# Report Lab 7

Nguyễn Khánh Nam

20225749

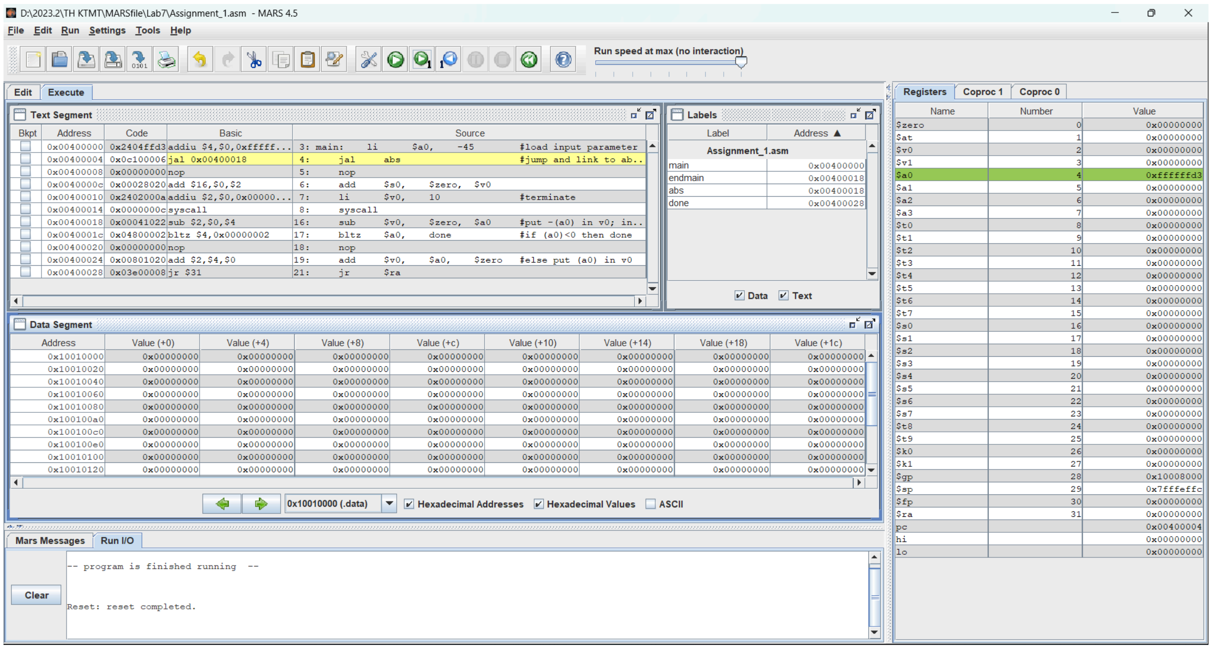
## Assignment 1

Code:

```
    #Laboratory Exercise 7, Assignment 1
.text
main:    li      $a0,    -45        #load input parameter
    jal     abs                     #jump and link to abs procedure
    nop
    add     $s0,    $zero,  $v0
    li      $v0,    10              #terminate
    syscall
endmain:

#---------------------------------------------------------------------
    # function abs
    # param[in] $a0 the interger need to be gained the absolute value
    # return $v0 absolute value

#---------------------------------------------------------------------
abs:
    sub     $v0,    $zero,  $a0     #put -(a0) in v0; in case (a0)<0
    bltz    $a0,    done            #if (a0)<0 then done
    nop
    add     $v0,    $a0,    $zero   #else put (a0) in v0
done:
    jr      $ra
```

Result:

- Trước khi chạy lệnh jal:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_1.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x2404ffd3 | addiu $4,$0,0xfffff... | 3: main: li $a0, -45 #load input parameter |
| ☐ | 0x00400004 | 0x0c100006 | jal 0x00400018 | 4: jal abs #jump and link to ab.. |
| ☐ | 0x00400008 | 0x00000000 | nop | 5: nop |
| ☐ | 0x0040000c | 0x00028020 | add $16,$0,$2 | 6: add $s0, $zero, $v0 |
| ☐ | 0x00400010 | 0x2402000a | addiu $2,$0,0x00000... | 7: li $v0, 10 #terminate |
| ☐ | 0x00400014 | 0x0000000c | syscall | 8: syscall |
| ☐ | 0x00400018 | 0x00041022 | sub $2,$0,$4 | 16: sub $v0, $zero, $a0 #put -(a0) in v0; in.. |
| ☐ | 0x0040001c | 0x04800002 | bltz $4,0x00000002 | 17: bltz $a0, done #if (a0)<0 then done |
| ☐ | 0x00400020 | 0x00000000 | nop | 18: nop |
| ☐ | 0x00400024 | 0x00801020 | add $2,$4,$0 | 19: add $v0, $a0, $zero #else put (a0) in v0 |
| ☐ | 0x00400028 | 0x03e00008 | jr $31 | 21: jr $ra |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_1.asm | |
| main | 0x00400000 |
| endmain | 0x00400018 |
| abs | 0x00400018 |
| done | 0x00400028 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

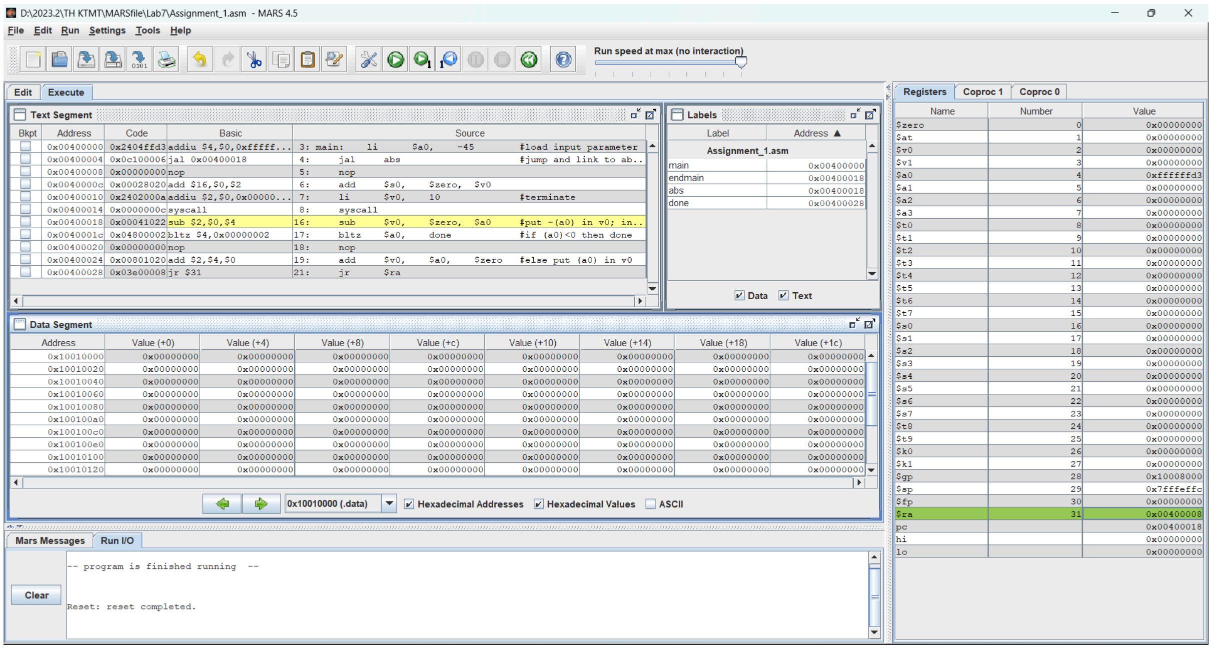
Mars Messages

```
-- program is finished running --

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0xffffffd3 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00000000 |
| pc | | 0x00400004 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

- Sau khi chạy lệnh jal:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_1.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x2404ffd3 | addiu $4,$0,0xfffff... | 3: main: li $a0, -45 #load input parameter |
| ☐ | 0x00400004 | 0x0c100006 | jal 0x00400018 | 4: jal abs #jump and link to ab.. |
| ☐ | 0x00400008 | 0x00000000 | nop | 5: nop |
| ☐ | 0x0040000c | 0x00028020 | add $16,$0,$2 | 6: add $s0, $zero, $v0 |
| ☐ | 0x00400010 | 0x2402000a | addiu $2,$0,0x00000... | 7: li $v0, 10 #terminate |
| ☐ | 0x00400014 | 0x0000000c | syscall | 8: syscall |
| ☐ | 0x00400018 | 0x00041022 | sub $2,$0,$4 | 16: sub $v0, $zero, $a0 #put -(a0) in v0; in.. |
| ☐ | 0x0040001c | 0x04800002 | bltz $4,0x00000002 | 17: bltz $a0, done #if (a0)<0 then done |
| ☐ | 0x00400020 | 0x00000000 | nop | 18: nop |
| ☐ | 0x00400024 | 0x00801020 | add $2,$4,$0 | 19: add $v0, $a0, $zero #else put (a0) in v0 |
| ☐ | 0x00400028 | 0x03e00008 | jr $31 | 21: jr $ra |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_1.asm | |
| main | 0x00400000 |
| endmain | 0x00400018 |
| abs | 0x00400018 |
| done | 0x00400028 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

Mars Messages

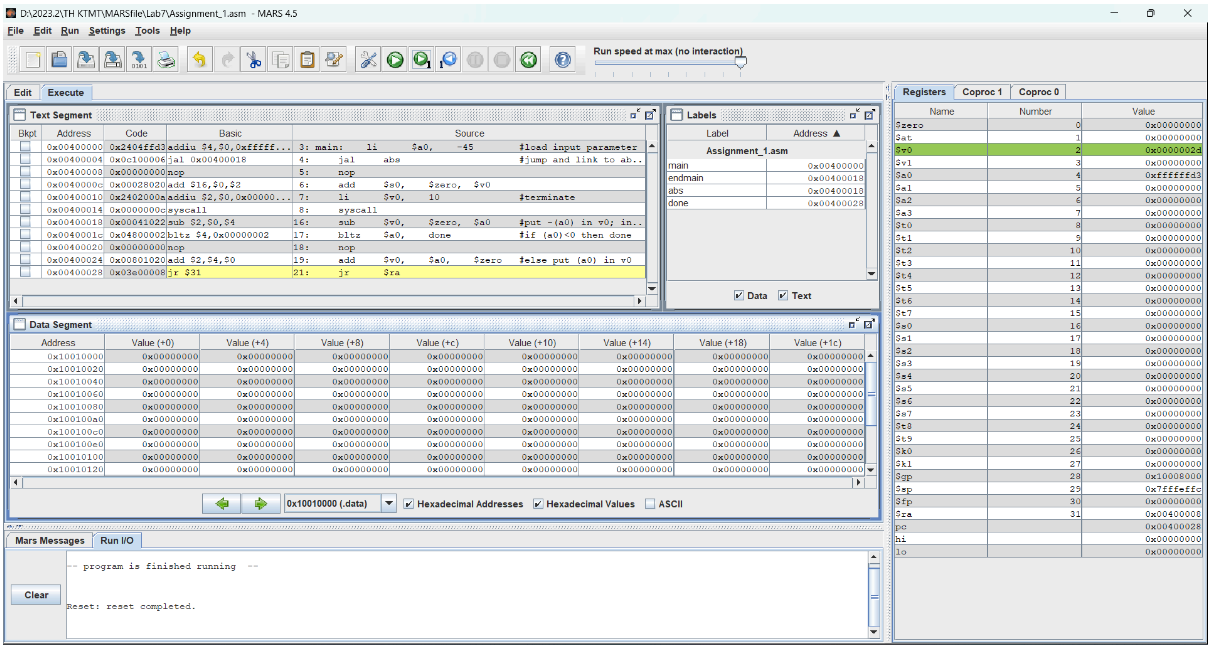
```
-- program is finished running --

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0xffffffd3 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400008 |
| pc | | 0x00400018 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

+ Lệnh jal nhảy chương trình tới địa chỉ được gán abs (0x00400018) bằng cách ghi nó vào thanh ghi pc, đồng thời lưu địa chỉ trở về (0x00400008) vào thanh ghi $ra

- Trước khi chạy lệnh jr $ra:


D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_1.asm - MARS 4.5

Run speed at max (no interaction)

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x2404ffd3 | addiu $4,$0,0xfffff... | 3: main:    li      $a0,    -45         #load input parameter |
| ☐ | 0x00400004 | 0x0c100006 | jal 0x00400018 | 4:      jal     abs                     #jump and link to ab.. |
| ☐ | 0x00400008 | 0x00000000 | nop | 5:      nop |
| ☐ | 0x0040000c | 0x00028020 | add $16,$0,$2 | 6:      add     $s0,    $zero,  $v0 |
| ☐ | 0x00400010 | 0x2402000a | addiu $2,$0,0x00000... | 7:      li      $v0,    10                  #terminate |
| ☐ | 0x00400014 | 0x0000000c | syscall | 8:      syscall |
| ☐ | 0x00400018 | 0x00041022 | sub $2,$0,$4 | 16:     sub     $v0,    $zero,  $a0         #put -(a0) in v0; in.. |
| ☐ | 0x0040001c | 0x04800002 | bltz $4,0x00000002 | 17:     bltz    $a0,    done                #if (a0)<0 then done |
| ☐ | 0x00400020 | 0x00000000 | nop | 18:     nop |
| ☐ | 0x00400024 | 0x00801020 | add $2,$4,$0 | 19:     add     $v0,    $a0,    $zero   #else put (a0) in v0 |
| ☐ | 0x00400028 | 0x03e00008 | jr $31 | 21:     jr      $ra |

Labels

| Label | Address ▲ |
|---|---|
| **Assignment_1.asm** | |
| main | 0x00400000 |
| endmain | 0x00400018 |
| abs | 0x00400018 |
| done | 0x00400028 |

☑ Data ☑ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

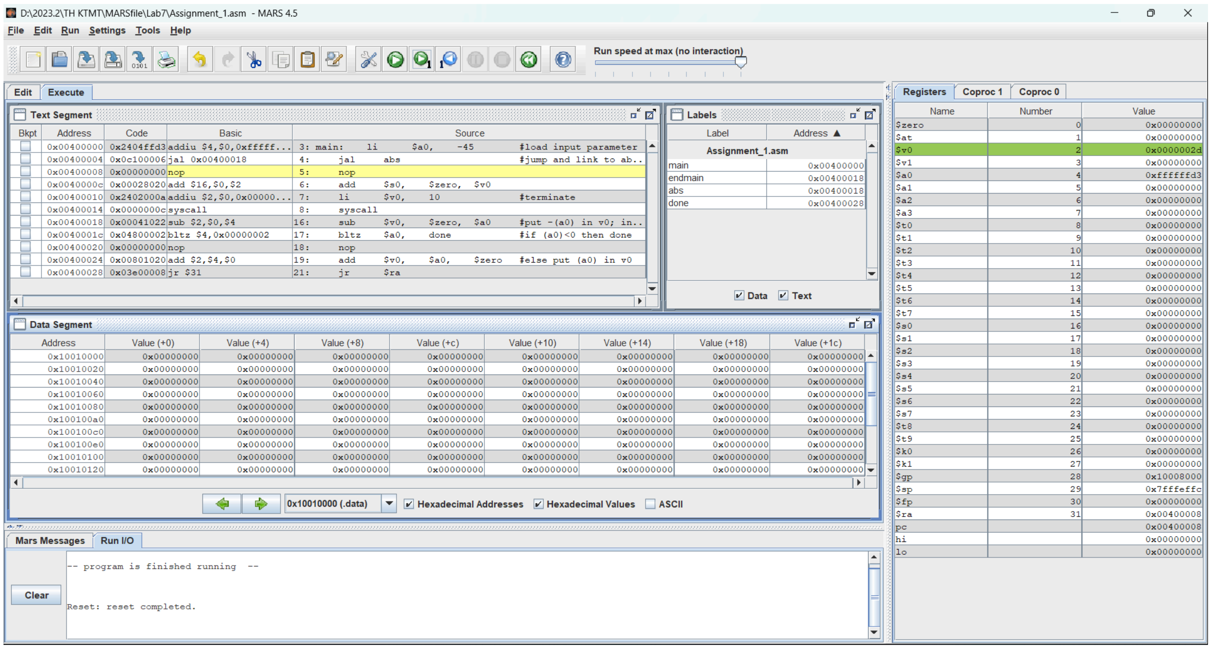
Mars Messages | Run I/O

```
-- program is finished running --

Reset: reset completed.
```

Registers | Coproc 1 | Coproc 0

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x0000002d |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0xffffffd3 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400008 |
| pc | | 0x00400028 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

- Sau khi chạy lệnh jr $ra:


D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_1.asm - MARS 4.5

Run speed at max (no interaction)

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x2404ffd3 | addiu $4,$0,0xfffff... | 3: main:    li      $a0,    -45         #load input parameter |
| ☐ | 0x00400004 | 0x0c100006 | jal 0x00400018 | 4:      jal     abs                     #jump and link to ab.. |
| ☐ | 0x00400008 | 0x00000000 | nop | 5:      nop |
| ☐ | 0x0040000c | 0x00028020 | add $16,$0,$2 | 6:      add     $s0,    $zero,  $v0 |
| ☐ | 0x00400010 | 0x2402000a | addiu $2,$0,0x00000... | 7:      li      $v0,    10                  #terminate |
| ☐ | 0x00400014 | 0x0000000c | syscall | 8:      syscall |
| ☐ | 0x00400018 | 0x00041022 | sub $2,$0,$4 | 16:     sub     $v0,    $zero,  $a0         #put -(a0) in v0; in.. |
| ☐ | 0x0040001c | 0x04800002 | bltz $4,0x00000002 | 17:     bltz    $a0,    done                #if (a0)<0 then done |
| ☐ | 0x00400020 | 0x00000000 | nop | 18:     nop |
| ☐ | 0x00400024 | 0x00801020 | add $2,$4,$0 | 19:     add     $v0,    $a0,    $zero   #else put (a0) in v0 |
| ☐ | 0x00400028 | 0x03e00008 | jr $31 | 21:     jr      $ra |

Labels

| Label | Address ▲ |
|---|---|
| **Assignment_1.asm** | |
| main | 0x00400000 |
| endmain | 0x00400018 |
| abs | 0x00400018 |
| done | 0x00400028 |

☑ Data ☑ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages | Run I/O

```
-- program is finished running --

Reset: reset completed.
```

Registers | Coproc 1 | Coproc 0

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x0000002d |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0xffffffd3 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400008 |
| pc | | 0x00400008 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

+ Lệnh jr $ra lấy địa chỉ được lưu trong thanh ghi $ra (chứa địa chỉ trở về) trả lại cho thanh ghi pc.

- Kết quả chương trình đưa ra giá trị tuyệt đối của -45 vào thanh ghi $v0.

## Assignment 2

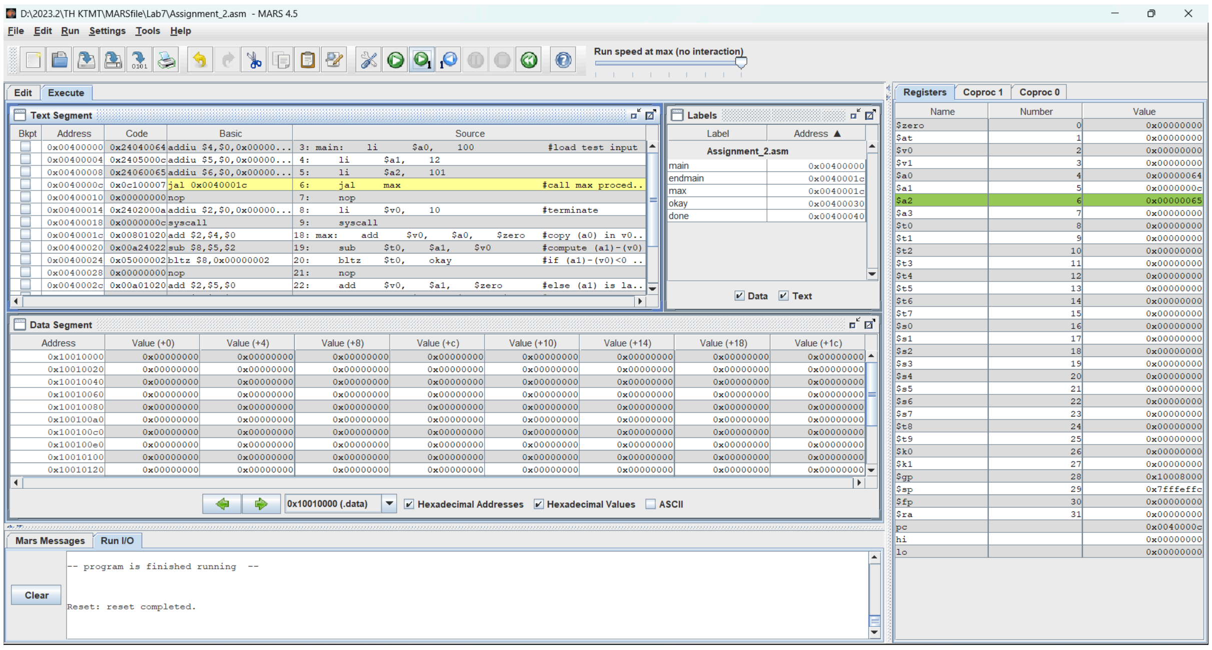
Code:

```
    #Laboratory Exercise 7, Assignment 2
```

```
.text
main:    li      $a0,    100            #load test input
    li      $a1,    12
    li      $a2,    101
    jal     max                         #call max procedure
    nop
    li      $v0,    10                  #terminate
    syscall
endmain:

#-----------------------------------------------------------------------
    #Procedure max: find the largest of three integers
    #param[in] $a0 integers
    #param[in] $a1 integers
    #param[in] $a2 integers
    #return $v0 the largest value

#-----------------------------------------------------------------------
max:    add     $v0,    $a0,    $zero   #copy (a0) in v0; largest so
far
    sub     $t0,    $a1,    $v0         #compute (a1)-(v0)
    bltz    $t0,    okay                #if (a1)-(v0)<0 then no change
    nop
    add     $v0,    $a1,    $zero       #else (a1) is largest thus far
okay:    sub     $t0,    $a2,    $v0    #compute (a2)-(v0)
    bltz    $t0,    done                #if (a2)-(v0)<0 then no change
    nop
    add     $v0,    $a2,    $zero       #else (a2) is largest overall
done:    jr      $ra                    #return to calling program
```

Result:

- Trước khi chạy lệnh jal:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_2.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| | 0x00400000 | 0x24040064 | addiu $4,$0,0x00000... | 3: main: li $a0, 100 #load test input |
| | 0x00400004 | 0x2405000c | addiu $5,$0,0x00000... | 4: li $a1, 12 |
| | 0x00400008 | 0x24060065 | addiu $6,$0,0x00000... | 5: li $a2, 101 |
| | 0x0040000c | 0x0c100007 | jal 0x0040001c | 6: jal max #call max proced.. |
| | 0x00400010 | 0x00000000 | nop | 7: nop |
| | 0x00400014 | 0x2402000a | addiu $2,$0,0x00000... | 8: li $v0, 10 #terminate |
| | 0x00400018 | 0x0000000c | syscall | 9: syscall |
| | 0x0040001c | 0x00801020 | add $2,$4,$0 | 18: max: add $v0, $a0, $zero #copy (a0) in v0.. |
| | 0x00400020 | 0x00a24022 | sub $8,$5,$2 | 19: sub $t0, $a1, $v0 #compute (a1)-(v0) |
| | 0x00400024 | 0x05000002 | bltz $8,0x00000002 | 20: bltz $t0, okay #if (a1)-(v0)<0 .. |
| | 0x00400028 | 0x00000000 | nop | 21: nop |
| | 0x0040002c | 0x00a01020 | add $2,$5,$0 | 22: add $v0, $a1, $zero #else (a1) is la.. |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_2.asm | |
| main | 0x00400000 |
| endmain | 0x0040001c |
| max | 0x0040001c |
| okay | 0x00400030 |
| done | 0x00400040 |

✓ Data ✓ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ASCII

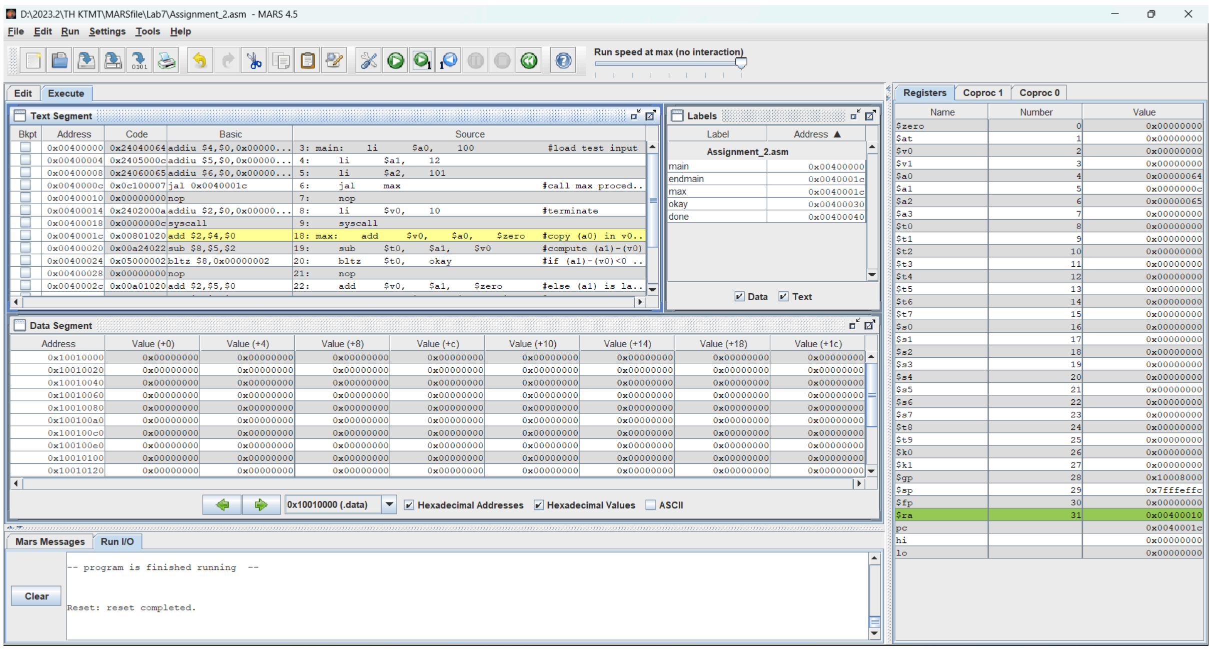
Mars Messages / Run I/O

```
-- program is finished running --

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000064 |
| $a1 | 5 | 0x0000000c |
| $a2 | 6 | 0x00000065 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00000000 |
| pc | | 0x0040000c |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

- Sau khi chạy lệnh jal:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_2.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| | 0x00400000 | 0x24040064 | addiu $4,$0,0x00000... | 3: main: li $a0, 100 #load test input |
| | 0x00400004 | 0x2405000c | addiu $5,$0,0x00000... | 4: li $a1, 12 |
| | 0x00400008 | 0x24060065 | addiu $6,$0,0x00000... | 5: li $a2, 101 |
| | 0x0040000c | 0x0c100007 | jal 0x0040001c | 6: jal max #call max proced.. |
| | 0x00400010 | 0x00000000 | nop | 7: nop |
| | 0x00400014 | 0x2402000a | addiu $2,$0,0x00000... | 8: li $v0, 10 #terminate |
| | 0x00400018 | 0x0000000c | syscall | 9: syscall |
| | 0x0040001c | 0x00801020 | add $2,$4,$0 | 18: max: add $v0, $a0, $zero #copy (a0) in v0.. |
| | 0x00400020 | 0x00a24022 | sub $8,$5,$2 | 19: sub $t0, $a1, $v0 #compute (a1)-(v0) |
| | 0x00400024 | 0x05000002 | bltz $8,0x00000002 | 20: bltz $t0, okay #if (a1)-(v0)<0 .. |
| | 0x00400028 | 0x00000000 | nop | 21: nop |
| | 0x0040002c | 0x00a01020 | add $2,$5,$0 | 22: add $v0, $a1, $zero #else (a1) is la.. |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_2.asm | |
| main | 0x00400000 |
| endmain | 0x0040001c |
| max | 0x0040001c |
| okay | 0x00400030 |
| done | 0x00400040 |

✓ Data ✓ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ASCII

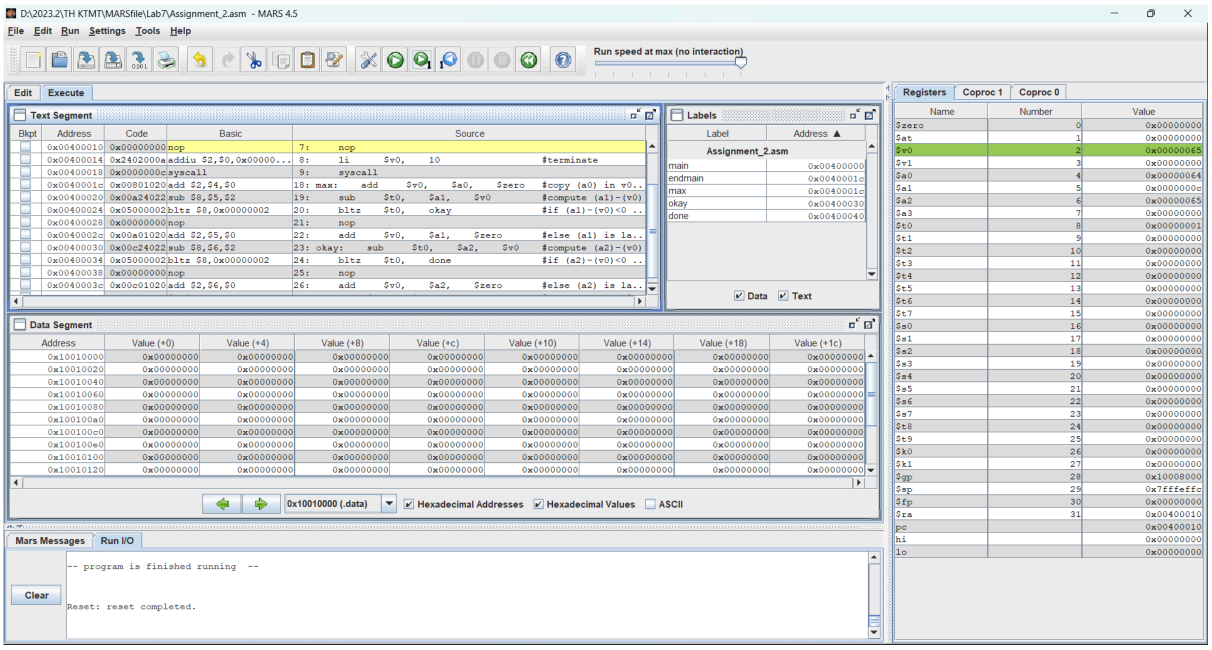
Mars Messages / Run I/O

```
-- program is finished running --

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000064 |
| $a1 | 5 | 0x0000000c |
| $a2 | 6 | 0x00000065 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400010 |
| pc | | 0x0040001c |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

- Kết quả:


D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_2.asm - MARS 4.5

Run speed at max (no interaction)

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| | 0x00400010 | 0x00000000 | nop | 7:     nop |
| | 0x00400014 | 0x2402000a | addiu $2,$0,0x00000... | 8:     li      $v0,    10                    #terminate |
| | 0x00400018 | 0x0000000c | syscall | 9:     syscall |
| | 0x0040001c | 0x00801020 | add $2,$4,$0 | 18: max:    add     $v0,    $a0,    $zero   #copy (a0) in v0.. |
| | 0x00400020 | 0x00a24022 | sub $8,$5,$2 | 19:     sub     $t0,    $a1,    $v0     #compute (a1)-(v0) |
| | 0x00400024 | 0x05000002 | bltz $8,0x00000002 | 20:     bltz    $t0,    okay            #if (a1)-(v0)<0 .. |
| | 0x00400028 | 0x00000000 | nop | 21:     nop |
| | 0x0040002c | 0x00a01020 | add $2,$5,$0 | 22:     add     $v0,    $a1,    $zero   #else (a1) is la.. |
| | 0x00400030 | 0x00c24022 | sub $8,$6,$2 | 23: okay:   sub     $t0,    $a2,    $v0     #compute (a2)-(v0) |
| | 0x00400034 | 0x05000002 | bltz $8,0x00000002 | 24:     bltz    $t0,    done            #if (a2)-(v0)<0 .. |
| | 0x00400038 | 0x00000000 | nop | 25:     nop |
| | 0x0040003c | 0x00c01020 | add $2,$6,$0 | 26:     add     $v0,    $a2,    $zero   #else (a2) is la.. |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_2.asm | |
| main | 0x00400000 |
| endmain | 0x0040001c |
| max | 0x0040001c |
| okay | 0x00400030 |
| done | 0x00400040 |

☑ Data ☑ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Mars Messages | Run I/O

```
-- program is finished running --

Reset: reset completed.
```

Registers

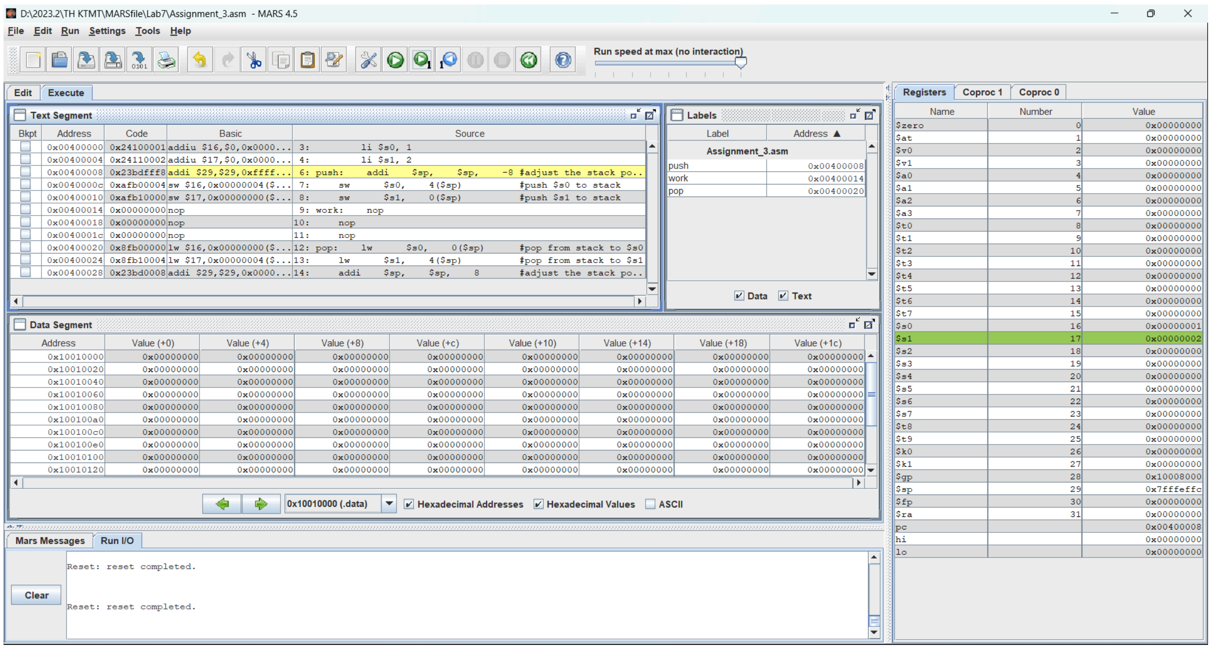
| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000065 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000064 |
| $a1 | 5 | 0x0000000c |
| $a2 | 6 | 0x00000065 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000001 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400010 |
| pc | | 0x00400010 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

# Assignment 3

Code:

```
    #Laboratory Exercise 7, Assignment 3
.text
    li      $s0,    1
    li      $s1,    2

push:    addi    $sp,    $sp,    -8 #adjust the stack pointer
    sw      $s0,    4($sp)        #push $s0 to stack
    sw      $s1,    0($sp)        #push $s1 to stack
work:    nop
    nop
    nop
pop:    lw      $s0,    0($sp)        #pop from stack to $s0
    lw      $s1,    4($sp)        #pop from stack to $s1
    addi    $sp,    $sp,    8        #adjust the stack pointer
```

Result:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_3.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x24100001 | addiu $16,$0,0x0000... | 3: li $s0, 1 |
| ☐ | 0x00400004 | 0x24110002 | addiu $17,$0,0x0000... | 4: li $s1, 2 |
| ☐ | 0x00400008 | 0x23bdfff8 | addi $29,$29,0xffff... | 6: push: addi $sp, $sp, -8 #adjust the stack po.. |
| ☐ | 0x0040000c | 0xafb00004 | sw $16,0x00000004($... | 7: sw $s0, 4($sp) #push $s0 to stack |
| ☐ | 0x00400010 | 0xafb10000 | sw $17,0x00000000($... | 8: sw $s1, 0($sp) #push $s1 to stack |
| ☐ | 0x00400014 | 0x00000000 | nop | 9: work: nop |
| ☐ | 0x00400018 | 0x00000000 | nop | 10: nop |
| ☐ | 0x0040001c | 0x00000000 | nop | 11: nop |
| ☐ | 0x00400020 | 0x8fb00000 | lw $16,0x00000000($... | 12: pop: lw $s0, 0($sp) #pop from stack to $s0 |
| ☐ | 0x00400024 | 0x8fb10004 | lw $17,0x00000004($... | 13: lw $s1, 4($sp) #pop from stack to $s1 |
| ☐ | 0x00400028 | 0x23bd0008 | addi $29,$29,0x0000... | 14: addi $sp, $sp, 8 #adjust the stack po.. |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_3.asm | |
| push | 0x00400008 |
| work | 0x00400014 |
| pop | 0x00400020 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

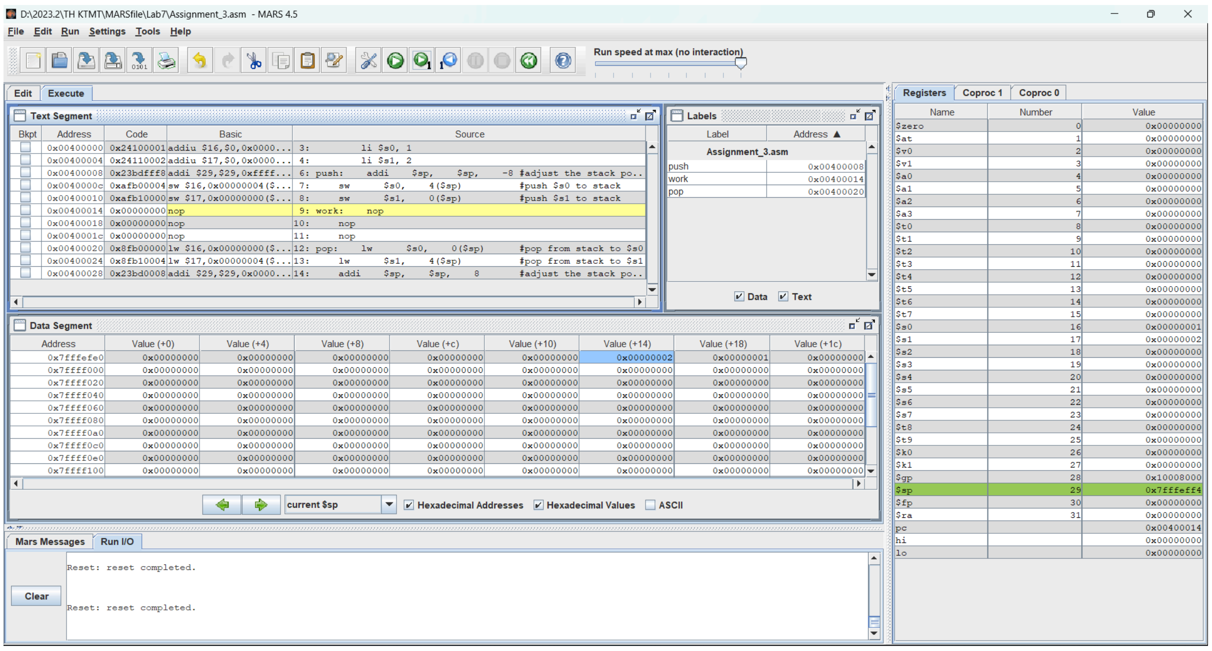
Mars Messages

```
Reset: reset completed.

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000000 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000001 |
| $s1 | 17 | 0x00000002 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00000000 |
| pc | | 0x00400008 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

- Load giá trị vào stack:



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_3.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | Source |
|---|---|---|---|---|
| ☐ | 0x00400000 | 0x24100001 | addiu $16,$0,0x0000... | 3: li $s0, 1 |
| ☐ | 0x00400004 | 0x24110002 | addiu $17,$0,0x0000... | 4: li $s1, 2 |
| ☐ | 0x00400008 | 0x23bdfff8 | addi $29,$29,0xffff... | 6: push: addi $sp, $sp, -8 #adjust the stack po.. |
| ☐ | 0x0040000c | 0xafb00004 | sw $16,0x00000004($... | 7: sw $s0, 4($sp) #push $s0 to stack |
| ☐ | 0x00400010 | 0xafb10000 | sw $17,0x00000000($... | 8: sw $s1, 0($sp) #push $s1 to stack |
| ☐ | 0x00400014 | 0x00000000 | nop | 9: work: nop |
| ☐ | 0x00400018 | 0x00000000 | nop | 10: nop |
| ☐ | 0x0040001c | 0x00000000 | nop | 11: nop |
| ☐ | 0x00400020 | 0x8fb00000 | lw $16,0x00000000($... | 12: pop: lw $s0, 0($sp) #pop from stack to $s0 |
| ☐ | 0x00400024 | 0x8fb10004 | lw $17,0x00000004($... | 13: lw $s1, 4($sp) #pop from stack to $s1 |
| ☐ | 0x00400028 | 0x23bd0008 | addi $29,$29,0x0000... | 14: addi $sp, $sp, 8 #adjust the stack po.. |

Labels

| Label | Address ▲ |
|---|---|
| Assignment_3.asm | |
| push | 0x00400008 |
| work | 0x00400014 |
| pop | 0x00400020 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x7fffefe0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000002 | 0x00000001 | 0x00000000 |
| 0x7ffff000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

current $sp ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

Mars Messages

```
Reset: reset completed.

Reset: reset completed.
```

Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000000 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000000 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000000 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000001 |
| $s1 | 17 | 0x00000002 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeff4 |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00000000 |
| pc | | 0x00400014 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

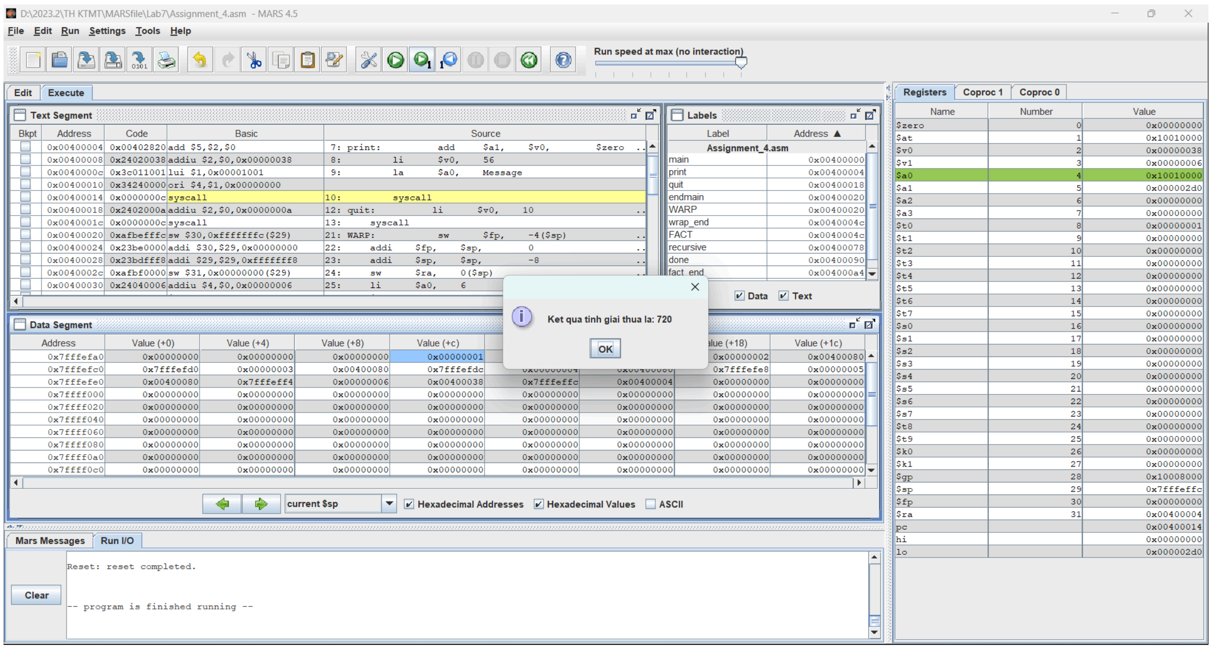
## Assignment 4

Code:

```
    #Laboratory Exercise 7, Home Assignment 4
.data
Message:    .asciiz "Ket qua tinh giai thua la: "
```

```
.text
main:         jal    WARP

print:        add    $a1,    $v0,       $zero   # $a1 = result from N!
        li     $v0,    56
        la     $a0,    Message
        syscall

quit:        li     $v0,    10                #terminate
   syscall

endmain:
   #-----------------------------------------------------------------

   #Procedure WARP: assign value and call FACT
   #-----------------------------------------------------------------

WARP:         sw     $fp,    -4($sp)           #save frame pointer (1)
   addi    $fp,    $sp,       0               #new frame pointer point to the top (2)
   addi    $sp,    $sp,       -8              #adjust stack pointer (3)
   sw      $ra,    0($sp)                      #save return address (4)
   li      $a0,    6                          #load test input N
   jal     FACT                                #call fact procedure
   nop
   lw      $ra,    0($sp)                      #restore return address (5)
   addi    $sp,    $fp,       0               #return stack pointer (6)
   lw      $fp,    -4($sp)                     #return frame pointer (7)
   jr      $ra

wrap_end:
   #-----------------------------------------------------------------

   #Procedure FACT: compute N!
   #param[in] $a0 integer N
   #return $v0 the largest value
   #-----------------------------------------------------------------

FACT:         sw     $fp,    -4($sp)           #save frame pointer
   addi    $fp,    $sp,       0               #new frame pointer point to stack's top
   addi    $sp,    $sp,        -12                 #allocate space for $fp,$ra,$a0 in
stack
   sw      $ra,    4($sp)                          #save return address
   sw      $a0,    0($sp)                          #save $a0 register
   slti    $t0,    $a0,        2                   #if input argument N < 2
   beq     $t0,    $zero,      recursive           #if it is false ((a0 = N) >=2)
   nop
   li      $v0,    1                               #return the result N!=1
   j       done
   nop

recursive:
   addi    $a0,    $a0,        -1                  #adjust input argument
   jal     FACT                                    #recursive call
   nop
   lw      $v1,    0($sp)                          #load a0
   mult    $v1,    $v0                             #compute the result
   mflo    $v0

done:           lw      $ra,    4($sp)             #restore return address
   lw      $a0,    0($sp)                          #restore a0
   addi    $sp,    $fp,        0                   #restore stack pointer
   lw      $fp,    -4($sp)                         #restore frame pointer
   jr      $ra                                     #jump to calling

fact_end:
```

Result:


D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_4.asm - MARS 4.5

Text Segment

```
Bkpt Address    Code       Basic                         Source
     0x00400004 0x00402820 add $5,$2,$0                  7: print:     add   $a1,   $v0,    $zero  ..
     0x00400008 0x24020038 addiu $2,$0,0x00000038        8:       li    $v0,   56
     0x0040000c 0x3c011001 lui $1,0x00001001             9:       la    $a0,   Message
     0x00400010 0x34240000 ori $4,$1,0x00000000
     0x00400014 0x0000000c syscall                       10:      syscall
     0x00400018 0x2402000a addiu $2,$0,0x0000000a        12: quit:      li    $v0,   10       ..
     0x0040001c 0x0000000c syscall                       13:    syscall
     0x00400020 0xafbefffc sw $30,0xfffffffc($29)        21: WARP:      sw    $fp,   -4($sp)  ..
     0x00400024 0x23be0000 addi $30,$29,0x00000000       22:    addi   $fp,   $sp,     0      ..
     0x00400028 0x23bdfff8 addi $29,$29,0xfffffff8       23:    addi   $sp,   $sp,     -8     ..
     0x0040002c 0xafbf0000 sw $31,0x00000000($29)        24:    sw     $ra,   0($sp)          ..
     0x00400030 0x24040006 addiu $4,$0,0x00000006        25:    li     $a0,   6
```

Labels

| Label | Address ▲ |
|---|---|
| Assignment_4.asm | |
| main | 0x00400000 |
| print | 0x00400004 |
| quit | 0x00400018 |
| endmain | 0x00400020 |
| WARP | 0x00400020 |
| wrap_end | 0x0040004c |
| FACT | 0x0040004c |
| recursive | 0x00400078 |
| done | 0x00400090 |
| fact_end | 0x004000a4 |

✓ Data ✓ Text

Ket qua tinh giai thua la: 720

OK

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x7fffefa0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000001 | [illegible] | [illegible] | 0x00000002 | 0x00400080 |
| 0x7fffefc0 | 0x7fffefd0 | 0x00000003 | 0x00400080 | 0x7fffefdc | [illegible] | [illegible] | 0x7fffefe8 | 0x00000005 |
| 0x7fffefe0 | 0x00400080 | 0x7fffeff4 | 0x00000006 | 0x00400038 | 0x7fffeffc | 0x00400004 | 0x00000000 | 0x00000000 |
| 0x7ffff000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

current $sp ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

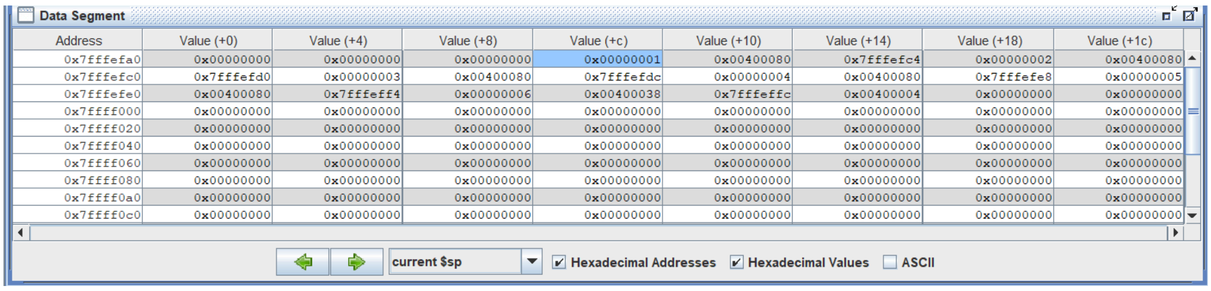
Registers

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x10010000 |
| $v0 | 2 | 0x00000038 |
| $v1 | 3 | 0x00000006 |
| $a0 | 4 | 0x10010000 |
| $a1 | 5 | 0x000002d0 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000001 |
| $t1 | 9 | 0x00000000 |
| $t2 | 10 | 0x00000000 |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00000000 |
| $t5 | 13 | 0x00000000 |
| $t6 | 14 | 0x00000000 |
| $t7 | 15 | 0x00000000 |
| $s0 | 16 | 0x00000000 |
| $s1 | 17 | 0x00000000 |
| $s2 | 18 | 0x00000000 |
| $s3 | 19 | 0x00000000 |
| $s4 | 20 | 0x00000000 |
| $s5 | 21 | 0x00000000 |
| $s6 | 22 | 0x00000000 |
| $s7 | 23 | 0x00000000 |
| $t8 | 24 | 0x00000000 |
| $t9 | 25 | 0x00000000 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x00400004 |
| pc | | 0x00400014 |
| hi | | 0x00000000 |
| lo | | 0x000002d0 |

Mars Messages / Run I/O

```
Reset: reset completed.

-- program is finished running --
```

- Stack:


Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x7fffefa0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000001 | 0x00400080 | 0x7fffefc4 | 0x00000002 | 0x00400080 |
| 0x7fffefc0 | 0x7fffefd0 | 0x00000003 | 0x00400080 | 0x7fffefdc | 0x00000004 | 0x00400080 | 0x7fffefe8 | 0x00000005 |
| 0x7fffefe0 | 0x00400080 | 0x7fffeff4 | 0x00000006 | 0x00400038 | 0x7fffeffc | 0x00400004 | 0x00000000 | 0x00000000 |
| 0x7ffff000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

current $sp ✓ Hexadecimal Addresses ✓ Hexadecimal Values ☐ ASCII

- Với n = 3:

| | |
|---|---|
| 0x7fffeff8 | $fp = 0x00000000 |
| 0x7fffeff4 | $ra = 0x00400004 |
| 0x7fffeff0 | $fp = 0x7fffeffc |
| 0x7fffefec | $ra = 0x00400038 |
| 0x7fffefe8 | $a0 = 0x00000003 |
| 0x7fffefe4 | $fp = 0x7fffeff4 |
| 0x7fffefe0 | $ra = 0x00400080 |
| 0x7fffefdc | $a0 = 0x00000002 |
| 0x7fffefd8 | $fp = 0x7fffefe4 |
| 0x7fffefd4 | $ra = 0x00400080 |
| 0x7fffefd0 | $a0 = 0x00000001 |

## Assignment 5

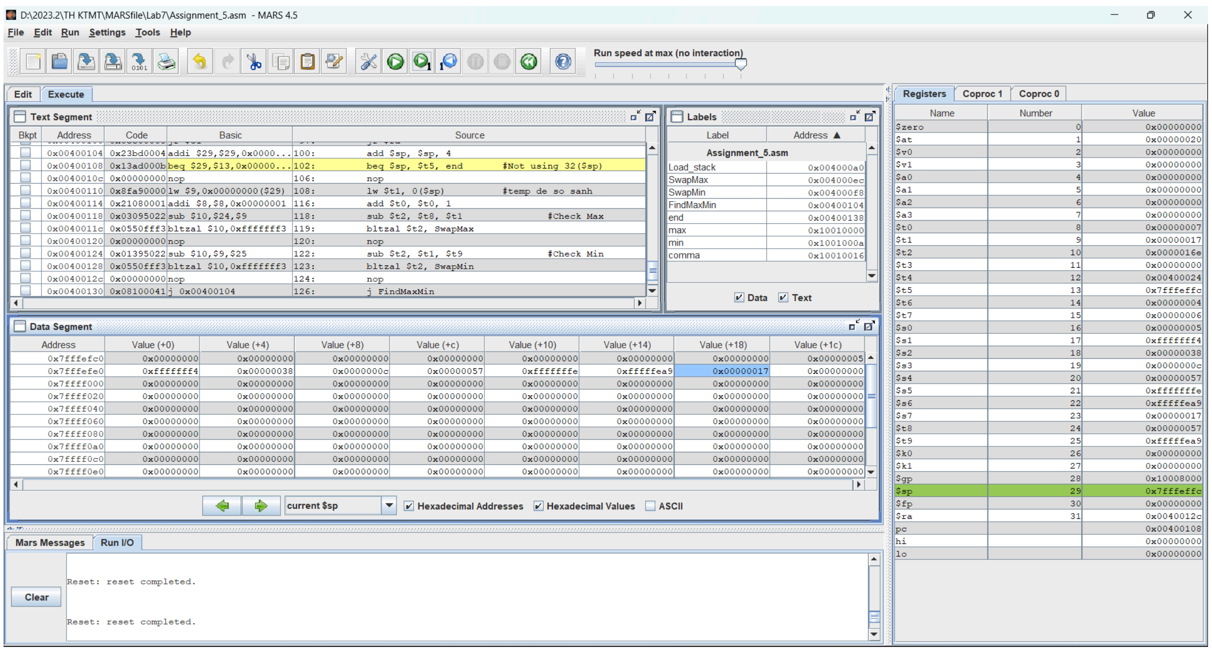
Code:

```
#Assignment 5

.data
	max: .asciiz "Largest: "
	min: .asciiz "\nSmallest: "
	comma: .asciiz ","
.text
#Load
	li $s0, 5
	li $s1, -12
	li $s2, 56
	li $s3, 12
	li $s4, 87
	li $s5, -2
	li $s6, -343
	li $s7, 23

	jal Load_stack
	nop
#----------------------------

#	 $t8 = MAX, $t6 = index of MAX
#	$t9 = MIN, $t7 = index of MIN

#-------------------------------

	li $v0, 4		#Print max
	la $a0, max
	syscall

	li $v0, 1
	add $a0, $t8, $0
	syscall

	li $v0, 4
	la $a0, comma
	syscall
```

```
        li $v0, 1
        add $a0, $t6, $0
        syscall

        li $v0, 4               #Print MIN
        la $a0, min
        syscall

        li $v0, 1
        add $a0, $t9, $0
        syscall

        li $v0, 4
        la $a0, comma
        syscall

        li $v0, 1
        add $a0, $t7, $0
        syscall

        li $v0, 10     #EXIT
        syscall

Load_stack:
        addi $sp, $sp, -32
        sw $s0, 0($sp)
        sw $s1, 4($sp)
        sw $s2, 8($sp)
        sw $s3, 12($sp)
        sw $s4, 16($sp)
        sw $s5, 20($sp)
        sw $s6, 24($sp)
        sw $s7, 28($sp)

        #not using 32($sp)
        la $t5, 32($sp)                 #Save address of 32($sp) -> use address to
stop the program
        add $t4, $ra, $0
        #sw $ra, 32($sp)            #Save return address to print result
        #add $t5, $ra, $0           #Save original address of sp to end program

        li $t6, 0               #Initiate index and min = max = first value
        li $t7, 0
        lw $t8, 0($sp)
        lw $t9, 0($sp)

        li $t0, 0               #i = 0
        j FindMaxMin
        nop

SwapMax:
        add $t6, $t0, $0
        add $t8, $t1, 0
        jr $ra

SwapMin:
        add $t7, $t0, $0
        add $t9, $t1, 0
        jr $ra

FindMaxMin:
        add $sp, $sp, 4

        beq $sp, $t5, end  #Not using 32($sp)

        #lw $t4, 0($sp)             #Check stop find
        #beq $t4, $t5, end
        nop

        lw $t1, 0($sp)              #temp de so sanh

#-----------------------------

#        $t8 = MAX, $t6 = index of MAX
#       $t9 = MIN, $t7 = index of MIN

#--------------------------------
        add $t0, $t0, 1
        sub $t2, $t8, $t1           #Check Max
        bltzal $t2, SwapMax
        nop

        sub $t2, $t1, $t9           #Check Min
        bltzal $t2, SwapMin
        nop

        j FindMaxMin
        nop

end:
        add $ra, $t4, $0            #Not using 32($sp)
        #lw $ra, 0($sp)
        jr $ra
```

Result:

- Arr = {5. -12, 56, 12, 87, -2, -343, 23} được khởi tạo $s0 - $s7



D:\2023.2\TH KTMT\MARSfile\Lab7\Assignment_5.asm - MARS 4.5

Text Segment

| Bkpt | Address | Code | Basic | | Source |
|---|---|---|---|---|---|
| ☐ | 0x00400104 | 0x23bd0004 | addi $29,$29,0x0000... | 100: | add $sp, $sp, 4 |
| ☐ | 0x00400108 | 0x13ad000b | beq $29,$13,0x00000... | 102: | beq $sp, $t5, end #Not using 32($sp) |
| ☐ | 0x0040010c | 0x00000000 | nop | 106: | nop |
| ☐ | 0x00400110 | 0x8fa90000 | lw $9,0x00000000($29) | 108: | lw $t1, 0($sp) #temp de so sanh |
| ☐ | 0x00400114 | 0x21080001 | addi $8,$8,0x00000001 | 116: | add $t0, $t0, 1 |
| ☐ | 0x00400118 | 0x03095022 | sub $10,$24,$9 | 118: | sub $t2, $t8, $t1 #Check Max |
| ☐ | 0x0040011c | 0x0550fff3 | bltzal $10,0xfffffff3 | 119: | bltzal $t2, SwapMax |
| ☐ | 0x00400120 | 0x00000000 | nop | 120: | nop |
| ☐ | 0x00400124 | 0x01395022 | sub $10,$9,$25 | 122: | sub $t2, $t1, $t9 #Check Min |
| ☐ | 0x00400128 | 0x0550fff3 | bltzal $10,0xfffffff3 | 123: | bltzal $t2, SwapMin |
| ☐ | 0x0040012c | 0x00000000 | nop | 124: | nop |
| ☐ | 0x00400130 | 0x08100041 | j 0x00400104 | 126: | j FindMaxMin |

Labels

| Label | Address ▲ |
|---|---|
| **Assignment_5.asm** | |
| Load_stack | 0x004000a0 |
| SwapMax | 0x004000ec |
| SwapMin | 0x004000f8 |
| FindMaxMin | 0x00400104 |
| end | 0x00400138 |
| max | 0x10010000 |
| min | 0x1001000a |
| comma | 0x10010016 |

[✓ Data] [✓ Text]

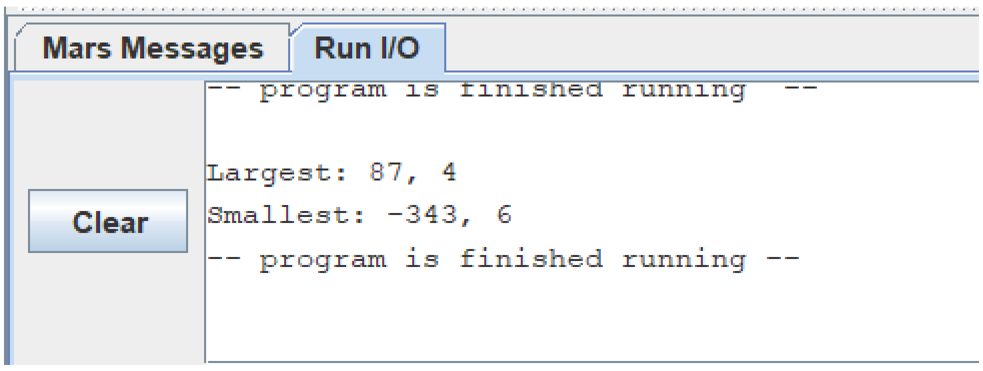
Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x7fffefc0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000005 |
| 0x7fffefe0 | 0xfffffff4 | 0x00000038 | 0x0000000c | 0x00000057 | 0xfffffffe | 0xfffffea9 | 0x00000017 | 0x00000000 |
| 0x7ffff000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x7ffff0e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

current $sp [✓ Hexadecimal Addresses] [✓ Hexadecimal Values] ☐ ASCII

```
Reset: reset completed.

Reset: reset completed.
```

| Name | Number | Value |
|---|---|---|
| $zero | 0 | 0x00000000 |
| $at | 1 | 0x00000020 |
| $v0 | 2 | 0x00000000 |
| $v1 | 3 | 0x00000000 |
| $a0 | 4 | 0x00000000 |
| $a1 | 5 | 0x00000000 |
| $a2 | 6 | 0x00000000 |
| $a3 | 7 | 0x00000000 |
| $t0 | 8 | 0x00000007 |
| $t1 | 9 | 0x00000017 |
| $t2 | 10 | 0x0000016e |
| $t3 | 11 | 0x00000000 |
| $t4 | 12 | 0x00400024 |
| $t5 | 13 | 0x7fffeffc |
| $t6 | 14 | 0x00000004 |
| $t7 | 15 | 0x00000006 |
| $s0 | 16 | 0x00000005 |
| $s1 | 17 | 0xfffffff4 |
| $s2 | 18 | 0x00000038 |
| $s3 | 19 | 0x0000000c |
| $s4 | 20 | 0x00000057 |
| $s5 | 21 | 0xfffffffe |
| $s6 | 22 | 0xfffffea9 |
| $s7 | 23 | 0x00000017 |
| $t8 | 24 | 0x00000057 |
| $t9 | 25 | 0xfffffea9 |
| $k0 | 26 | 0x00000000 |
| $k1 | 27 | 0x00000000 |
| $gp | 28 | 0x10008000 |
| $sp | 29 | 0x7fffeffc |
| $fp | 30 | 0x00000000 |
| $ra | 31 | 0x0040012c |
| pc | | 0x00400108 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |



```
-- program is finished running  --

Largest: 87, 4
Smallest: -343, 6
-- program is finished running --
```