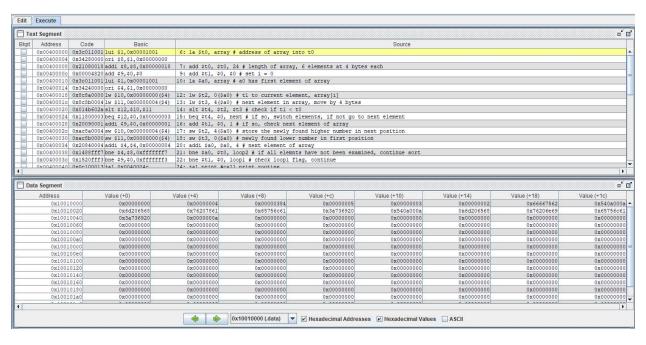
Kevin Martin Syracuse University CIS655 – Summer 2020, Tuesday @ 9:00pm EST Homework 2

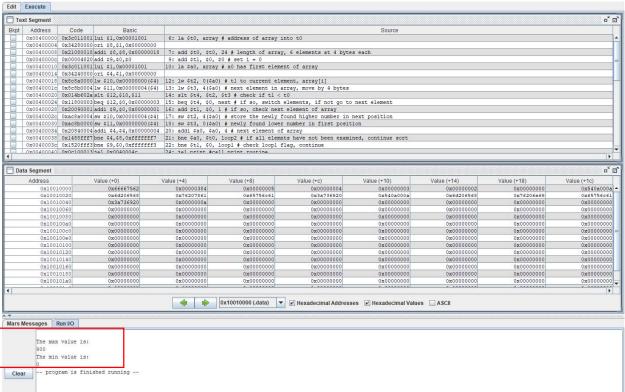
Question 1

To find the min and max from an array in MARS, I used a very simple double loop to sort the array. Then I took the first and last values from the array as the min and max values, respectively. I assumed a fix length of array that was known at the start of the program, which made the indexing much easier. Finally, I added a few lines related to printing the outputs in the I/O window in MARS. While the registers show the correct values, I enjoyed seeing the values printed in the terminal as well. Screenshots below. First, two of the code before it is assembled. Then the text/data segments after it is assembled but before it is run. Finally, the end result after it is successfully ran, including registers.

```
Edit Execute
 HW2_Q1_Kevin_Martin.asm
 2 #array: .word 1,3,5,7,9,11 #change array elements here
 3 array: .word 0,4,900,5,3,2 #change array elements here
 4 .text
 5 main:
 6 la $t0, array # address of array into t0
   add $t0, $t0, 24 # length of array, 6 elements at 4 bytes each
 8 loop1: # check if the entire array has been visited
 9 add $t1, $0, $0 # set i = 0
10 la $a0, array # a0 has first element of array
11 loop2: # sorting
12 lw $t2, 0($a0) # t1 to current element, array[i]
13 lw $t3, 4($a0) # next element in array, move by 4 bytes
14 slt $t4, $t2, $t3 # check if t1 < t0
15 beq $t4, $0, next # if so, switch elements, if not go to next element
16 add $t1, $0, 1 # if so, check next element of array
17 sw $t2, 4($a0) # store the newly found higher number in next position
18 sw $t3, O($aO) # newly found lower number in first position
19 next:
20 addi $a0, $a0, 4 # next element of array
21 bne $a0, $t0, loop2 # if all elemnts have not been examined, continue sort
22 bne $tl, $0, loop1 # check loop1 flag, continue
23
24 jal print #call print routine
25 li $v0, 10 #10 is syscall for exit
26 syscall
27
28 .data
                  .asciiz "buff n"
29 buff:
                 .asciiz "\nThe max value is:\n"
30 max text:
                   .asciiz "\nThe min value is:\n"
31 min text:
32 .text
33 print:
34 li $v0, 4 #syscall 4 is print string
35 la $a0, buff # needed a buffer to print correctly
    externall
Line: 1 Column: 1 Show Line Numbers
```

```
Edit Execute
HW2_Q1_Kevin_Martin.asm
25 li $v0, 10 #10 is syscall for exit
26 syscall
27
28 .data
29 buff:
                   .asciiz "buff\n"
                   .asciiz "\nThe max value is:\n"
30 max_text:
31 min_text:
                   .asciiz "\nThe min value is:\n"
32 .text
33 print:
34 li $v0, 4 #syscall 4 is print string
35 la $aO, buff # needed a buffer to print correctly
36 syscall
37
38 li $v0, 4 #syscall 4 for print string
39 la $a0, max_text # identify max value
40 syscall
41
42 li $v0, 1 # syscall for integer
43 la $a0, array # load sorted array into a0
44 lw $tl, 4($a0) # first element in array
45 add $a0, $t1, $zero # load element into a0 for print
46 syscall
47
48 li $v0, 4 # syscall 4 for print string
49 la $a0, min_text # identify min value
50 syscall
51
52 li $v0, 1 # syscall for integer
53 la $aO, array # load sorted array into aO
54 lw $t2, 24($a0) # last element in array
55 add $a0, $t2, $zero # load element into a0 for print
56 syscall
57
58 jr $ra #jump back to main part of function, ready to exit
59
4
Line: 1 Column: 1 Show Line Numbers
```





Registers	Coproc 1	Coproc 0		
Name		Number		Value
\$zero		0		0x0000000
\$at		1		0x1001000
\$v0		2		0x0000000
\$vl		3		0x0000000
\$a0		4		0x0000000
\$al		5		0x0000000
\$a2		6		0x0000000
\$a3		7		0x000000
\$t0		8		0x100100
\$t1		9		0x0000038
\$t2		10		0x0000000
\$t3		11		0x000000
\$t4		12		0x0000000
\$t5		13		0x000000
\$t6		14		0x0000000
\$t7		15		0x000000
\$80		16		0x0000000
\$31		17		0x000000
\$s2		18		0x0000000
\$83		19		0x000000
\$84		20		0x0000000
\$85		21		0x000000
\$86		22		0x0000000
\$87		23		0x000000
\$t8		24		0x0000000
\$t9		25		0x000000
\$k0		26		0x0000000
\$kl		27		0x000000
\$gp		28		0x1000800
\$sp		29		0x7fffeff
\$fp		30		0x000000
\$ra		31		0x0040004
pc				0x0040004
hi				0x0000000
10				0x0000000