Problem 1:

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Freq. of Inst.</th>
<th>CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branches</td>
<td>20%</td>
<td>1</td>
</tr>
<tr>
<td>Loads</td>
<td>20%</td>
<td>2</td>
</tr>
<tr>
<td>Stores</td>
<td>10%</td>
<td>1</td>
</tr>
<tr>
<td>ALU</td>
<td>50%</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Total Execution Time = \( \frac{(0.2 \times 1 + 0.2 \times 2 + 0.1 \times 1 + 0.5 \times 0.5) \text{ cycles/inst} \times 5 \text{B inst}}{5 \text{G cycles/sec}} \)

= 0.95 seconds

Problem 2:

a) Better memory organization: CPI for loads reduced to 1.

CPU Time = \( (0.2 \times 1 + 0.2 \times 1 + 0.1 \times 1 + 0.5 \times 0.5) \times 5 \text{B inst} / 5 \text{G} = 0.75 \text{ seconds} \)

b) Reduced the number of ALU instructions: # of ALU instructions: \( 5 \text{B} \times 50\% \times 80\% \).

CPU Time = \( (0.2 \times 1 + 0.2 \times 2 + 0.1 \times 1 + 0.5 \times 0.8 \times 0.5) \times 5 \text{B} / 5 \text{G} = 0.90 \text{ seconds} \)

Therefore, better memory organization gives better performance.
Problem 3:

a) ILP before register renaming: ILP = # of instructions / Longest Path in Dep’ Graph = 8/6
   
   cycle 1: I1, I2
   cycle 2: I3
   cycle 3: I4
   cycle 4: I5, I6
   cycle 5: I7
   cycle 6: I8

   Please note that the last instruction is BNE, a branch instruction which never writes (except for condition flags).

b) After register renaming: ILP remains the same = 8/6
   
   ADD dopey, shovel, dirt
   NOT grumpy, shovel
   ADD doc, dopey, dirt
   SUB happy, doc, grumpy
   MUL bashful, grumpy, happy
   AND sneezy, grumpy, happy
   XOR sleepy, bashful, happy
   BNE sleepy, bashful, label

   Note that every destination registers need to be renamed.
The problem clearly assumed that the result of an instruction can be written in the last cycle of its execution. Some of students ignored this assumption, so made a mistake: for example, the first instruction writes at cycle 10.