Amrita Vishwa Vidyapeetham

Amrita School of Engineering, Coimbatore

Department of Computer Science and Engineering

Topic: MIPS ALP using QtSPIM

Sub Code: 19CS211 **Sub Title: COA**

Roll No: CB.EN.U4CSE19453 **Name: R.ABHINAV**

Lab Evaluation No: MIPS **Date: 03-05-2021**

---

1. **Write a MIPS ALP to find the minimum and maximum elements in an array and also find their position (5 Marks).Size and elements in the array can be decided by you.**

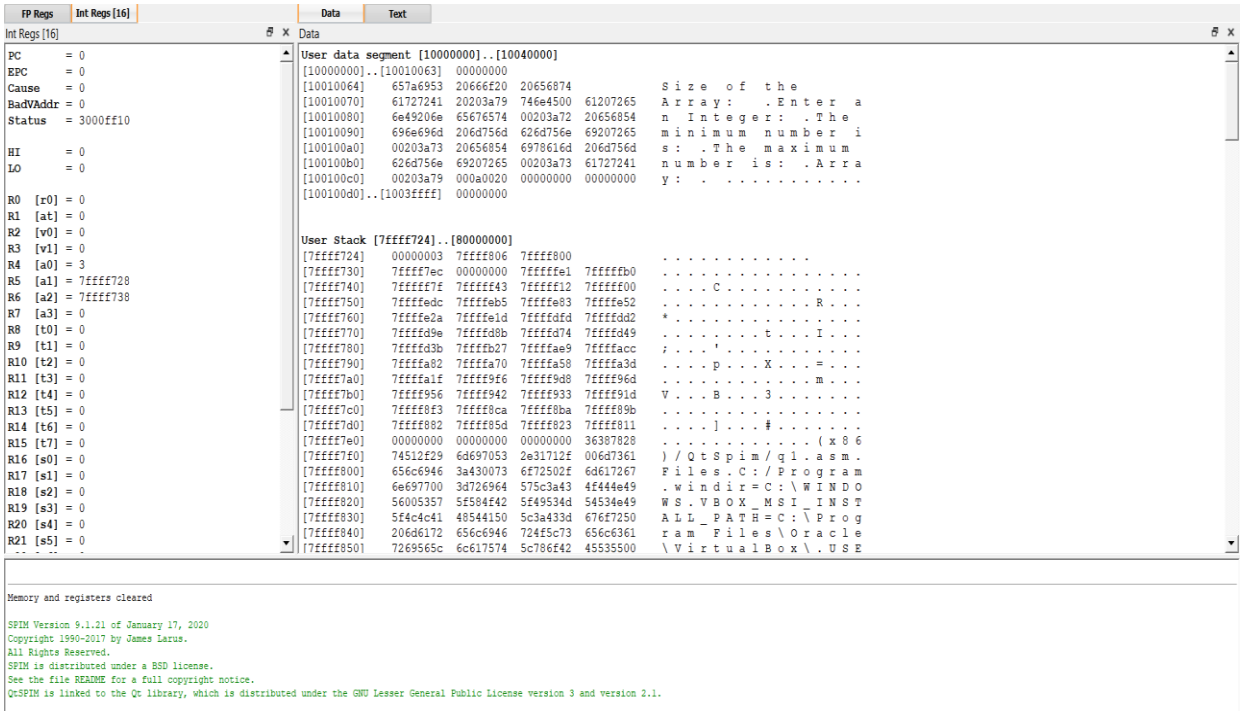
   Code :

```
.text
    .globl main

main:
    la $a0, ask              #ask user for integer thats going
to be the size of the array
    li $v0, 4
    syscall

    li $v0, 5                #store that int
    syscall
    move $t1,$v0             #size of my array stored in $t1

    la $t0, array            #load address of our array
    li $t2, 0                #counter = 0
    lw $t3,($t0)             #initialize min = array[0]
    lw $t4,($t0)             #initialize max = array[0]


while:
    la $a0, intask           #ask user for integer
    li $v0, 4
    syscall

    li $v0, 5                #store that int
    syscall
    sw $v0, ($t0)            #store that int in the array
```

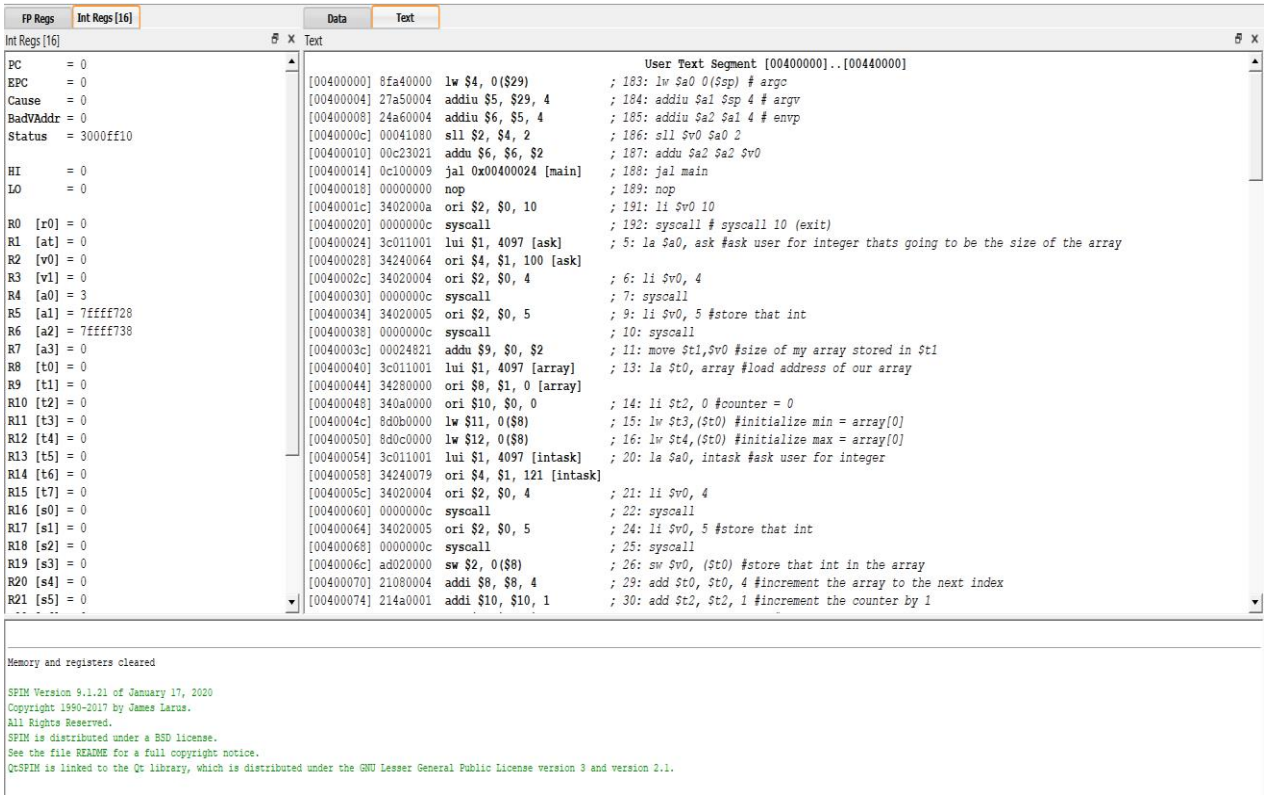
```
end:    add $t0, $t0, 4          #increment the array to the
next index
    add $t2, $t2, 1          #increment the counter by 1
    blt $t2, $t1,while       #branch to while if counter < size
of array

endw:
    la $a0,display           # Display "Array is: "
    li $v0,4
    syscall

    li $t0, 0                # initilize array index value back
to 0
    li $t2, 0                 # initial size counter back to
zero
    la $t0, array             # load address of array back into
$t0

 sprint:
    lw $t6,($t0)             #load word into temp $t2
    move $a0, $t6            #store it to a safer place
    li $v0, 1                #print it out
    syscall

    la $a0,space             # Display " "
    li $v0,4
    syscall

    add $t0, $t0, 4          #increment the array to the next
index
    add $t2, $t2, 1          #increment the counter by 1

    blt $t2, $t1,sprint      #branch to while if counter < size
of array

    li $t2, 0                 # initial size counter back to
zero
    la $t0, array             # load address of array back into
$t0
    add $t0, $t0, 4          #increment the array to the next
index
    add $t2, $t2, 1          #increment the counter by 1


 loop:  lw $t8,($t0)              # t8 = next element in array
    bge $t8, $t3, notMin      #if array element is >= min goto
notMin
    move $t3,$t8              #min = array[i]
    j notMax
```

```
 notMin: ble $t8,$t4, notMax          #if array element is <=
max goto notMax
    move $t4,$t8              #max = array[i]

 notMax:    add $t2,$t2,1             #incr counter
    add $t0,$t0, 4            #go up in index
    blt $t2, $t1,loop         #if counter < size, go to loop

 eprint:
    la $a0,nextline           # Display "\n"
    li $v0,4
    syscall

    la $a0,min                # Display "min number is "
    li $v0,4
    syscall

    move $a0, $t3             #displays min number in array
    li $v0,1
    syscall

    la $a0,nextline           # Display "\n"
    li $v0,4
    syscall

    la $a0,max                # Display "max number is "
    li $v0,4
    syscall

    move $a0, $t4             #displays max number in array
    li $v0,1
    syscall

    li $v0,10                 # End Of Program
    syscall

.data
array: .space 100
ask: .asciiz "Size of the Array:  "
intask: .asciiz "Enter an Integer: "
min: .asciiz "The minimum number is: "
max: .asciiz "The maximum number is: "
display: .asciiz "Array: "
space: .asciiz " "
nextline: .asciiz "\n"
```

Data Segment:



Text Segment :



Output:

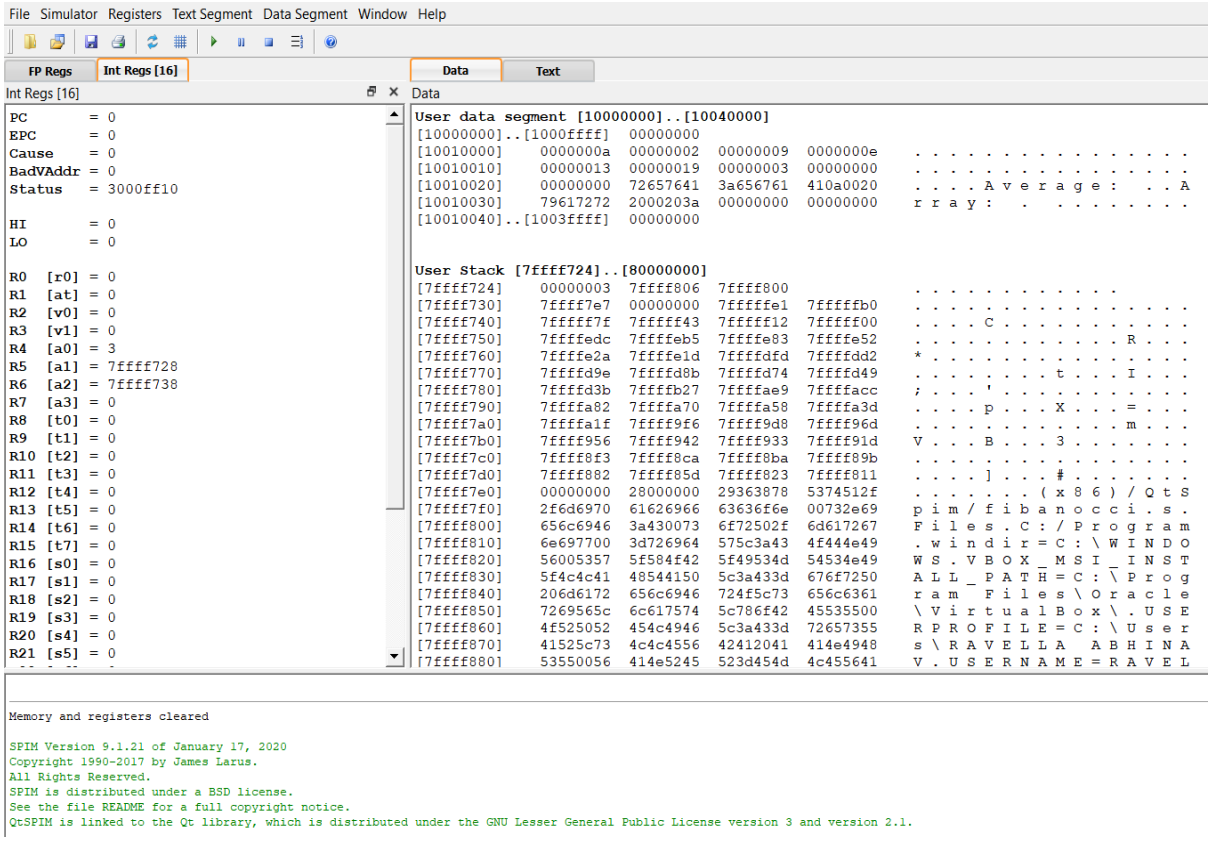
2. **Write a MIPS ALP to find the average of various elements in a array and append the average value at the end of the array. Size and elements in the array can be decided by you.**

   Code :

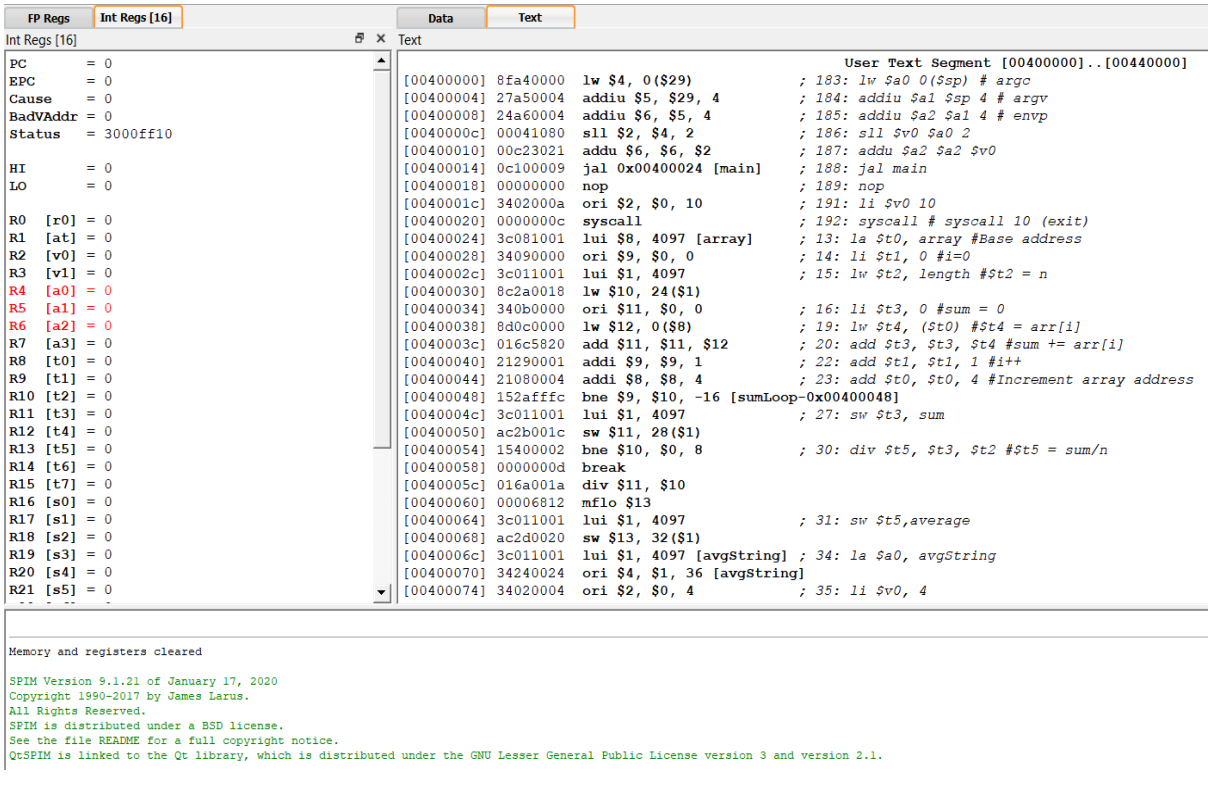
```
.data
    array: .word 10, 2, 9, 14, 19, 25
    length: .word 6
    sum: .word 0
    average: .word 0
    avgString: .asciiz "Average: "
    arrayString: .asciiz "\nArray: "
    space: .asciiz " "

.text

main:
    la $t0, array #Base address
    li $t1, 0 #i=0
    lw $t2, length #$t2 = n
    li $t3, 0 #sum = 0

    sumLoop:
        lw $t4, ($t0) #$t4 = arr[i]
        add $t3, $t3, $t4 #sum += arr[i]

        add $t1, $t1, 1 #i++
```

```
        add $t0, $t0, 4 #Increment array address

        bne $t1, $t2, sumLoop #i != n continue

    sw $t3, sum

    #Average
    div $t5, $t3, $t2 #$t5 = sum/n
    sw $t5,average

    #Print string system call
    la $a0, avgString
    li $v0, 4
    syscall

    #Print integer system call
    li $v0, 1
    add $a0, $t5, 0
    syscall

    #Add integer to array
    sw $t5, ($t0)
    add $t2, $t2, 1 #n+=1

    #Print string system call
    la $a0, arrayString
    li $v0, 4
    syscall

    la $t0, array #baseaddress
    li $t1, 0 #i=0
    displayLoop:
       #Print integer system call
        li $v0, 1

        lw $t4, ($t0)
        add $a0, $t4, 0
        syscall

        #Print space to screen
        la $a0, space
        li $v0, 4
        syscall

        add $t1, $t1, 1 #i++
        add $t0, $t0, 4 #Increment array address
        bne $t1, $t2, displayLoop

    #Exit program
    li $v0, 10
    syscall
```

Data Segment:



Text Segment :



Output:

Input Array : 10, 2, 9, 14, 19, 25