

CS224

Section No.: 04

Fall 2019

Lab No. 1

Okan Şen/21202377

```

1)
# -----
#
# Start of convertToDec
#
# -----
la $a0, octalNo

move $t0, $a0
#-----
# Calculate length of string
sLength:
    lb $t1, 0($t0)
    beq $t1, $zero, endLength

    addi $t0, $t0, 1
    j sLength

endLength:

    move $t1, $a0
    sub $t3, $t0, $t1          #$t3 now contains the length of the string
#-----
jal convertToDec
# result comes in $v0
    move $a1, $a0

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

exit:
    li $v0, 10                # system call to exit
    syscall                   # bye bye
# stop execution here by syscall

convertToDec:
    # This section takes first digit of string and stores them in a stack.
    addi $sp, $sp, -28
    sw $s0, 24($sp)
    sw $s1, 20($sp)
    sw $s2, 16($sp)

```

```
sw $s3, 12($sp)
sw $s4, 8($sp)
sw $s5, 4($sp)
sw $s6, 0($sp)
```

```
move $t0, $a0          # address of string
li $t9, 0
```

traverse:

```
lbu $t1, ($t0)          #Get first digit of string

addi $t1,$t1,-48         #Convert from ASCII to digit
```

```
sw $t1, 0($sp)
addi $sp, $sp, 4
```

```
addi $t0, $t0, 1        #Advance string pointer
addi $t9, $t9, 1         # increase loop counter
blt $t9, $t3, traverse
```

```
li $t8, 0               # keeps track of power
li $t5, 0               # keep total of sum in t5
addi $sp, $sp, -4       # sp was increased by 4 bytes one last time unnecessarily so we
have to decrease the address
# Continue with the calculations
```

decimalC:

```
lw $t1, 0($sp)
```

```
j power
cont:
addi $sp, $sp, -4
addi $t8, $t8, 1
blt $t8, $t3, decimalC
j endConvert
```

power:

```
move $t6, $t1
li $t4, 0
bnez $t8, notzero
```

```

        add $t5, $t5, $t1          # for last digit add the digit value itself
        j cont

```

notzero:

```

        mul $t6, $t6, 8
        addi $t4, $t4, 1

        blt $t4, $t8, notzero
        add $t5, $t5, $t6
        j cont

```

endConvert:

```

        lw $s0, 24($sp) # restore $s0 value from stack
        lw $s1, 20($sp) # restore $s1 value from stack
        lw $s2, 16($sp) # restore $s2 value from stack
        lw $s3, 12($sp) # restore $s3 value from stack
        lw $s4, 8($sp)  # restore $s4 value from stack
        lw $s5, 4($sp)  # restore $s5 value from stack
        lw $s6, 0($sp)  # restore $s6 value from stack
        addi $sp, $sp, 28 # restore $sp to original value (i.e. pop 7 items)

        move $v0, $t5

```

```

        jr $ra

```

.data

```

octalNo:      .asciiz "20"

```

```

# -----
#
# End of convertToDec
#
# -----

```

2)

# -----

#

# Start of interactWithUser

#

# -----

start:

la \$a0, msg1

li \$v0, 4

syscall

li \$v0, 8

# take in input

la \$a0, word # load byte space into address

li \$a1, 28 # allot the byte space for string

move \$t0, \$a0 # save string to t0

syscall

#-----

# Calculate length of string

sLength:

lb \$t1, 0(\$t0)

beq \$t1, \$zero, endLength

addi \$t0, \$t0, 1

j sLength

endLength:

move \$t1, \$a0

sub \$t3, \$t0, \$t1

#\$t3 now contains the length of the string

#-----

#check before continuing

#-----

#length first

li \$t7, 8

bgt \$t3, \$t7, lengthL # if length is greater than 7 jump to this branch

j octalcheck #if length is okay check if it is in octal form

lengthL:

la \$a0, lengtherr

# print necessary message and jump back to start

```

        li $v0, 4
        syscall

        j start
#-----
# now the octal form
#-----
octalcheck:
    # checks all digits by converting them into integers then comparing it to 7.
    move $t0, $a0          # address of string
    li $t9, 1
    li $t7, 7

    iterate:

        lbu $t1, ($t0)      #Get first digit of string

        addi $t1,$t1,-48     #Convert from ASCII to digit

        bgt $t1, $t7, octalL # if greater than 7 jump to error message and back to start else
continue iterating

        addi $t0, $t0, 1    #Advance string pointer
        addi $t9, $t9, 1    # increase loop counter
        blt $t9, $t3, iterate

        # if all conditions are met, now we can get to the rest of the program
        j rest
#-----
    octalL:
        la $a0, octalerr    # print necessary message and jump back to start
        li $v0, 4
        syscall

        j start
#-----
#-----
rest:

jal convertToDec
# result comes in $v0
    move $a1, $a0

```

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

exit:

```
li $v0, 10          # system call to exit
syscall             # bye bye
# stop execution here by syscall
```

convertToDec:

```
# This section takes first digit of string and stores them in a stack.
addi $sp, $sp, -28
sw $s0, 24($sp)
sw $s1, 20($sp)
sw $s2, 16($sp)
sw $s3, 12($sp)
sw $s4, 8($sp)
sw $s5, 4($sp)
sw $s6, 0($sp)
```

```
move $t0, $a0          # address of string
li $t9, 1
```

traverse:

```
lbu $t1, ($t0)         #Get first digit of string

addi $t1,$t1,-48        #Convert from ASCII to digit

sw $t1, 0($sp)
addi $sp, $sp, 4

addi $t0, $t0, 1        #Advance string pointer
addi $t9, $t9, 1        # increase loop counter
blt $t9, $t3, traverse

li $t8, 0               # keeps track of power
li $t5, 0               # keep total of sum in t5
```

```
        addi $sp, $sp, -4      # sp was increased by 4 bytes one last time unnecessarily so we
have to decrease the address
```

```
        # Continue with the calculations
decimalC:
```

```
        lw $t1, 0($sp)
```

```
        j power
cont:
```

```
        addi $sp, $sp, -4
        addi $t8, $t8, 1
        blt $t8, $t3, decimalC
        j endConvert
```

```
power:
```

```
        move $t6, $t1
        li $t4, 0
        bnez $t8, notzero
```

```
        add $t5, $t5, $t1      # for last digit add the digit value itself
        j cont
```

```
notzero:
```

```
        mul $t6, $t6, 8
        addi $t4, $t4, 1

        blt $t4, $t8, notzero
        add $t5, $t5, $t6
        j cont
```

```
endConvert:
```

```
        lw $s0, 24($sp) # restore $s0 value from stack
        lw $s1, 20($sp) # restore $s1 value from stack
        lw $s2, 16($sp) # restore $s2 value from stack
        lw $s3, 12($sp) # restore $s3 value from stack
        lw $s4, 8($sp)  # restore $s4 value from stack
        lw $s5, 4($sp)  # restore $s5 value from stack
        lw $s6, 0($sp)  # restore $s6 value from stack
        addi $sp, $sp, 28 # restore $sp to original value (i.e. pop 7 items)

        move $v0, $t5
```



jr \$ra

.data

msg1: .ascii "Enter an octal number(max 7 digits): "  
endl: .ascii "\n"  
word: .space 28  
lengtherr: .ascii "Do not enter more than 7 digits!\n"  
octalerr: .ascii "Enter in octal form (all digits lower than 8)!\n"

# -----  
#  
# End of interactWithUser  
#  
# -----

3)

Bne = 11 0E 00 40 C0