

Lab Report 06

Assignment 1

Code:

Comments:

- Input 1: $(-2, 6, -1, 3, -2)$
⇒ Max sum = $-2 + 6 - 1 + 3 = 6$, w/ length = 4

[illegible]

\$v0 = 4

$$v_1 = 6$$

- Input 2: (2, -3, 2, 5, -4)
 - ⇒ Max sum = 2 + -3 + 2 + 5 = 5, w/ length = 4

\$v0 = 4

$$v_1 = 6$$

Assignment 2

Code:

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

main:	la	\$a0, A	#\$a0 = Address(A[0])
	la	\$a1, Aend	
	addi	\$a1, \$a1, -4	#\$a1 = Address(A[n-1])
	j	sort	#sort
after_sort:	li	\$v0, 10	#exit

syscall

end_main:

#-----

#procedure sort (ascending selection sort using pointer)

#register usage in sort program

#\$a0 pointer to the first element in unsorted part

#\$a1 pointer to the last element in unsorted part

#\$t0 temporary place for value of last element

#\$v0 pointer to max element in unsorted part

#\$v1 value of max element in unsorted part

#-----

sort: beq \$a0, \$a1, done #single element list is sorted

 j max #call the max procedure

after_max: lw \$t0, 0(\$a1) #load last element into \$t0

 sw \$t0, 0(\$v0) #copy last element to max location

 sw \$v1, 0(\$a1) #copy max value to last element

 addi \$a1, \$a1, -4 #decrement pointer to last element

 j print

after_print: j sort #repeat sort for smaller list

done: j after_sort

#-----

#Procedure max

#function: find the value and address of max element in the list

#\$a0 pointer to first element

#\$a1 pointer to last element

#-----

max:

 addi \$v0, \$a0, 0 #init max pointer to first element

 lw \$v1, 0(\$v0) #init max value to first value

```

                                addi      $t0, $a0, 0    #init next pointer to first
loop:
                                beq       $t0, $a1, ret  #if next=last, return
                                addi      $t0, $t0, 4    #advance to next element
                                lw        $t1, 0($t0)    #load next element into $t1
                                slt       $t2, $t1, $v1  #(next)<(max) ?
                                bne       $t2, $zero, loop #if (next)<(max), repeat
                                addi      $v0, $t0, 0    #next element is new max element
                                addi      $v1, $t1, 0    #next value is new max value
                                j         loop           #change completed; now repeat
ret:
                                j         after_max

```

#-----

#Procedure print

#-----

```

print:    add      $s7, $v0, $zero    #Save
          add      $s6, $a0, $zero    #Save
          la       $s0, A             # $s0 = Address(A[0])
          la       $s1, Aend
          addi     $s1, $s1, -4        # $s1 = Address(A[n-1])

loop2:    blt      $s1, $s0, endprint  #exit if Aend < A[i]

          li       $v0, 1             #service 01: print integer
          lw       $a0, 0($s0)        #value of A[i]
          syscall

          li       $v0, 11
          li       $a0, ' '           #print space
          syscall

```

j loop2

```
endprint:    li      $v0, 11
```

```
li      $a0, '\n'      #print enter
```

syscall

```
add    $v0, $s7, $zero    #Load
```

```
add      $a0, $s6, $zero    #Load
```

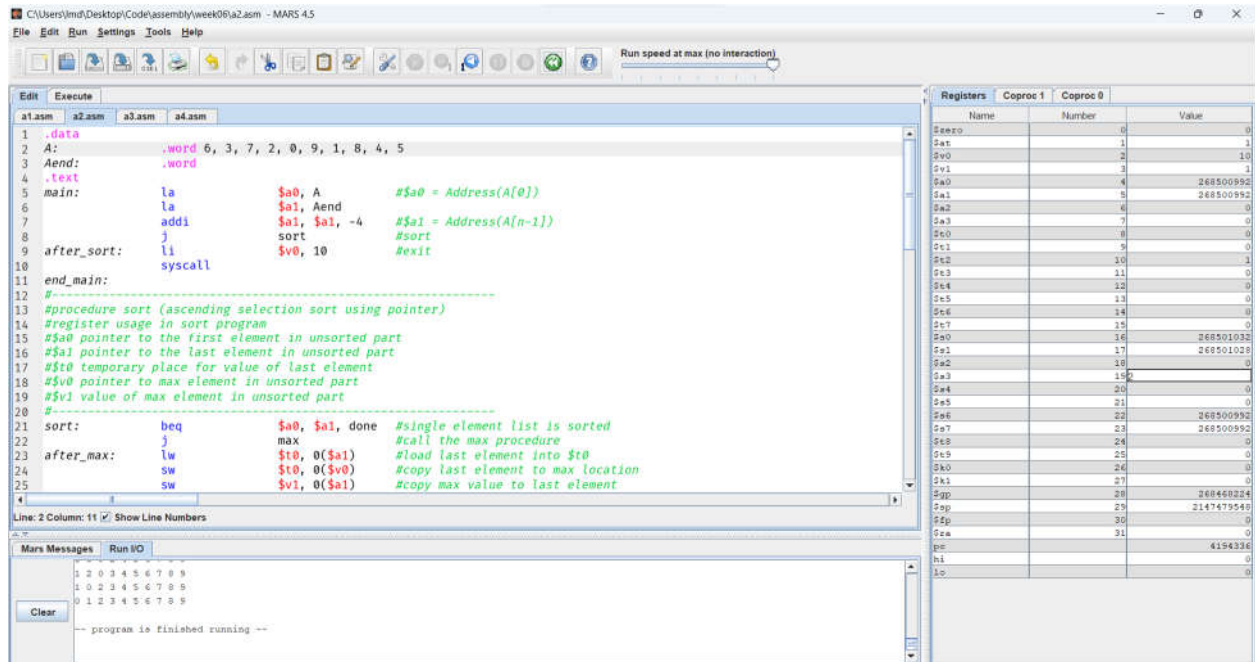
j	after_print
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55
11	89
12	144
13	233
14	377
15	610
16	987
17	1597
18	2584
19	4181
20	6765
21	10946
22	17711
23	28657
24	46368
25	75025
26	121393
27	196418
28	317811
29	514229
30	832040
31	1346269
32	2178309
33	3524558
34	5699067
35	9223685
36	14921352
37	24146091
38	39073723
39	63245984
40	102397085
41	165580149
42	267914286
43	433491860
44	701406145
45	1134903014
46	1836309159
47	2971212173
48	4807521332
49	7778733505
50	12585410206
51	20365731140
52	32951240346
53	53317071486
54	86268311832
55	139583543758
56	225851463690
57	365435311648
58	591286775338
59	956718286976
60	1548007761614
61	2504731968590
62	4052739729204
63	6557471697794
64	10610211416994
65	17167683114788
66	27777894531782
67	44945577646570
68	72723472178352
69	117671149715022
70	190392494721374
71	308063644436396
72	498456139157770
73	798519734594166
74	1276675873751936
75	2039079738346102
76	3246655612098038
77	5185735350444140
78	8224811062542178
79	13010546412990216
80	20735357475532394
81	32745893888522610
82	51481251364054904
83	79226645252577514
84	123707906616632418
85	192934551879190032
86	296642458495822450
87	450577010375012462
88	687219468870832480
89	1047801479245844930
90	1609020948116677410
91	2456822427362522340
92	3745843375479199750
93	5674864802841722090
94	8620708178310921840
95	13015572981152643930
96	19836281159463565770
97	29851854140616209700
98	45088135300080775570
99	68439989440736985300

Comments:

- Input: 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

[illegible]

- Input: 6, 3, 7, 2, 0, 9, 1, 8, 4, 5



Assignment 3

Code:

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

main:

```

lui      $s0, 0x1001      # A[0]
li       $t0, 0           # i = 0
li       $t1, 0           # j = 0
li       $s1, 13          # n = A.length
li       $s2, 13          # n - i for inner loop
add      $t2, $zero, $s0   # For iterating addr by i
add      $t3, $zero, $s0   # For iterating addr by j
addi     $s1, $s1, -1

```

outer_loop:

```

li       $t1, 0           # j = 0
addi     $s2, $s2, -1      # Decrease size for inner_loop

```

```

        add        $t3, $zero, $s0        # Reset addr itr j
inner_loop:
        lw         $s3, 0($t3)            # A[j]
        addi       $t3, $t3, 4            # Addr itr j += 4
        lw         $s4, 0($t3)            # A[j+1]
        addi       $t1, $t1, 1            # j++
        slt        $t4, $s3, $s4          # Set $t4 = 1 if $s3 < $s4
        bne        $t4, $zero, cond
swap:
        sw         $s3, 0($t3)
        sw         $s4, -4($t3)
        lw         $s4, 0($t3)
cond:
        bne        $t1, $s2, inner_loop    #j != n-i
        j          Print
EndPrt:
        addi       $t0, $t0, 1            #i++
        bne        $t0, $s1, outer_loop    #i != n
        li         $t0, 0
        addi       $s1, $s1, 1
exit:
        li         $v0, 10
        syscall

```

```

#-----

```

```

# Procedure Print

```

```

#-----

```

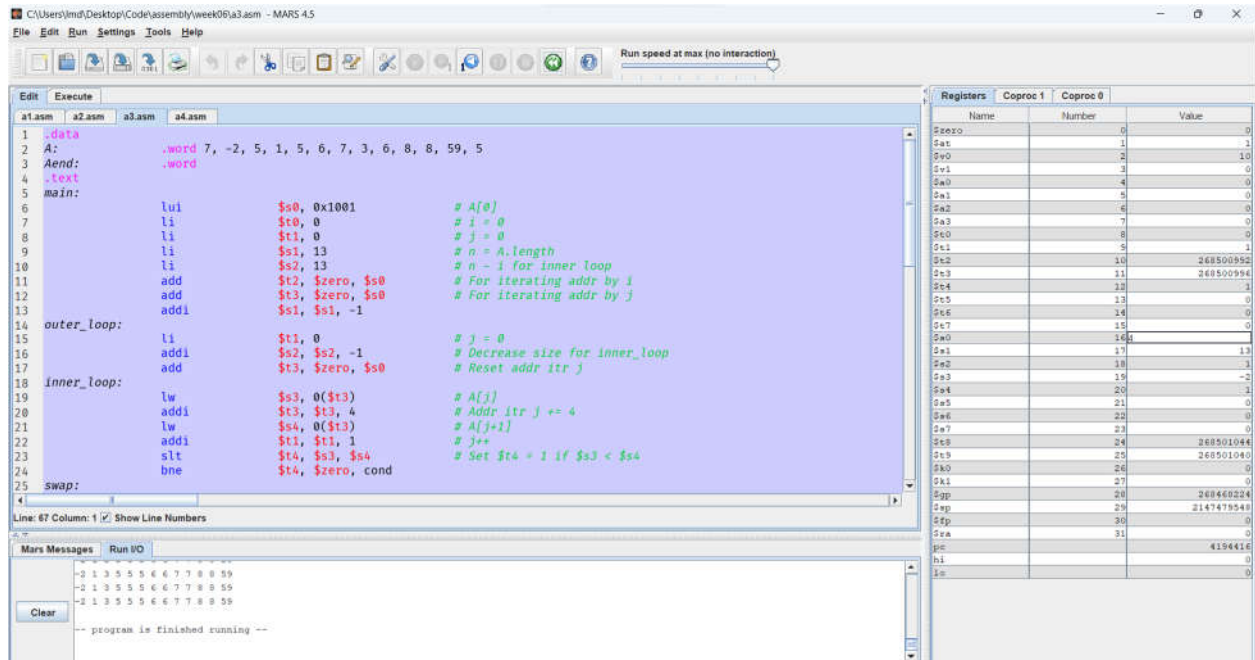
```

Print:    add        $s7, $v0, $zero        # Save
          add        $s6, $a0, $zero        # Save

```

	la	\$t8, A	# \$t8 = Address(A[0])
	la	\$t9, Aend	
	addi	\$t9, \$t9, -4	# \$t9 = Address(A[n-1])
LoopPrt:	blt	\$t9, \$t8, EndLoopPrt	# Exit if Aend < A[i]
	li	\$v0, 1	# Service 01: print integer
	lw	\$a0, 0(\$t8)	# Value of A[i]
	syscall		
	li	\$v0, 11	
	li	\$a0, ' '	# Print space
	syscall		
	addi	\$t8, \$t8, 4	# Move to A[i+1]
	j	LoopPrt	
EndLoopPrt:	li	\$v0, 11	
	li	\$a0, '\n'	# Print enter
	syscall		
	add	\$v0, \$s7, \$zero	# Load
	add	\$a0, \$s6, \$zero	# Load
	j	EndPrt	

Result:



Assignment 4

Code:

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

```

init:      la          $s0, A          # Array A
           li          $s1, 13         # Length of A
           li          $s2, 1          # i
           li          $s3, 0          # j
           li          $s4, 0          # v
           li          $t0, 0          # Address of A[i]
           li          $t1, 0          # Address of A[j]
           li          $t2, 0          # Value of A[i]
           li          $t3, 0          # Value of A[j]

```

```

Loop1:     sll         $t0, $s2, 2     # $t0 = 4 * i
           add         $t0, $t0, $s0   # Address(A[i])

```



```

        lw          $s4, 0($t0)    # v = A[i]
        addi       $s3, $s2, -1    # j = i - 1
        sll        $t1, $s3, 2     # $t1 = 4 * j
        add        $t1, $t1, $s0    # Address(A[j])

Loop2:   lw          $t3, 0($t1)    # Load A[j]
        blt        $t3, $s4, EndL2 # Continue looping if A[j] >= v
        sw          $t3, 4($t1)     # A[j+1] = A[j]
        addi       $s3, $s3, -1    # --j
        addi       $t1, $t1, -4    # Keep memory access consistent with j
        bge        $s3, $0, Loop2  # Loop if j >= 0

EndL2:   sw          $s4, 4($t1)    # A[j+1] = v
        addi       $s2, $s2, 1     # ++i
        j          Print

EndPrt:  blt        $s2, $s1, Loop1 # Loop while i < A.length

Exit:    li          $v0, 10 # Load exit operation
        syscall

#-----
# Procedure Print
#-----

Print:   add        $s7, $v0, $zero # Save
        add        $s6, $a0, $zero # Save
        la         $t8, A           # $t8 = Address(A[0])
        la         $t9, Aend        # $t9 = Address(A[n-1])
        addi       $t9, $t9, -4     # $t9 = Address(A[n-1])

```

LoopPrt:	blt	\$t9, \$t8, EndLoopPrt	# Exit if Aend < A[i]
	li	\$v0, 1	# Service 01: print integer
	lw	\$a0, 0(\$t8)	# Value of A[i]
	syscall		
	li	\$v0, 11	
	li	\$a0, ' '	# Print space
	syscall		
	addi	\$t8, \$t8, 4	# Move to A[i+1]
	j	LoopPrt	
EndLoopPrt:	li	\$v0, 11	
	li	\$a0, '\n'	# Print enter
	syscall		
	add	\$v0, \$s7, \$zero	# Load
	add	\$a0, \$s6, \$zero	# Load
	j	EndPrt	

Result:

C:\Users\lmd\Desktop\Code\assembly\week06\ja4.asm - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

a1.asm a2.asm a3.asm a4.asm

```
1 .data
2 A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 59, 5
3 .end: .word
4 .text
5 init: la $s0, A # Array A
6 li $s1, 13 # Length of A
7 li $s2, 1 # i
8 li $s3, 0 # j
9 li $s4, 0 # v
10 li $t0, 0 # Address of A[i]
11 li $t1, 0 # Address of A[j]
12 li $t2, 0 # Value of A[i]
13 li $t3, 0 # Value of A[j]
14
15 Loop1: sll $t0, $s2, 2 # $t0 = 4 * i
16 add $t0, $t0, $s0 # Address(A[i])
17 lw $s4, 0($t0) # v = A[i]
18 addi $s3, $s2, -1 # j = j - 1
19 sll $t1, $s3, 2 # $t1 = 4 * j
20 add $t1, $t1, $s0 # Address(A[j])
21
22 Loop2: lw $t3, 0($t1) # Load A[j]
23 blt $t3, $s4, EndL2 # Continue looping if A[j] >= v
24 sw $t3, 4($t1) # A[j+1] = A[j]
25 addi $s3, $s3, -1 # --j
26
27 EndL2:
28
29
30
31
32
```

Line: 1 Column: 1 Show Line Numbers

Mars Messages Run IO

Clear

-- program is finished running --

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	268501040
\$t1	9	268501000
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	4
\$a1	17	13
\$a2	18	13
\$a3	19	0
\$t0	20	0
\$t1	21	0
\$t2	22	0
\$t3	23	0
\$t4	24	268501044
\$t5	25	268501040
\$t6	26	0
\$t7	27	0
\$gp	28	268468224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
\$pc		4194438
\$hi		0
\$lo		0