

TPS Activity #1:

1. Downloaded MIPS reference sheet.
2. Done.
3. Observed how each line of code was translated into both a machine and basic code equivalent.
4. Each machine instruction contains 32 bits.
5. There are 3 types of machine instructions: R, I, and J..
 - R: add, slt
 - I: sw, addi
 - J: j, jal
6. a. The instruction on line 7 of proc1.s is an arithmetic instruction. This instruction has 4 fields, one for registers to add and store sum to, one for opcode, and another for the immediate we want to add.
b. The opcode for addi is 0x8. Registers \$rs and \$rt are the registers being added to the intermediate and storing the sum, respectively. In line 7, \$rs is \$zero and has a hexadecimal value of 0x00000000. In line 7, the \$rt is \$s0 and has a value of 0x10. The value of the intermediate is 0x00000019.
c. 0x20100019. Binary format: 00100000000100000000000000011001.
7. a. 0x0230402a is the machine code at the address 0x00400010. The binary equivalent is 0000 0010 0011 0000 0100 0000 0010 1010.
b. This is instruction type R because the opcode is 0. There are 6 fields: opcode, \$rs, \$rt, \$rd, shamt, and function.
c. \$rs = 10001 = 17 = 0x11
\$rt = 10000 = 16 = 0x1
\$rd = 01000 = 8 = 0x8
\$shamt = 00000 = 0 = 0x0
\$function = 10 1010 = 0x2a
d. This is slt, an R instruction. I can tell by looking at the opcode and rs = \$t0, rt = \$s1, rd = \$s0.
e. The final mips instruction is li \$v0, 10. This is the same as the instruction in the Source Column in the Text Segment window.
8. a. BNE is an instruction type of I.
b. Rs = \$t0 = 0x8, rt = \$zero = 0x0, opcode = 0x5.
c. LESS is the branch's name and its address is 0x00000001.
d. We will be adding the branch address to the PC if BNE branches, therefore we will need to add an intermediate value.
e. Intermediate field can be found by looking up its value in the test segment tab.
f. 001 0101 0000 0000 0000 0001 and 0x15001
9. a. There are two fields in J type instruction.
b. The instruction has an opcode of 2.

- c. It jumps to GREQ and it has an address of 0x00400030.
- d. There are 26 bits allocated for address in J type instruction. This is done to better accommodate a long binary value. 0000 0000 0100 0000 0000 0000 0011 0000.
- e. In hex: 0x0810000c and in binary: 0000 1000 0001 0000 0000 0000 0000 1100. Yes it is the same as what's in the text segment window.