

**Computer Architecture**  
**Spring 2019**  
**Homework No. 3**

**1.**

Let f, g, and h are stored at \$s0, \$s1, \$s2

```
addi $t0, $s2, -5
add $s0, $s1, $t0
```

**2.**

```
sub $t0, $s3, $s4
sll $t0, $t0, 2
add $t1, $s6, $t0
lw $t2, 0($t1)
sw $t2, 32($s7)
```

**3.**

- Memory address where Fact procedure is located

Fact:

```
200    addi    $sp, $sp, -8
204    sw      $ra, 4($sp)
208    sw      $a0, 0($sp)
212    slti    $t0, $a0, 1
216    beq     $t0, $zero, L1
220    addi    $v0, $zero, 1
224    addi    $sp, $sp, 8
228    jr      $ra
L1:
232    addi    $a0, $a0, -1
236    jal     fact
240    lw      $a0, 0($sp)
244    lw      $ra, 4($sp)
248    addi    $sp, $sp, 8
252    mul     $v0, $a0, $v0
256    jr      $ra
```

- Code execution

300	jal	Fact	# \$ra = 304, \$pc = 200, \$sp = 500, \$a0 = 3
200	addi	\$sp, \$sp, -8	# \$sp = 492
204	sw	\$ra, 4(\$sp)	# save M[496] = 304
208	sw	\$a0, 0(\$sp)	# save M[492] = 3
212	slti	\$t0, \$a0, 1	# since 3 >= 1, \$t0 = 0
216	beq	\$t0, \$zero, L1	# since \$ t0 = 0, go to L1 : \$pc = 232
232	addi	\$a0, \$a0, -1	# \$a0 = 2

236	jal	Fact	# go to factorial \$ra = 240, \$pc = 200
200	addi	\$sp, \$sp, -8	# sp = 484
204	sw	\$ra, 4(\$sp)	# save M[488] = 240
208	sw	\$a0, 0(\$sp)	# save M[484] = 2
212	slti	\$t0, \$a0, 1	# since 2 >= 1, \$t0 = 0
216	beq	\$t0, \$zero, L1	# since \$ t0 = 0, go to L1 : \$pc = 232
232	addi	\$a0, \$a0, -1	# \$a0 = 1
236	jal	fact	# \$ra = 240, \$pc = 200
200	addi	\$sp, \$sp, -8	# sp = 476
204	sw	\$ra, 4(\$sp)	# save M[480] = 240
208	sw	\$a0, 0(\$sp)	# save M[476] = 1
212	slti	\$t0, \$a0, 1	# since 1 >= 1, \$t0 = 0
216	beq	\$t0, \$zero, L1	# since \$ t0 = 0, go to L1 : \$pc = 232
232	addi	\$a0, \$a0, -1	# \$a0 = 0
236	jal	fact	# go to factorial \$ra = 240, \$pc = 200
200	addi	\$sp, \$sp, -8	# sp = 468
204	sw	\$ra, 4(\$sp)	# save M[472] = 240
208	sw	\$a0, 0(\$sp)	# save M[468] = 0
212	slti	\$t0, \$a0, 1	# since 0 < 1, \$t0 = 1
216	beq	\$t0, \$zero, L1	# since \$ t0 = 1, does not go to L1
220	addi	\$v0, \$zero, 1	# \$v0 = 1
224	addi	\$sp, \$sp, 8	# sp = 476
228	jr	\$ra	# \$pc = 240
240	lw	\$a0, 0(\$sp)	# \$a0 = M[476] = 1
244	lw	\$ra, 4(\$sp)	# \$ra = M[480] = 240
248	addi	\$sp, \$sp, 8	# \$sp = 484
252	mul	\$v0, \$a0, \$v0	# \$v0 = 1 * 1 = 1
256	jr	\$ra	# \$pc = 240
240	lw	\$a0, 0(\$sp)	# \$a0 = M[484] = 2
244	lw	\$ra, 4(\$sp)	# \$ra = M[488] = 240
248	addi	\$sp, \$sp, 8	# \$sp = 492
252	mul	\$v0, \$a0, \$v0	# \$v0 = 2 * 1 = 2
256	jr	\$ra	# go to 240
240	lw	\$a0, 0(\$sp)	# \$a0 = M[492] = 3
244	lw	\$ra, 4(\$sp)	# \$ra = M[496] = 304
248	addi	\$sp, \$sp, 8	# \$sp = 500
252	mul	\$v0, \$a0, \$v0	# \$v0 = 3 * 2 = 6
256	jr	\$ra	# go to 304

## - Assembly code

```

        .text

fact:
    addi $sp, $sp, -8
    sw   $ra, 4($sp)
    sw   $a0, 0($sp)

    slti $t0, $a0, 1
    beq  $t0, $zero, L1

    addi $v0, $zero, 1
    addi $sp, $sp, 8
    jr   $ra

L1:
    addi $a0, $a0, -1
    jal  fact

    lw   $a0, 0($sp)
    lw   $ra, 4($sp)
    addi $sp, $sp, 8

    mul  $v0, $a0, $v0

    jr   $ra

main:
    addi $a0, $zero, 3          # Set n = 3
    jal  fact

    add $a0, $v0, $zero         # Store the result of fact at a0
    li  $v0, 1                  # Option : print integer
    syscall

```

## - Result

The screenshot displays the QSPin simulator interface. The main window shows the assembly code from the previous block, with the 'Text' segment selected. The 'FP Regs' and 'Int Regs [16]' tabs are visible. The 'Int Regs' tab shows the following register values:

Register	Value
R0	0
R1	0
R2	1
R3	0
R4	4
R5	7ffff3ac
R6	7ffff3b4
R7	0
R8	1
R9	0
R10	0
R11	0
R12	0
R13	0
R14	0
R15	0
R16	0
R17	0
R18	0
R19	0
R20	0
R21	0
R22	0
R23	0
R24	0
R25	0
R26	0
R27	0

The 'Text' segment shows the assembly code with comments. The 'User Text Segment' is visible, showing the code for the 'fact' and 'main' functions. The 'Kernel Text Segment' is also visible, showing the code for the 'main' function. The 'Console' window on the right is empty.

4.

- code

```

.data
x: .ascii "as"
y: .ascii "Computer A"

.text

strcpy:
    addi $sp, $sp, -4
    sw    $s0, 0($sp)
    add   $s0, $zero, $zero

L1: add   $t1, $s0, $a1
    lbu   $t2, 0($t1)
    add   $t3, $s0, $a0
    sb    $t2, 0($t3)
    beq   $t2, $zero, L2
    addi  $s0, $s0, 1
    j     L1

L2: lw    $s0, 0($sp)
    addi  $sp, $sp, 4
    jr    $ra

main:
    la    $a0, x
    la    $a1, y

    jal   strcpy

    li    $v0, 4                # Option : print string
    syscall                       # Print x

```

- Result

The screenshot shows the QtSpim MIPS simulator interface. The 'Registers' panel on the left lists registers from \$0 to \$31. Registers \$a0 and \$a1 are highlighted in red, indicating they are the current selection. The 'Text' panel in the center shows the assembly code with comments. The 'Console' panel on the right shows the output 'Computer A'.