**Lab2 Preliminary**

CS224

Section No: 1

Fall 2019

Lab No. 2

Fatih Sevban Uyanık

21602486

**Q1)**

.data

octal\_no: .asciiz "245"

new\_line: .asciiz "\n"

.text

main:

la $a1, octal\_no

jal convertToDec

move $s1, $v0

# printing an integer

addi $v0, $zero, 1

addi $a0, $s1, 0

syscall

# main

li $v0, 10

syscall

convertToDec:

addi $sp, $sp, -8

sw $ra, 0($sp)

sw $a1, 4($sp)

addi $t0, $a1, 0

addi $t1, $zero, 0

loop\_1: lbu $t2, 0($t0)

beq $t2, $zero, exit\_1

addi $t0, $t0, 1

addi $t1, $t1, 1

j loop\_1

exit\_1:

addi $t2, $zero, 0

addi $t4, $zero, 1

addi $t5, $zero, 0

loop\_2: beq $t1, $zero, exit\_2

addi $t1, $t1, -1

addi $t0, $t0, -1

lb $t3, 0($t0)

addi $t3, $t3, -48

beq $t5, $zero, jump\_1

mul $t4, $t4, 8

mul $t3, $t3, $t4

jump\_1:

add $t5, $t5, 1

add $t2, $t2, $t3

j loop\_2

exit\_2:

addi $v0, $t2, 0

lw $ra, 0($sp)

lw $a1, 4($sp)

jr $ra

**Q2)**

.data

octal\_no: .space 20

new\_line: .asciiz "\n"

wrong\_input\_message: .asciiz "Input is wrong."

string\_space: .space 1024

request\_octal: .asciiz "Please enter an octal number: "

output\_decimal: .asciiz "Decimal Number: "

.text

main:

jal interactWithUser

la $a1, octal\_no

jal convertToDec

move $s1, $v0

addi $v0, $zero, 4

la $a0, output\_decimal

syscall

# printing an integer

addi $v0, $zero, 1

addi $a0, $s1, 0

syscall

# main

li $v0, 10

syscall

convertToDec:

addi $sp, $sp, -8

sw $ra, 0($sp)

sw $a1, 4($sp)

addi $t0, $a1, 0

addi $t1, $zero, 0

loop\_1: lbu $t2, 0($t0)

beq $t2, $zero, exit\_1

addi $t0, $t0, 1

addi $t1, $t1, 1

j loop\_1

exit\_1:

addi $t2, $zero, 0

addi $t4, $zero, 1

addi $t5, $zero, 0

addi $t1, $t1, -1

addi $t0, $t0, -1

loop\_2: beq $t1, $zero, exit\_2

addi $t1, $t1, -1

addi $t0, $t0, -1

lb $t3, 0($t0)

addi $t3, $t3, -48

beq $t5, $zero, jump\_1

mul $t4, $t4, 8

mul $t3, $t3, $t4

jump\_1:

add $t5, $t5, 1

add $t2, $t2, $t3

j loop\_2

exit\_2:

addi $v0, $t2, 0

lw $ra, 0($sp)

lw $a1, 4($sp)

jr $ra

# convertToDec

interactWithUser:

request\_input:

addi $v0, $zero, 4

la $a0, request\_octal

syscall

li $v0, 8

la $a1, 20

la $a0, octal\_no

syscall

la $t1, octal\_no

loop\_3: lb $t3, 0($t1)

addi $t1, $t1, 1

addi $t2, $zero, 10

beq $t3, $t2, exit\_3

addi $t2, $zero, 57

slt $t4, $t3, $t2

addi $t2, $zero, 48

slt $t5, $t3, $t2

addi $t6, $zero, 1

bne $t4, $t6, exit\_4

bne $t5, $zero, exit\_4

j loop\_3

exit\_4:

addi $v0, $zero, 4

la $a0, wrong\_input\_message

syscall

addi $v0, $zero, 4

la $a0, new\_line

syscall

j request\_input

exit\_3:

addi $v0, $zero, 1

jr $ra

# interactWithUser

**Q3)**

**beq $t0, $t6, next**

Binary: 000100 01000 01110 0000000000000010 (I type)

Hex: 0x110E0002

**bne $t0, $t6, again**

Binary: 000101 01000 011101111111111111010 (I type)

Hex: 0x150EFFFA

**j again**

Binary: 000010 00000100000000000000101000 (J type)

Hex: 0x08100028

**la $t0, array2**

Address of array2 is --> 1001 0064

la $t0, array2 --> is equivalent of the following instructions:

lui $at, 0x1001

Binary: 001111 00000 00001 0001 0000 0000 0001 (I type)

Hex: 0x3C011001

ori $t0, $at, 0x0064

Binary: 001101 00001 01000 0000 0000 0110 0100 (I type)

Hex: 0x34280064

**lw $t1, array2**

Address of array2 is --> 1001 0064

lw $t1, array2 --> is equivalent of the following instructions:

lui $at, 0x1001

Binary: 001111 00000 00001 0001000000000001 (I type)

Hex: 0x3C011001

lw $t1, 0x0064($at)

Binary: 100011 00001 01001 0000000001100100 (I type)

Hex: 0x8C290064