1) [5 points] What MIPS instruction is this? 0010 0011 0010 0010 0010 0001 0000 1000

001000 11001 00010 0010000100001000
Op Rs Rt Imm

addi $r2, $r25, 8456

2) [5] What is the binary representation of this instruction? lw $r2, -8($r12)

100011 01100 00010 1111111111111000
Op = lw Rs = 12 Rt = 2 Immed = -8 (2's complement)

3)

a. -O0: 57 total instructions
   -O3: 32 total instructions
b. -O0: lw: 7
   sw: 3
   -O3: lw: 2
   sw: 2
c. -O3 optimized code uses more registers
d. From the above analysis, the general strategies I suspect the optimizer takes to improve performance are that it reduces the number of instructions in the original code by minimizing the number of lw and sw instructions to the amount necessary to prevent accessing memory, again and again, using the stack. It also increases the number of registers used to compensate for saving and loading less data. As registers are faster to access and utilize than main memory, this optimizes the code.