

1. Median address: 0x00400190

2. lui \$1,0x00001001

2. ori \$16,\$1,0x00000008

3. 0x3c011001

3. 0x34300008

Part 2.1

1. "sum: "

2. 73 75 6d 3a

3. 0x100100a0

4. 0x73006576 0x203a6d75

Something new I learned was that when strings are stored in memory, they are actually stored backwards, and can be in the same word as other words in memory. This could be useful in the future to help identify where the strings are stored if I need to load it from memory. One question I had was why the strings were stored backwards. To find the info for this exercise, I had to use MARs breakpoint function and line numbers to accurately determine what a variables address was in memory. Looking at the instructions, constants are stored in the register, because ORI stores the result of a register and an immediate value there.

<input type="checkbox"/>	0x004001d8	0x3c011001	lui \$1,0x00001001	71:	la	\$a0, two
<input type="checkbox"/>	0x004001dc	0x34240018	ori \$4,\$1,0x00000018			

	0x73006576	0x203a6d75
--	------------	------------

Bkpt	Address	Code	Basic	
<input type="checkbox"/>	0x004001bc	0x3c011001	lui \$1,0x00001001	64: la \$a0, one
<input type="checkbox"/>	0x004001c0	0x34240014	ori \$4,\$1,0x00000014	
<input type="checkbox"/>	0x004001c4	0x8c840000	lw \$4,0x00000000(\$4)	65: lw \$a0, 0(\$a0)
<input type="checkbox"/>	0x004001c8	0x0000000c	syscall	66: syscall
<input type="checkbox"/>	0x004001cc	0x081000b4	j 0x004002d0	67: j MEDIAN_END
<input type="checkbox"/>	0x004001d0	0x16530006	bne \$18,\$19,0x00000006	69: bne \$s2, \$s3, JUMP_2 # else if (two == three)
<input type="checkbox"/>	0x004001d4	0x20020001	addi \$2,\$0,0x00000001	70: addi \$v0, \$zero, 1
<input type="checkbox"/>	0x004001d8	0x3c011001	lui \$1,0x00001001	71: la \$a0, two
<input type="checkbox"/>	0x004001dc	0x34240018	ori \$4,\$1,0x00000018	
<input type="checkbox"/>	0x004001e0	0x8c840000	lw \$4,0x00000000(\$4)	72: lw \$a0, 0(\$a0)
<input type="checkbox"/>	0x004001e4	0x0000000c	syscall	73: syscall
<input type="checkbox"/>	0x004001e8	0x081000b4	j 0x004002d0	74: j MEDIAN_END
<input type="checkbox"/>	0x004001ec	0x0232482a	slt \$9,\$17,\$18	76: slt \$t1,\$s1,\$s2 #t1 = s1<s2
<input type="checkbox"/>	0x004001f0	0x0233502a	slt \$10,\$17,\$19	77: slt \$t2,\$s1,\$s3 #t2 = s1<s3
<input type="checkbox"/>	0x004001f4	0x0253582a	slt \$11,\$18,\$19	78: slt \$t3,\$s2,\$s3 #t3 = s2<s3
<input type="checkbox"/>	0x004001f8	0x20020004	addi \$2,\$0,0x00000004	79: addi \$v0, \$zero, 4
<input type="checkbox"/>	0x004001fc	0x3c011001	lui \$1,0x00001001	80: la \$a0, COMPARISONS_STR
<input type="checkbox"/>	0x00400200	0x34240068	ori \$4,\$1,0x00000068	
<input type="checkbox"/>	0x00400204	0x0000000c	syscall	81: syscall
<input type="checkbox"/>	0x00400208	0x20020001	addi \$2,\$0,0x00000001	82: addi \$v0, \$zero, 1
<input type="checkbox"/>	0x0040020c	0x01202020	add \$4,\$9,\$0	83: add \$a0, \$t1, \$zero
<input type="checkbox"/>	0x00400210	0x0000000c	syscall	84: syscall
<input type="checkbox"/>	0x00400214	0x20020004	addi \$2,\$0,0x00000004	85: addi \$v0, \$zero, 4
<input type="checkbox"/>	0x00400218	0x3c011001	lui \$1,0x00001001	86: la \$a0, SPACE_STR
<input type="checkbox"/>	0x0040021c	0x34240076	ori \$4,\$1,0x00000076	
<input type="checkbox"/>	0x00400220	0x0000000c	syscall	87: syscall
<input type="checkbox"/>	0x00400224	0x20020001	addi \$2,\$0,0x00000001	88: addi \$v0, \$zero, 1
<input type="checkbox"/>	0x00400228	0x01402020	add \$4,\$10,\$0	89: add \$a0, \$t2, \$zero
<input type="checkbox"/>	0x0040022c	0x0000000c	syscall	90: syscall
<input type="checkbox"/>	0x00400230	0x20020004	addi \$2,\$0,0x00000004	91: addi \$v0, \$zero, 4
<input checked="" type="checkbox"/>	0x00400234	0x3c011001	lui \$1,0x00001001	92: la \$a0, SPACE_STR
<input type="checkbox"/>	0x00400238	0x34240076	ori \$4,\$1,0x00000076	
<input type="checkbox"/>	0x0040023c	0x0000000c	syscall	93: syscall
<input type="checkbox"/>	0x00400240	0x20020001	addi \$2,\$0,0x00000001	94: addi \$v0, \$zero, 1
<input type="checkbox"/>	0x00400244	0x01602020	add \$4,\$11,\$0	95: add \$a0, \$t3, \$zero
<input type="checkbox"/>	0x00400248	0x0000000c	syscall	96: syscall
<input type="checkbox"/>	0x0040024c	0x2002000b	addi \$2,\$0,0x0000000b	98: addi \$v0, \$zero, 11
<input type="checkbox"/>	0x00400250	0x2004000a	addi \$4,\$0,0x0000000a	99: addi \$a0, \$zero, 0xa