This problem set has 10 questions, for a total of 100 points. Answer the questions below and mark your answers in the spaces provided. For all questions, please provide details on how your answer was calculated.

Your Name: Yevry Crulvu Visiano

- 1. For the following C statements, write the corresponding MIPS assembly code. Assume that variables i, j are assigned to registers \$50, \$51, respectively. Assume that the base address of the arrays A, B are in registers \$s2, \$s3, respectively.
 - (a) [5 points] B[5] = A[i+j];

(b) [5 points] B[9] = A[i-j];

Sub \$ to, \$50, \$6, # index -7 i-J

SLL \$ to, \$to, \$2 # Mult by 4

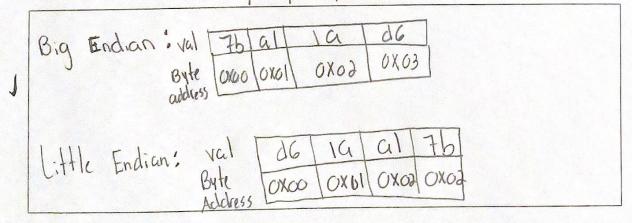
Cidd I to, \$to, \$52 # A[1-J] Mem addy

LW \$ t1, O(\$to) # value A[1-J]

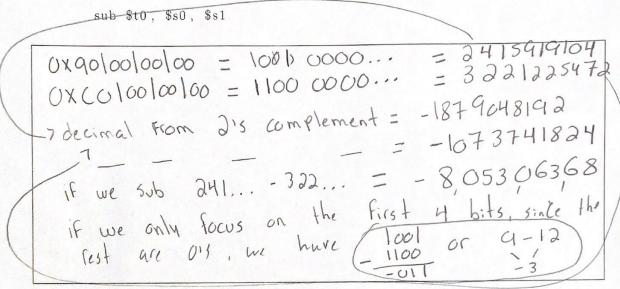
SW \$ t1, 36(\$53) # store val OF B[9]

2. [10 points] For the MIPS assembly instructions below, what is the corresponding C statement? Assume that variables f, g are assigned to registers \$s0., \$s1, respectively. Assume that the base address of arrays A, B are in registers \$s2, \$s3, respectively.

3. [5 points] Show how the value 0x7ba11ad6 would be arranged in memory for a little-endian and for a big-endian machine. Assume that the data are stored starting at address 0 and that the word size is 4 bytes.



- Assume that registers \$50, \$51 hold the values 0x90000000 and 0xC0000000, respectively.
 - (a) [5 points] What is the value of \$t0 for the following assembly code? Is the result in \$t0 the desired result, or has there been overflow?



(b) [5 points] What is the value of \$t0 for the following assembly code? Is the result in \$t0 the desired result, or has there been overflow?

add \$t0, \$s0, \$s1 add \$t0, \$t0, \$s0

As shown above, if we add 2415919104

+ 3221225472

now this would be an 5,637,144576

overflow because an integer only
holds HB and if we account for
holds HB and if we account for
signed two's complement too, then it will
signed two's complement overflow

5. [10 points] Provide the type, assembly language instruction, and hexadecimal representation of the instruction described by the following MIPS fields: op=0, rs=2, rt=4, rd=1, shamt=0, funct=32.

6. [10 points] Provide the type, assembly language instruction, and hexadecimal representation of the instruction described by the following MIPS fields: op=0x21, rs=0, rt=1, const=0x6.

Type = MIPS I-formed of=0x21 = locably

Assembly language = 1h sat, 6 (3zero) (5=0 = ochodo
t = ochodosta

Const = ochodosta

C

- 7. Assume the following register contents for \$t0, \$t1 respectively: 0xAAAAABEEF and 0x12341234
 - (a) [5 points] What is the value of \$t2 for the following sequence of instructions?

sll \$t2, \$t0, 16 \ Byte = 2 hex valves or \$t2, \$t2, \$t1 16 shifts = 2 Bytes = 4 hox valves

OAAAABEEF 22 4 = 100 x BEEF0000 color turn both to bits then perform on bits.

Or obstwise 2 speciation on bits.

(b) [5 points] What is the value of \$t2 for the following sequence of instructions?

sll \$t2, \$t0, 8 andi \$t2, \$t2, -1

Afterst

OXAAAABEEF -> AABEEFOO where adding -1 to)

the = OXAABEEFOO (c) [5 points] What is the value of \$t2 for the following sequence of instructions?

srl \$t2, \$t0, 16
andi \$t2, \$t2, 0x0000FFEF

After Shifting right OCCOAAAA And OCCOAAAA OCCOAAAA	
Ignore all o's because it will always give a will always give always g	
(0x0000AAAA)	

8. [10 points] For the following C statement, write a minimal sequence of MIPS assembly instructions that does the identical operation. Assume variable A is in \$t0 and \$s0 is the base address of array C.

$$A = C[8] << 8;$$

lu \$ to, 32 (\$50) 511 \$ to, \$to, 8	the load val at CE8] H shift left

9. [10 points] Translate the following loop into C. Assume that the integer a is held in register \$11, \$50 holds the integer ans, and \$52 holds the base address of the integer array data.

10. For the following code, assume that register \$t1 contains the address 0x20000100 and the data at that address is 0xa1b2c3d4.

Big Endian: Oxa1b2 C3d4

lbu \$t0, @(\$t1)sw \$t0, 0(\$t1) Little Endian: OX d4C3b2al

(a) [5 points] What value is stored in 0x20000104 on a little-endian machine?

UX000000d4

(b) [5 points] What value is stored in 0x20000104 on a big-endian machine?

0x41000000