For all of the questions in this homework, consider the pipelined CPU discussed in class, which is similar to the CPU in lab #4. That is, it has branch and load delay slots, forwarding logic from class (NOT the book), and accelerated branches.

1.) For the following code, explain what is happening in each stage of the pipelined processor during cycle 5.

   ```
   addi $t0, $0, 102
   lw $t1, 10($t0)
   sw $t0, 10($t0)
   beq $t0, $t1, FOO
   sub $t2, $t1, $t0
   ```

2.) Our processor has a load delay slot, which means code cannot use the rt register of a load in the instruction right after a load. What happens if we ignore this restriction? Specifically, assume the following code is given to the CPU. Assume MEM[20] = -12. What value will end up in register $9? Think carefully – this one is tricky!

   ```
   addi $8, $0, 16
   lw $8, 4($8)
   add $9, $8, $8
   ```

3.) The following code contains a “read after write” data hazard that is resolved by forwarding:

   ```
   add $2, $3, $4
   add $5, $2, $6
   ```

   Consider the following code where a memory read occurs after a memory write:

   ```
   sw $7, 100($2)
   lw $8, 100($2)
   ```

   Does the code work correctly? Why/why not? Will the forwarding unit need to be altered to handle this code?

4.) Consider a processor with the follow delays in each of the individual units:

<table>
<thead>
<tr>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>MEM</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>200ps</td>
<td>150ps</td>
<td>120ps</td>
<td>190ps</td>
<td>140ps</td>
</tr>
</tbody>
</table>

   a.) What is the clock cycle time in a pipelined and non-pipelined processor?

   b.) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split, and what is the new clock cycle time of the processor?
5.) For the following code, explain what the forwarding unit is doing during the fifth cycle of execution. If any comparisons are being made, mention them. Remember, use the forwarding unit from class, not the book.

LOOP:

lw $1, 40($6)
add $5, $5, $8
add $6, $6, $8
sw $1, 20($5)
beq $1, $0, LOOP