Homework 2
Liane Yanglian
Net ID: xy48

1. sub $s0,$t9,$s2

   The encoding for sub is 0000 00 ss ssst tttt dddd d000 0010 0010, which fits the instruction given, so it's a sub. For the middle 15 numbers11001 denotes $t9, 10010 denotes $s2, 10000 denotes $s0, therefore sub $s0,$t9,$s2. This is a R-type instruction, with semantics R[d] = R[s]+R[t].

2. 1010 1100 1000 0111 0000 0000 0000 1100

   The first 6 numbers correspond to sw, next 5 $r4, next 5 $r7, last 16 12. The format for sw's encoding is 1010 11ss ssst ttt iiii iiii iiii iiii,which fits the instruction given. This is an I-type instruction with 12 as the immediate, with meaning, sw R[t],immediate(R[s]).

3. a)

   The code without optimization has 66 instructions while the code with optimization has 33 instructions.

   b)

   The code without optimization has 18 memory accesses while the code with optimization has 6 memory accesses.

   c)

   The code with optimization uses more registers (14) than the code without optimization (8).

   d)

   Some of the general strategies the optimizer uses to improve performance include using fewer registers. Since getting a value from a register is much faster than getting a value from memory, the optimizer uses more registers so that it can have faster access to values. The un-optimized code has fewer registers, which means that it has to store more values in memory whereas the optimized code can do more of the calculations in

the registers since it has more of them. As a result, the optimized code does not need as many memory accesses (loads and stores) since it stores more values directly in the registers. It also has much fewer instructions than the un-optimized code as a consequence, which reduces its run time and increases it efficiency.