CMPE140 Lab 3 Task 2 Test Log
Algorithm 1

Programmer's Name: Nick Schiffer & Salvatore Nicosia

Checked by: _____, Date: 02/27/19

Adr	MIPS Instruction	Machine Code	Registers				
			$a0	$a1	$s0	$s1	$s2
00	# addiu $a0, $zero, 32768 ($a0 = 32768)	0x24048000	0x80000	0	0	0	0
04	# addiu $a1, $zero, 169 ($a1 = 169)	0x240500A9	0x80000	0xA9	0	0	0
08	# addiu $s0, $zero, 1974 ($s0 = 1974)	0x241007B6	0x80000	0xA9	0x7B6	0	0
0C	# mult $a0, $a0 ($hi = High($a0 * $a0); $lo = Low($a0 * $a0))	0x00840018	0x80000	0xA9	0x7B6	0	0
10	# mflo $s1 ($s1 = $lo)	0x00008812	0x80000	0xA9	0x7B6	0x40000000	0
14	# sw $s1, 32($zero) (mem[$zero + 32] = $s1)	0xAC110020	0x80000	0xA9	0x7B6	0x40000000	0
18	# mult $s1, $a1 ($hi = High($s1 * $a1); $lo = Low($s1 * $a1))	0x02250018	0x80000	0xA9	0x7B6	0x40000000	0
1C	# mfhi $s2 ($s2 = $hi)	0x00009010	0x80000	0xA9	0x7B6	0x40000000	0x2A
20	# mflo $s3 ($s3 = $lo)	0x00009812	0x80000	0xA9	0x7B6	0x40000000	0x2A
24	# srl $s3, $s3, 16 ($s3 = $s3 >> 16)	0x00139C02	0x80000	0xA9	0x7B6	0x40000000	0x2A
28	# sll $s2, $s2, 16 ($s2 = $s2 << 16)	0x00129400	0x80000	0xA9	0x7B6	0x40000000	0x2A0000
2C	# or $s2, $s2, $s3 ($s2 = $s2 | $s3)	0x02539025	0x80000	0xA9	0x7B6	0x40000000	0x2A4000
30	# sw $s2, 36($zero) (mem[$zero + 36] = $s2)	0xAC120024	0x80000	0xA9	0x7B6	0x40000000	0x2A4000
34	# divu $s2, $s0 ($Hi = $s2 div $s0; $Lo = $s2 mod $s0)	0x0250001B	0x80000	0xA9	0x7B6	0x40000000	0x2A4000
38	# mflo $t0 ($t0 = $lo)	0x00004012	0x80000	0xA9	0x7B6	0x40000000	0x2A4000
3C	# addu $s0, $t0, $s0 ($s0 = $t0 + $s0)	0x01108021	0x80000	0xA9	0xD30	0x40000000	0x2A4000
40	# srl $s0, $s0, 1 ($s0 = $s0 >> 1)	0x00108042	0x80000	0xA9	0x698	0x40000000	0x2A4000
44	# sw $s0, 44($zero) (mem[$zero + 44] = $s0)	0xAC10002C	0x80000	0xA9	0x698	0x40000000	0x2A4000
48	# slti #RD, $s0, 1665 (if ($s0 < 1665) #RD = 1 else #RD = 0)	0x2A0B0681	0x80000	0xA9	0x680	0x40000000	0x2A4000
4C	# bne $t3, $zero, 5 (if ($t3 != $zero) goto 5)	0x140B0005	0x80000	0xA9	0x698	0x40000000	0x2A4000
50	# divu $s2, $s0 ($Hi = $s2 div $s0; $Lo = $s2 mod $s0)	0x0250001B	0x80000	0xA9	0x698	0x40000000	0x2A4000
54	# mflo $t0 ($t0 = $lo)	0x00004012	0x80000	0xA9	0x698	0x40000000	0x2A4000
58	# addu $s0, $t0, $s0 ($s0 = $t0 + $s0)	0x01108021	0x80000	0xA9	0xD00	0x40000000	0x2A4000
5C	# srl $s0, $s0, 1 ($s0 = $s0 >> 1)	0x00108042	0x80000	0xA9	0x680	0x40000000	0x2A4000
60	# j 0x0012 (jump to addr 0x0048)	0x08000012	0x80000	0xA9	0x680	0x40000000	0x2A4000
64	# sll $s0, $s0, 8 ($s0 = $s0 << 8)	0x00108200	0x80000	0xA9	0x68000	0x40000000	0x2A4000
68	# sw $s0, 48($zero) (mem[$zero + 48] = $s0)	0xAC100030	0x80000	0xA9	0x68000	0x40000000	0x2A4000
6C	# j 0x0000 (jump to addr 0x0000)	0x08000000	0x80000	0xA9	0x68000	0x40000000	0x2A4000

Memory Contents			
Word @ 0x20	**Word @ 0x24**	**Word @ 0x2C**	**Word @ 0x30**
0x40000000	0x00000024	0x00000698	0x00068000