1. Answer the following questions regarding pipelined execution of this instruction sequence:

   lw $1,40($6)
   add $6,$2,$2
   sw $6,50($1)

(a) Indicate dependences and their type.
(b) Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.
(c) Assume there is full forwarding. Indicate hazards and add nop instructions to eliminate them.
(d) Assuming the following clock cycle times,
    \[ ClockPeriod_{without-forwarding} = 300\text{ps}, \]
    \[ ClockPeriod_{full-forwarding} = 400\text{ps}, \]
    \[ ClockPeriod_{alu-alu-forwarding-only} = 360\text{ps} \]
    What is the total execution time of this instruction sequence without forwarding and with full forwarding? What is the speedup achieved by adding full forwarding to a pipeline that had no forwarding?
(e) Add nop instructions to this code to eliminate hazards if there is ALU-ALU forwarding only (no forwarding from the MEM to the EX stage).
(f) What is the total execution time of this instruction sequence with only ALU-ALU forwarding? What is the speedup over a no-forwarding pipeline?

2. Assume that the instructions executed by a pipelined processor are broken down as follows:

   add 50%
   beq 25%
   lw 15%
   sw 10%

(a) Assuming there are no stalls and that 60% of all conditional branches are taken, in what percentage of clock cycles does the branch adder in the EX stage generate a value that is actually used?
(b) Assuming there are no stalls, how often (as a percentage of all cycles) do we actually need to use all three register ports (two reads and a write) in the same cycle?
(c) Assuming there are no stalls, how often (as a percentage of all cycles) do we use the data memory?