Consider the following RISC-V code sequence

```
lw  $t2, 40($t5)
add $t5, $t2, $t8
sub $t3, $t2, $t5
sw $t3, 20($t5)
```

a) *Assuming no forwarding*, identify all pipeline hazards between pairs of instructions

b) *Assuming no forwarding*, insert *stalls* as needed to overcome these hazards. How many clock cycles are needed to finish these instructions?

c) Assuming we use forwarding, insert stalls as needed to overcome these hazards. *How many clock cycles are needed to finish executing these instructions*?

Consider the following RISC-V code sequence

```
lw  $x18, 40($x5)
add $x22, $x18, $x22
lw  $x19, 40($x6)
or $x23, $x18, $x19
lw  $x24, 40($x7)
sub $x22, $x24, $x23
```

a) Assuming we use forwarding, insert stalls as needed to overcome these hazards. How many clock cycles are needed to finish executing these instructions?

b) Rearrange the instructions to minimize the need for stalls. How many clock cycles are needed to finish executing these instructions?