San Jose State University
Department of Computer Engineering

CMPE 200 Lab Report

Assignment 4 Report

Title: Array Processing, Stack and Recursive Procedure

Semester: Fall 2023 **Date:** 10/03/2023

by

Name: Sai Kashyap Kurella **SID:** 016018925

Source Code

```
# $a0 = array base address
# $a1 = n
# $s0 = n!
Main:
 addi $a0, $0, 0x100 # array base address = 0x100
 addi $a1, $0, 0 # i = 0
 addi $t0, $0, 3
 addi $t1, $0, 50 # $t1 = 50
CreateArray_Loop:
 slt $t2, $a1, $t1 # i < 50?
 beq $t2, $0, Exit_Loop # if not then exit loop
 sll $t2, $a1, 2 # $t2 = i * 4 (byte offset)
 add $t2, $t2, $a0 # address of array[i]
 mult $a1, $t0
 mflo $t3 # $t3 = i * 3
 sw $t3, 0($t2) # save array[i]
 addi $a1, $a1, 1 # i = i + 1
 j CreateArray_Loop
Exit_Loop:
 # your code goes in here...
 # arithmetic calculation
 lw $s1,356($t2)
 lw $s2,376($t2)
 add $s2,$s2,$s1
 addi $s3,$zero,30
 div $s2,$s3
 mflo $s4
 move $a1,$s4
 sw $a1, 0($0)
 # ...
 # factorial computation
 jal factorial # call procedure
 add $s0, $v0, $0 # return value
 sw $s0,16($0)
 j exit
factorial:
 addi $sp, $sp, -8
 sw $ra, 4($sp)
 sw $a1, 0($sp)
slti $t0,$a1,2
beq $t0,$zero,L1
addi $v0,$zero,1
addi $sp,$sp,8
jr $ra
L1: addi $a1,$a1,-1
 jal factorial
lw $a1, 0($sp)
lw $ra, 4($sp)
addi $sp, $sp, 8
mul $v0,$a1,$v0
jr $ra
exit:
```

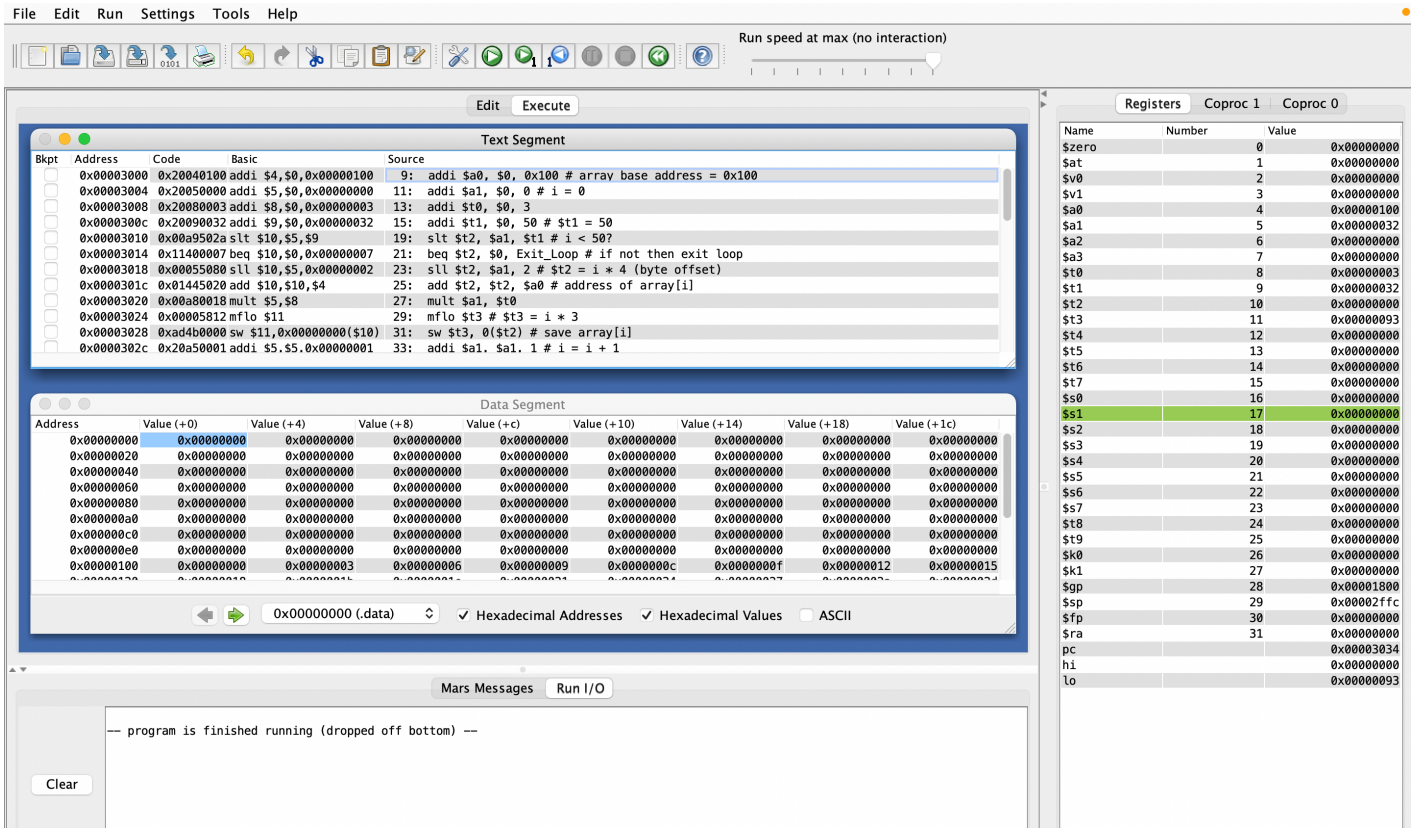
CMPE 200 Assignment 4 Test Log

Programmer's Names: Sai Kashyap Kurella **Date: 10/03/2023**

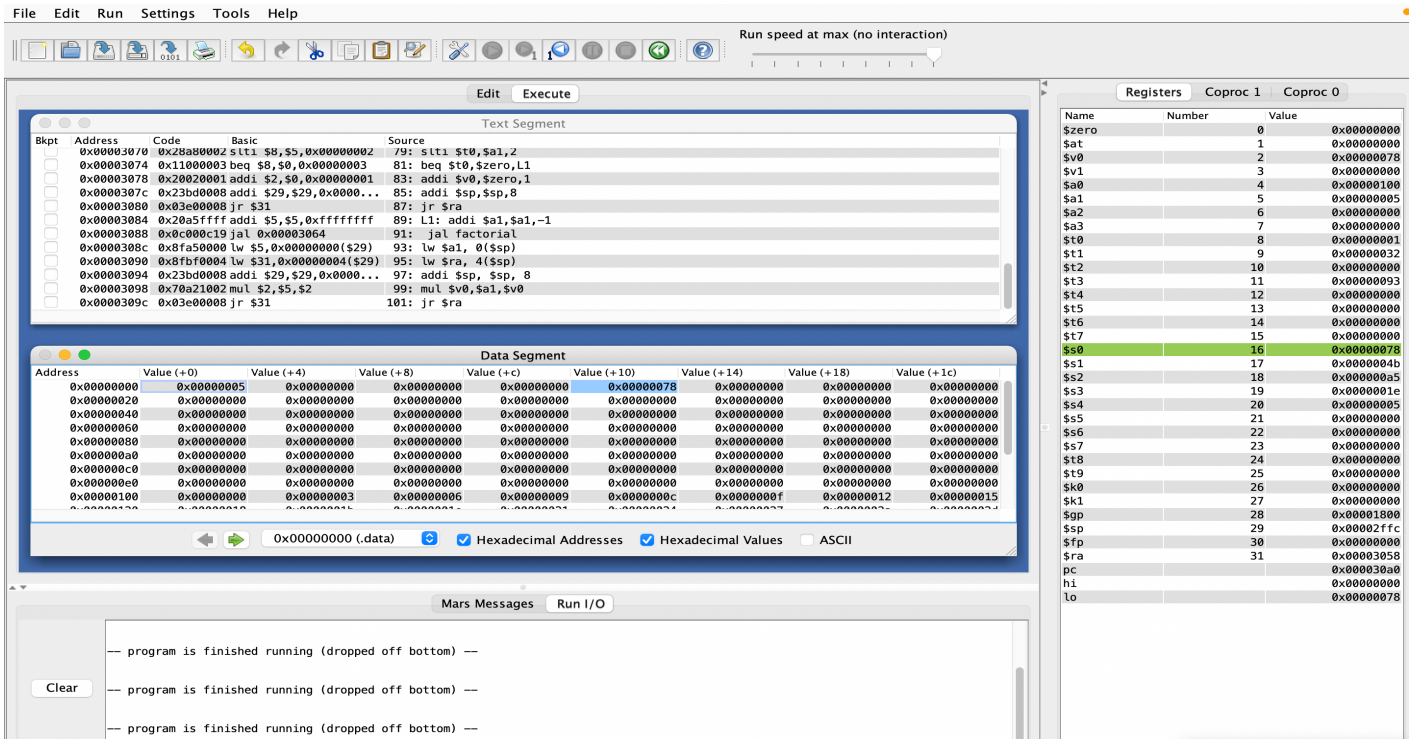
Record the observed contents of registers and data memory after each instruction is executed.

Addr	MIPS Instruction	Machine Code	Registers				Memory Content	
			$a1	$sp	$ra	$v0	[0x00]	[0x10]
3034	lw $s1,356($t2)	0x8d510164	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3038	lw $s2,376($t2)	0x8d520178	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
303c	add $s2,$s2,$s1	0x02519020	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3040	addi $s3,$zero,30	0x20130013	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3044	div $s2,$s3	0x0253001a	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3048	mflo $s4	0x0000a012	0x00000032	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
304c	move $a1,$s4	0x00142821	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3050	sw $a1, 0($0)	0xac050000	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3054	Jal factorial	0x0c000c19	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3058	add $s0, $v0, $0	0x00408020	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
305c	sw $s0,16($0)	0xac100010	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3060	J exit	0x08000c28	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3064	addi $sp, $sp, -8	0x23bdfff8	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3068	sw $ra, 4($sp)	0xafbf0004	0x00000005	0x00002ff4	0x0000000	0x0000000	0x0000000	0x0000000
306c	sw $a1, 0($sp)	0xafa50000	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3070	slti $t0,$a1,2	0x28a80002	0x00000005	0x00002ffc	0x0000000	0x0000000	0x0000000	0x0000000
3074	beq $t0,$zero,L1	0x10000003	0x00000005	0x00002ffc	0x0000000	0x0000000	0x00000005	0x0000000
3078	addi $v0,$zero,1	0x20020001	0x00000005	0x00002ffc	0x0000000	0x00000078	0x00000005	0x00000078
307c	addi $sp,$sp,8	0x23bd0008	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3080	jr $ra	0x03e00008	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3084	L1: addi $a1,$a1,-1	0x20a5ffff	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3088	jal factorial	0x0c00019	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
308c	lw $a1, 0($sp)	0x8fa5000	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3090	lw $ra, 4($sp)	0x8fbf0004	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3094	addi $sp, $sp, 8	0x23bd0008	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
3098	mul $v0,$a1,$v0	0x70a21002	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078
309c	jr $ra	0x03e00008	0x00000005	0x00002ffc	0x00003058	0x00000078	0x00000005	0x00000078

Snapshot before execution:



Snapshot after execution:



Conclusion:

In conclusion, this assignment has provided a valuable opportunity to delve into the world of MIPS assembly programming and gain a deeper understanding of various fundamental concepts within it. Building a 50-entry array at base address 0x100 and performing arithmetic calculations on it has allowed us to explore the implementation of arrays, stacks, procedures, and recursive procedures in MIPS.