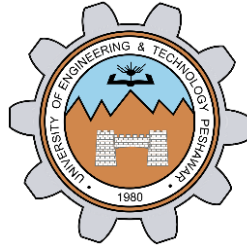

COA Lab

LAB # 03



Fall 2020

CSE304L Computer Organization and Architecture Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as a student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: _____

Submitted to:

Engr. Amaad Khalil

December 8, 2020

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Task1:

Write mips assembly code for a program that takes four input numbers from the user and identify and display the greater number.

Source code:

```
.data
    str: .asciiz"Enter the first number: "
    str1: .asciiz"Enter the second number: "
    str2: .asciiz"Enter the third number: "
    str3: .asciiz"Enter the fourth number: "
    str4: .asciiz"The Greatest number is: "

.text
main:
    li $v0,4      #system call code for printing string
    la $a0,str     #address of the string to print
    syscall       #print the string
    li $v0,5      #input first number
    syscall       #system call
    move $t0,$v0  #move first number to t0

    li $v0,4      #system call code for printing string
    la $a0,str1    #address of the string to print
    syscall       #print the string
    li $v0,5      #input second number
    syscall       #system call
    move $t1,$v0  #move second number to t1

    li $v0,4      #system call code for printing string
    la $a0,str2    #address of the string to print
    syscall       #print the string
    li $v0,5      #input third number
```

```

syscall      #system call
move $t2,$v0 #move third number to t1

li $v0,4     #system call code for printing string
la $a0,str3  #address of the string to print
syscall      #print the string
li $v0,5     #input fourth number
syscall      #system call
move $t3,$v0 #move fourth number to t1

li $v0,4     #system call code for printing string
la $a0,str4  #address of the string to print
syscall      #print the string

bgt $t0,$t1, t0GTt1  #check if t0 is greater than t1
move $t4,$t1        #move t1 to t4
j test2             #jump to test2
t0GTt1:
move $t4,$t0        #move t0 to t4
test2:
bgt $t2,$t4, t2GTt4  #check if t2 is greater than t4
j test3             #jump to test3

t2GTt4:
move $t4,$t2        #move t2 to t4
test3:
bgt $t3,$t4, t3GTt4  #check if t3 is greater than t4
j exit              #jump to exit
t3GTt4:
move $t4,$t3        #move t3 to t4
exit:
li $v0,1            #system call code to print int
move $a0,$t4        #move t4 to a0

```

```
syscall      #print value in a0
li $v0,10    #Terminate the program
syscall      #syscall
```

Output:

```
PC          = 4194532
EPC         = 0
Cause       = 0
BadVAddr    = 0
Status      = 805371664

HI          = 0
LO          = 0

R0 [r0]     = 0
R1 [at]     = 0
R2 [v0]     = 10
R3 [v1]     = 0
R4 [a0]     = 7
R5 [a1]     = 2147480968
R6 [a2]     = 2147480988
R7 [a3]     = 0
R8 [t0]     = 6
R9 [t1]     = 7
R10 [t2]    = 3
R11 [t3]    = 2
R12 [t4]    = 7
R13 [t5]    = 0
R14 [t6]    = 0
R15 [t7]    = 0
R16 [s0]    = 0
R17 [s1]    = 0
R18 [s2]    = 0
```

Console

```
Enter the first number: 6
Enter the second number: 7
Enter the third number: 3
Enter the fourth number: 2
The Greatest number is: 7
```

Task2:

Write mips assembly code for a program that takes two input numbers and display the results using the given instructions below. Here Lab1, Lab2, Lab3 and Lab4 are Labels.

```
blt $t0, $t1, Lab1 # Branch if $t0 < $t1
ble $t0, $t1, Lab2 # Branch if $t0 <= $t1
bgt $t0, $t1, Lab3 # Branch if $t0 > $t1
bge $t0, $t1, Lab4 # Branch if $t0 >= $t1
```

Source code:

```
.data
    str: .asciiz"Enter the first number: "
    str1: .asciiz"Enter the second number: "
    str2: .asciiz"t0 is less than t1\n"
    str3: .asciiz"t0 is less than or equal to t1\n"
    str4: .asciiz"t0 is greater than t1\n"
    str5: .asciiz"t0 is greater than or equal to t1\n"

.text

main:
    li $v0,4      #system call code for printing string
    la $a0,str     #address of the string to print
    syscall       #print the string
    li $v0,5      #input first number
    syscall       #system call
    move $t0,$v0   #move first number to t0

    li $v0,4      #system call code for printing string
    la $a0,str1    #address of the string to print
    syscall       #print the string
    li $v0,5      #input second number
    syscall       #system call
    move $t1,$v0   #move second number to t1
```

```

        blt $t0,$t1,Lab1      #check if t0 is less than t1
        j test2              #jump to test2

Lab1:
        li $v0,4              #system call code for printing string
        la $a0,str2           #address of the string to print
        syscall               #print the string

test2:
        ble $t0,$t1,Lab2      #check if t0 is less than or equal to t1
        j test3              #jump to test3

Lab2:
        li $v0,4              #system call code for printing string
        la $a0,str3           #address of the string to print
        syscall               #print the string

test3:
        bgt $t0,$t1,Lab3      #check if t0 is greater than t1
        j test4              #jump test4

Lab3:
        li $v0,4              #system call code for printing string
        la $a0,str4           #address of the string to print
        syscall               #print the string

test4:
        bge $t0,$t1,Lab4      #check if t0 is greater than or equal to t1
        j exit                #jump to exit

Lab4:
        li $v0,4              #system call code for printing string
        la $a0,str5           #address of the string to print
        syscall               #print the string

exit:
        li $v0,10             #Terminate the program
        syscall               #syscall

```

Output:

```
PC      = 4194508
EPC     = 0
Cause   = 0
BadVAddr = 0
Status  = 805371664

HI      = 0
LO      = 0

R0  [r0] = 0
R1  [at] = 268500992
R2  [v0] = 10
R3  [v1] = 0
R4  [a0] = 268501118
R5  [a1] = 2147480968
R6  [a2] = 2147480988
R7  [a3] = 0
R8  [t0] = 4
R9  [t1] = 3
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
```

 Console

```
Enter the first number: 4
Enter the second number: 3
t0 is greater than t1
t0 is greater than or equal to t1
|
```

Task3:

Write a mips assembly code for a program using .byte instruction that displays your name, father name, college, school, village, city, province and country name in new line.

Source code:

```
.data
msg: .byte 0x4E      #Hex code for ASCII N
      .byte 0x61      #Hex code for ASCII a
      .byte 0x6D      #Hex code for ASCII m
      .byte 0x65      #Hex code for ASCII e

      .byte 0x3A      #Hex code for ASCII :

      .byte 0x53      #Hex code for ASCII S
      .byte 0x68      #Hex code for ASCII h
      .byte 0x61      #Hex code for ASCII a
      .byte 0x68      #Hex code for ASCII h
      .byte 0x20      #Hex code for ASCII space
      .byte 0x52      #Hex code for ASCII R
      .byte 0x61      #Hex code for ASCII a
      .byte 0x7A      #Hex code for ASCII z
      .byte 0x61      #Hex code for ASCII a

      .byte 0xA       #Hex code for ASCII newline

      .byte 0x46      #Hex code for ASCII F
      .byte 0x61      #Hex code for ASCII a
      .byte 0x74      #Hex code for ASCII t
      .byte 0x68      #Hex code for ASCII h
      .byte 0x65      #Hex code for ASCII e
      .byte 0x72      #Hex code for ASCII r
      .byte 0x20      #Hex code for ASCII space
      .byte 0x4E      #Hex code for ASCII N
      .byte 0x61      #Hex code for ASCII a
```

.byte 0x6D	#Hex code for ASCII m
.byte 0x65	#Hex code for ASCII e
.byte 0x3A	#Hex code for ASCII :
.byte 0x47	#Hex code for ASCII G
.byte 0x75	#Hex code for ASCII u
.byte 0x6C	#Hex code for ASCII l
.byte 0x73	#Hex code for ASCII s
.byte 0x68	#Hex code for ASCII h
.byte 0x61	#Hex code for ASCII a
.byte 0x64	#Hex code for ASCII d
.byte 0xA	#Hex code for ASCII newline
.byte 0x43	#Hex code for ASCII C
.byte 0x6F	#Hex code for ASCII o
.byte 0x6C	#Hex code for ASCII l
.byte 0x6C	#Hex code for ASCII l
.byte 0x65	#Hex code for ASCII e
.byte 0x67	#Hex code for ASCII g
.byte 0x65	#Hex code for ASCII e
.byte 0x3A	#Hex code for ASCII :
.byte 0x53	#Hex code for ASCII S
.byte 0x63	#Hex code for ASCII c
.byte 0x68	#Hex code for ASCII h
.byte 0x6F	#Hex code for ASCII o
.byte 0x6C	#Hex code for ASCII l
.byte 0x61	#Hex code for ASCII a
.byte 0x72	#Hex code for ASCII r
.byte 0x73	#Hex code for ASCII s

.byte 0xA #Hex code for ASCII newline

.byte 0x53 #Hex code for ASCII S

.byte 0x63 #Hex code for ASCII c

.byte 0x68 #Hex code for ASCII h

.byte 0x6F #Hex code for ASCII o

.byte 0x6F #Hex code for ASCII o

.byte 0x6C #Hex code for ASCII l

.byte 0x3A #Hex code for ASCII :

.byte 0x53 #Hex code for ASCII S

.byte 0x63 #Hex code for ASCII c

.byte 0x68 #Hex code for ASCII h

.byte 0x6F #Hex code for ASCII o

.byte 0x6C #Hex code for ASCII l

.byte 0x61 #Hex code for ASCII a

.byte 0x72 #Hex code for ASCII r

.byte 0x73 #Hex code for ASCII s

.byte 0xA #Hex code for ASCII newline

.byte 0x56 #Hex code for ASCII V

.byte 0x69 #Hex code for ASCII i

.byte 0x6C #Hex code for ASCII l

.byte 0x6C #Hex code for ASCII l

.byte 0x61 #Hex code for ASCII a

.byte 0x67 #Hex code for ASCII g

.byte 0x65 #Hex code for ASCII e

.byte 0x3A #Hex code for ASCII :

.byte 0x4D	#Hex code for ASCII M
.byte 0x61	#Hex code for ASCII a
.byte 0x72	#Hex code for ASCII r
.byte 0x61	#Hex code for ASCII a
.byte 0x69	#Hex code for ASCII i
.byte 0xA	#Hex code for ASCII newline
.byte 0x43	#Hex code for ASCII C
.byte 0x69	#Hex code for ASCII i
.byte 0x74	#Hex code for ASCII t
.byte 0x79	#Hex code for ASCII y
.byte 0x3A	#Hex code for ASCII :
.byte 0x4B	#Hex code for ASCII K
.byte 0x6F	#Hex code for ASCII o
.byte 0x68	#Hex code for ASCII h
.byte 0x61	#Hex code for ASCII a
.byte 0x74	#Hex code for ASCII t
.byte 0xA	#Hex code for ASCII newline
.byte 0x50	#Hex code for ASCII P
.byte 0x72	#Hex code for ASCII r
.byte 0x6F	#Hex code for ASCII o
.byte 0x76	#Hex code for ASCII v
.byte 0x69	#Hex code for ASCII i
.byte 0x6E	#Hex code for ASCII n
.byte 0x63	#Hex code for ASCII c
.byte 0x65	#Hex code for ASCII e
.byte 0x3A	#Hex code for ASCII :

```

        .byte 0x4B      #Hex code for ASCII K
        .byte 0x50      #Hex code for ASCII P
        .byte 0x4B      #Hex code for ASCII K

        .byte 0xA       #Hex code for ASCII newline

        .byte 0x43      #Hex code for ASCII C
        .byte 0x6F      #Hex code for ASCII o
        .byte 0x75      #Hex code for ASCII u
        .byte 0x6E      #Hex code for ASCII n
        .byte 0x74      #Hex code for ASCII t
        .byte 0x72      #Hex code for ASCII r
        .byte 0x79      #Hex code for ASCII y

        .byte 0x3A      #Hex code for ASCII :

        .byte 0x50      #Hex code for ASCII P
        .byte 0x61      #Hex code for ASCII a
        .byte 0x6B      #Hex code for ASCII k
        .byte 0x69      #Hex code for ASCII i
        .byte 0x73      #Hex code for ASCII s
        .byte 0x74      #Hex code for ASCII t
        .byte 0x61      #Hex code for ASCII a
        .byte 0x6E      #Hex code for ASCII n
        .byte 0x0       #Hex code for ASCII NUL

.text
main:
    li $v0,4            #system call to print string
    la $a0,msg          #address of the string
    syscall             #print
    li $v0,10           #Terminate the program
    syscall             #syscall

```

Output:

A screenshot of a console window titled "Console". The window contains the following text output:

```
Name:Shah Raza  
Father Name:Gulshad  
College:Scholars  
School:Scholars  
Village:Marai  
City:Kohat  
Province:KPK  
Country:Pakistan
```

The console window has a standard Windows-style title bar and a vertical scrollbar on the right side.