

## Solution: HW#5 CSC390

HW#5\_Q1:

```
1  # Procedure Example (Nested Procedure)
2  # Call a function to perform  $n+(n-1)$ 
3  .data
4  n: .word 5
5  F: .word 0 # store the result
6  .text
7  # put the data in the argument register.
8  # arguments registers are used to pass-parameters in the procedure (function)
9  lw $a0, n
10 la $s0, F # store the location of F to store results
11
12 # Call the sum nested_procedure
13 jal sum
14 lw $s0, 8($sp) #Restore the address of F
15 sw $v0, 0($s0) # store the return value (i.e result) from the function into F
16 j halt
17 |
18 #the nested_procedure (sum)
19
```

```

20 sum:
21 addi $sp, $sp, -12 # reserve space in the stack to store $s0, $a0 and $ra
22                               # adjust stack for 3 items
23 sw    $s0, 8($sp)   # store the
24 sw    $ra, 4($sp)   # save return address
25 sw    $a0, 0($sp)   # save argument
26 slti  $t0, $a0, 1   # test for n < 1
27 beq   $t0, $zero, L1
28 addi  $s0, $zero, 0 # if so, result is 0
29 jr    $ra           # return to line 32
30 L1: addi $a0, $a0, -1 # else decrement n
31 jal   sum           # recursive call
32 addi  $sp, $sp, 12  # Increment stack by 12
33 lw    $ra, 4($sp)   # Restore $ra
34 lw    $a0, 0($sp)   # restore $a0
35 addu  $s0, $a0, $s0 # Add to get result
36 add   $v0, $s0, $zero # Copy the result to $V0
37
38 jr    $ra           # and return
39 halt:
40 nop

```

Sum is called for the first time:

C:\Users\Brinda\GoogleDrive\_CUA\CSC390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Execute

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x3c011001	lui \$1,0x00001001	9: lw \$a0, n
	0x00400004	0x8c240000	lw \$4,0x00000000(\$1)	
	0x00400008	0x3c011001	lui \$1,0x00001001	10: la \$s0, F # store the location of F to store results
	0x0040000c	0x34300004	ori \$16,\$1,0x00000004	
	0x00400010	0x0c100008	jal 0x00400020	13: jal sum
	0x00400014	0x8fb00008	lw \$16,0x00000008(\$29)	14: lw \$s0, 8(\$sp) #Restore the address of F
	0x00400018	0xae020000	sw \$2,0x00000000(\$16)	15: sw \$v0, 0(\$s0) # store the return value (i.e result) from the function into F
	0x0040001c	0x01000103	0x00400060	16: j halt
	0x00400020	0x23b0fff4	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$s0, \$a0 and \$ra
	0x00400024	0xafb00008	sw \$16,0x00000008(\$29)	23: sw \$s0, 8(\$sp) # store the
	0x00400028	0xafb00004	sw \$31,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address
	0x0040002c	0xafaf0000	sw \$4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument
	0x00400030	0x28880001	slti \$8,\$4,0x00000001	26: slti \$t0, \$a0, 1 # test for n < 1
	0x00400034	0x11000002	beq \$8,\$0,0x00000002	27: beq \$t0, \$zero, L1
	0x00400038	0x20100000	addi \$16,\$0,0x00000000	28: addi \$a0, \$zero, 0 # if so, result is 0

Labels

Label	Address
sum	0x00400020
L1	0x00400040
halt	0x00400060
n	0x10010000
F	0x10010004

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000005
\$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	0x10010004
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00400014
pc		0x00400020
hi		0x00000000
lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000005	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
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

Run IO

Clear

-- program is finished running (dropped off bottom) --

\$pc has the current location, \$ra contains the return address to the main program and \$sp has the highest location of the memory.

During the execution of sum procedure (1<sup>st</sup> iteration), the stack memory location is loaded the values of return address (\$ra) to the main program and the current valu of n (\$a0).

C:\Users\Brinda\GoogleDrive\_CUA\CSC390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Execute

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
	0x00400008	0x1e01001	lui \$1,0x000001001	10: la \$a0, F # store the location of F to store results
	0x0040000c	0x34300004	ori \$16,\$1,0x00000004	
	0x00400010	0x0c100008	jal 0x00400020	13: jal sum
	0x00400014	0x8fb00008	lw \$16,0x00000008(\$29)	14: lw \$a0, 8(\$sp) #Restore the address of F
	0x00400018	0xae020000	sw \$2,0x00000000(\$16)	15: sw \$v0, 0(\$a0) # store the return value (i.e result) from the function into F
	0x0040001c	0x08100018	j 0x00400060	16: j halt
	0x00400020	0x23bdfff4	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$a0, \$a0 and \$ra
	0x00400024	0xafb00008	sw \$16,0x00000008(\$29)	23: sw \$a0, 8(\$sp) # store the
	0x00400028	0xafb00004	sw \$31,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address
	0x0040002c	0xaf400000	sw \$4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument
	0x00400030	0x28800001	slli \$8,\$4,0x00000001	26: slli \$a0, \$a0, 1 # test for n < 1
	0x00400034	0x11000002	beq \$9,\$0,0x00000002	27: beq \$t0, \$zero, L1
	0x00400038	0x20100000	addi \$16,\$0,0x00000000	28: addi \$a0, \$zero, 0 # if so, result is 0
	0x0040003c	0x03e00008	jr \$31	29: jr \$ra # return to line 32
	0x00400040	0x2084fff4	addi \$4,\$4,0xfffffff4	30: li: addi \$a0, \$a0, -1 # else decrement n

Labels

Label	Address
sum	0x00400020
L1	0x00400040
halt	0x00400060
n	0x10010004
F	0x10010004

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7ffffef0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000005	0x00400014	0x10010004	0x00000000
0x7fffff00	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff20	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff40	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff60	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff80	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffffa0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffffc0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffffe0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff00	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Clear -- program is finished running (dropped off bottom) --

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000005
\$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
\$a0	16	0x10010004
\$a1	17	0x00000000
\$a2	18	0x00000000
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7fffff80
\$fp	30	0x00000000
\$ra	31	0x00400014
pc		0x00400040
hi		0x00000000
lo		0x00000000

Type here to search

9:17 PM 2/20/2018

2<sup>nd</sup> Iteration: \$ra has now the return address to the sum procedure location (recursive operation). \$sp has the value decremented by 12.

C:\Users\Brinda\GoogleDrive\_CUA\CS390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
0x00400006	0x34300004	ori \$t6,\$1,0x00000004		
0x00400010	0x0c100008	jal 0x00400020	13: jal sum	
0x00400014	0x8fb00008	lw \$t6,0x00000008(\$29)	14: lw \$a0, 8(\$sp) #Restore the address of F	
0x00400018	0xae020000	sw \$2,0x00000000(\$t6)	15: sw \$v0, 0(\$a0) # store the return value (i.e result) from the function into F	
0x0040001e	0x08100010	j 0x00400060	16: j halt	
0x00400020	0x23b0fff4	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$a0, \$a0 and \$ra	
0x00400024	0xafb00008	sw \$t6,0x00000008(\$29)	23: sw \$a0, 8(\$sp) # store the	
0x00400028	0xafb00004	sw \$31,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address	
0x0040002c	0xafaf0000	sw \$4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument	
0x00400030	0x28880001	slti \$t0,\$4,0x00000001	26: slti \$t0, \$a0, 1 # test for n < 1	
0x00400034	0x11000002	beq \$t0,\$0,0x00000002	27: beq \$t0, \$zero, L1	
0x00400038	0x20100000	addi \$t6,\$0,0x00000000	28: addi \$a0, \$zero, 0 # if so, result is 0	
0x0040003c	0x03e00008	jr \$31	29: jr \$ra # return to line 32	
0x00400040	0x2084ffff	addi \$4,\$4,0xffffffffff	30: li: addi \$a0, \$a0, -1 # else decrement n	
0x00400044	0x0c100008	jal 0x00400020	31: jal sum # recursive call	

Labels

Label	Address
sum	0x00400020
L1	0x00400040
halt	0x00400060
n	0x10010000
F	0x10010004

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000004
\$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	0x10010004
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffff0
\$fp	30	0x00000000
\$ra	31	0x00400048
pc		0x00400020
hi		0x00000000
lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7ffffef0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000005	0x00400014	0x10010004	0x00000000
0x7ffffef4	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef8	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff00	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff04	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff08	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff0c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff10	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff14	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7fffff18	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages

Run IO

Clear

-- program is finished running (dropped off bottom) --

Type here to search

9:18 PM 2/20/2018

The image shows the MARS MIPS simulator interface. At the top, the title bar reads "C:\Users\Brinda\Google Drive\_CUA\HW530\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5". Below the title bar is a menu bar with "File", "Edit", "Run", "Settings", "Tools", and "Help". A toolbar with various icons is located below the menu bar. The main window is divided into several panes. The top-left pane is titled "Text Segment" and contains assembly code with columns for "Bkpt", "Address", "Code", and "Source". The code includes instructions like "ori", "jal", "lw", "sw", "beq", "addi", "jr", and "jal" with comments. The top-right pane is titled "Registers" and shows a table of registers with columns for "Name", "Number", and "Value". Registers like \$zero, \$at, \$v0, \$v1, \$a0, \$a1, \$a2, \$a3, \$t0, \$t1, \$t2, \$t3, \$t4, \$t5, \$t6, \$t7, \$s0, \$s1, \$s2, \$s3, \$s4, \$s5, \$s6, \$s7, \$s8, \$s9, \$k0, \$k1, \$gp, \$sp, \$fp, \$ra, \$pc, \$hi, and \$lo are listed. The bottom-left pane is titled "Data Segment" and shows a table of memory addresses with columns for "Address", "Value (+0)", "Value (+4)", "Value (+8)", "Value (+c)", "Value (+10)", "Value (+14)", "Value (+18)", and "Value (+1c)". The bottom-right pane is titled "Mars Messages" and shows a message: "program is finished running (dropped off bottom)". The status bar at the bottom of the window shows the Windows taskbar with various icons and the system clock displaying "9:20 PM 2/20/2018".

## Values of the stack memory location when n decremented to 0

C:\Users\Brinda\GoogleDrive\_CUA\CSC390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Execute

Text Segment

Program Arguments:

Byte	Address	Code	Basic	Source
0x00400020	0x23bdfbf4	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$a0, \$a0 and \$ra	
0x00400024	0xafbf0008	sw \$16,0x00000008(\$29)	23: sw \$a0, 8(\$sp) # store the	
0x00400028	0xafbf0004	sw \$31,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address	
0x0040002c	0xafbf0000	sw \$4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument	
0x00400030	0x28880001	slti \$8,\$4,0x00000001	26: slti \$t0, \$a0, 1 # test for n < 1	
0x00400034	0x11000002	beq \$8,\$0,0x00000002	27: beq \$t0, \$zero, L1	
0x00400038	0x20100000	addi \$16,\$0,0x00000000	28: addi \$a0, \$zero, 0 # if so, result is 0	
0x0040003c	0x03e00000	jr \$31	29: jr \$ra	
0x00400040	0x2084ffff	addi \$4,\$4,0xffffffff	30: li: addi \$a0, \$a0, -1 # else decrement n	
0x00400044	0x0c100008	jal 0x00400020	31: jal sum # recursive call	
0x00400048	0x23bd000c	addi \$29,\$29,0x0000...	32: addi \$sp, \$sp, 12 # Increment stack by 12	
0x0040004c	0x8fbf0004	lw \$31,0x00000004(\$29)	33: lw \$ra, 4(\$sp) # Restore \$ra	
0x00400050	0x8fa40000	lw \$4,0x00000000(\$29)	34: lw \$a0, 0(\$sp) # restore \$a0	
0x00400054	0x00908021	addu \$16,\$4,\$16	35: addu \$a0, \$a0, \$a0 # Add to get result	
0x00400058	0x02001020	add \$2,\$16,\$0	36: add \$v0, \$a0, \$zero # Copy the result to \$V0	

Labels

Label	Address
sum	0x00400020
L1	0x00400040
halt	0x00400060
n	0x10010000
F	0x10010004

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000001
\$a1	5	0x00000000
\$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	0x7ffffefc0
\$fp	30	0x00000000
\$ra	31	0x00400048
pc		0x00400058
hi		0x00000000
lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7ffffefa0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00400048	0x10010004
0x7ffffefc0	0x00000001	0x00400048	0x10010004	0x00000002	0x00400048	0x10010004	0x00000003	0x00400048
0x7ffffefe0	0x10010004	0x00000004	0x00000004	0x10010004	0x00000005	0x00400014	0x10010004	0x00000000
0x7ffffef00	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef20	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef40	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef60	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef80	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef0c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef10	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef12	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffffef14	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages Run I/O

Clear -- program is finished running (dropped off bottom) --

Contents of \$ra, \$pc, and \$sp before returning to the main program:

C:\Users\Brinda\GoogleDrive\_CUA\CSC390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Text Segment**

Program Arguments:

Byte	Address	Code	Basic	Source
0x00400014	0x00400014	lw \$t6,0x00000008(\$29)	14: lw \$t6, 8(\$sp) #Restore the address of F	
0x00400018	0x00400018	sw \$t6,0x00000000(\$t6)	15: sw \$t6, 0(\$t6) # store the return value (i.e result) from the function into F	
0x0040001c	0x0040001c	j halt	16: j halt	
0x00400020	0x23bdffff	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$a0, \$a0 and \$ra	
0x00400024	0xafb00008	sw \$t6,0x00000008(\$29)	23: sw \$t6, 8(\$sp) # store the	
0x00400028	0xafb00004	sw \$t1,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address	
0x0040002c	0xafb00000	sw \$t4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument	
0x00400030	0x28880001	slli \$t0,\$t4,0x00000001	26: slli \$t0, \$a0, 1 # test for n < 1	
0x00400034	0x11000002	beq \$t0,\$t0,0x00000002	27: beq \$t0, \$zero, L1	
0x00400038	0x20100000	addi \$t6,\$t6,0x00000000	28: addi \$a0, \$zero, 0 # if so, result is 0	
0x0040003c	0x03e00008	jr \$31	29: jr \$ra # return to line 32	
0x00400040	0x2084ffff	addi \$t4,\$t4,0xffffffff	30: li: addi \$a0, \$a0, -1 # else decrement n	
0x00400044	0x0c100008	jal 0x00400020	31: jal sum # recursive call	
0x00400048	0x23bd0000	addi \$29,\$29,0x0000...	32: addi \$sp, \$sp, 12 # Increment stack by 12	
0x0040004c	0xafb00004	lw \$t1,0x00000004(\$29)	33: lw \$ra, 4(\$sp) # Restore \$ra	

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7fffffa0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00400048	0x10010004
0x7fffffc0	0x00000001	0x00400048	0x10010004	0x00000002	0x00400048	0x10010004	0x00000003	0x00400048
0x7fffffe0	0x10010004	0x00000004	0x00400048	0x10010004	0x00000005	0x00400014	0x10010004	0x00000000
0x7fffff00	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
0x7ffff120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x7ffff140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Registers

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000f
\$v1	3	0x00000000
\$a0	4	0x00000005
\$a1	5	0x00000000
\$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	0x0000000f
\$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
\$sp	28	0x10008000
\$fp	29	0x7ffff000
\$gp	30	0x00000000
\$ra	31	0x00400014
pc		0x00400014
hi		0x00000000
lo		0x00000000

Mars Messages

Run I/O

Clear

-- program is finished running (dropped off bottom) --



After result is stored into the memory location of F. check that the memory location indicated by \$S0 has the correct results 15 (i.e. f in Hex).

C:\Users\Brinda\GoogleDrive\_CUA\CSC390\MIPS\_test\HW#5\_Q1\_sp2018 - MARS 4.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

**Text Segment**

Program Arguments:

Byte	Address	Code	Basic	Source
	0x00400014	0x5fb00008	lw \$16,0x00000008(\$29)	14: lw \$s0, 8(\$sp) #Restore the address of F
	0x00400018	0xae020000	sw \$2,0x00000000(\$16)	15: sw \$r0, 0(\$s0) # store the return value (i.e result) from the function into F
	0x0040001c	0x00000000	j 0x00400060	16: j halt
	0x00400020	0x23bdffff	addi \$29,\$29,0xffff...	21: addi \$sp, \$sp, -12 # reserve space in the stack to store \$s0, \$a0 and \$ra
	0x00400024	0xafb00008	sw \$16,0x00000008(\$29)	23: sw \$s0, 8(\$sp) # store the
	0x00400028	0xafbf0004	sw \$31,0x00000004(\$29)	24: sw \$ra, 4(\$sp) # save return address
	0x0040002c	0xafaf0000	sw \$4,0x00000000(\$29)	25: sw \$a0, 0(\$sp) # save argument
	0x00400030	0x28880001	slti \$8,\$4,0x00000001	26: slti \$t0, \$a0, 1 # test for n < 1
	0x00400034	0x11000002	beq \$8,\$0,0x00000002	27: beq \$t0, \$zero, L1
	0x00400038	0x20100000	addi \$16,\$0,0x00000000	28: addi \$s0, \$zero, 0 # if so, result is 0
	0x0040003c	0x00e00008	jr \$31	29: jr \$ra # return to line 32
	0x00400040	0x2084ffff	addi \$4,\$4,0xffffffff	30: li: addi \$a0, \$a0, -1 # else decrement n
	0x00400044	0x00c10008	jal 0x00400020	31: jal sum # recursive call
	0x00400048	0x23bd000c	addi \$29,\$29,0x0000...	32: addi \$sp, \$sp, 12 # Increment stack by 12
	0x0040004c	0x5fbf0004	lw \$31,0x00000004(\$29)	33: lw \$ra, 4(\$sp) # Restore \$ra

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000005	0x0000000f	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
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

**Registers**

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000f
\$v1	3	0x00000000
\$a0	4	0x00000005
\$a1	5	0x00000000
\$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	0x10010004
\$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	0x7ffff000
\$fp	30	0x00000000
\$ra	31	0x00400014
pc		0x0040001c
hi		0x00000000
lo		0x00000000

**Mars Messages** Run I/O

Clear -- program is finished running (dropped off bottom) --

## 2.26

**2.26.1** 20

**2.26.2** `i = 10;`  
    `do {`  
        `B += 2;`  
        `i = i - 1;`  
    `} while ( i > 0)`

**2.26.3**  $5*N+2$