**Preliminary Work / Preliminary Design Report  
 (30 points)**

CS224

Section No.: 05

Spring 2018

Lab No.: 02

Asuman Aydın/21502604

**1**. **(20 points)** Write MIPS assembly language programs as described below.

**a**. **(10 points)** **convertHexToDec**: Write a subprogram, called convertHexToDec, that receives the beginning address of an asciiz string, in register $a0, that contains a hexadecimal number in the form of a string, for example, like "1A", and returns its decimal (1A16= 2610) equivalent in register $v0. If the number stored in the input string is not a proper hexadecimal number it returns -1 in $v0. A sketch of this convertHexToDec is as follows.

.text

la $a0, hexNo

move $t0, $a0

jal convertHexToDec

# result comes in $v0

move $v0, $s2

li $v0, 1

syscall

j done

convertHexToDec:

lb $t1, 0($t0)

beqz $t1, end2

bgt $t1, 48, end2

add $s2, $s2, $t1

lb $t2, 4($t0)

beqz $t2, end2

bgt $t2, 48, end2

add $s2, $s2, $t2

jr $ra

end2:

addi $s1, $s1, -1

li $v0, 1

add $a0, $s1, $zero

syscall

done:

li $v0, 10 #ends program

syscall

.data

hexNo: .asciiz "2B"

**b**. **(10 points)** **interactWithUser**: Write a subprogram, called interactWithUser, that asks the user enter an hexadecimal number in the form of a string, makes sure that it is a proper hexadecimal number if not generates an error message and ensures that a proper input is received. It passes this address to the subprogram defined above convertHexToDec, if necessary modify it, and gets the result from it and returns the decimal value back to the caller, i.e. the main program. The main program is the one that comes right after the assembler directive .text. How to read a string: See MIPS system calls on the web to understand how to read a string. Invoke interactWithUser from the main program.

Use the $s registers during the implementation of the above subprograms. Using $s registers means that you have to save them to stack and restore them back from stack. (Question: Why do we use the $s registers? Answer: To have practice about the use of stack in MIPS.) Make sure that you also save/restore any other register that need to be saved.

.text

la $a0, hexNo

move $t0, $a0

jal interactWithUser

jal convertHexToDec

# result comes in $v0

move $v0, $s2

li $v0, 1

syscall

j done

interactWithUser:

move $v0, askUser

li $v0,1

syscall

li $v0, 5

syscall

move $t3,$v0

convertHexToDec:

lb $t1, 0($t3)

beqz $t1, end2

bgt $t1, 48, end2

add $s2, $s2, $t1

lb $t2, 4($t0)

beqz $t2, end2

bgt $t2, 48, end2

add $s2, $s2, $t2

jr $ra

end2:

addi $s1, $s1, -1

li $v0, 1

add $a0, $s1, $zero

syscall

done:

li $v0, 10 #ends program

syscall

.data

hexNo: .asciiz "2B"

askUser: .asciiz " enter an hexadecimal number in the form of a string"

**2**.  **(10 points)** Generating machine instructions   
(**jump instructions 10 points**) Give the object code in hexadecimal for the following jump (j) instructions.

...

previous: nop

nop

nop

nop

**j next**

nop

add $t0 ,$t1, $t2

next: **j previous**

Assume that the label previous is located at memory location 10 01 00 1C16. If you think that you do not have enough information to generate the code explain. See the textbook and Chap 6 slides of the textbook for the MIPS memory map (available at our unilica web site, Documents folder).