $v0, 5
syscall

move $t0, $v0
sll $a0, $t0, 4          # 8 bytes for name, 4 for rollno, 4 for cgpa
li $v0, 9
syscall

move $s0, $v0            #Base address in s0

li $t1, 0
loop1:
        move $t2, $t1                   #t2 = i
        sll $t2, $t2, 4                 #t2 = 16*i
        move $s1, $s0                   #s1 = base
        add $s1, $s1, $t2               #For the i student goto base+16*i location

        la $a0, nameinput
        jal print

        add $a0, $s1, 0                 #take name input
        li $a1, 8
        li $v0, 8
        syscall

        la $a0, rollinput
```

```
        jal print

        add $a0, $s1, 8          #take roll input
        li $v0, 5
        syscall
        sw $v0, 0($a0)

        la $a0, cgpainput
        jal print

        add $a0, $s1, 12         #take cgpa input
        li $v0, 5
        syscall
        sw $v0, 0($a0)

        add $t1, $t1, 1
        bne $t1, $t0, loop1

li $t1, 0
loop2:
        move $t2, $t1            #t2 = i
        sll $t2, $t2, 4          #t2 = 16*i
        move $s1, $s0            #s1 = base
        add $s1, $s1, $t2        #For i student goto base+16*i location

        move $a0, $s1
        li $v0, 4
        syscall

        lw $a0, 8($s1)
        li $v0, 1
```

```
        syscall

        jal printnewline

        lw $a0, 12($s1)
        li $v0, 1
        syscall

        jal printnewline

        add $t1, $t1, 1
        bne $t1, $t0, loop2

li $v0, 10
syscall

print:
        li $v0, 4
        syscall
        jr $ra

printnewline:
        la $a0, newline
        li $v0, 4
        syscall
        jr $ra
```