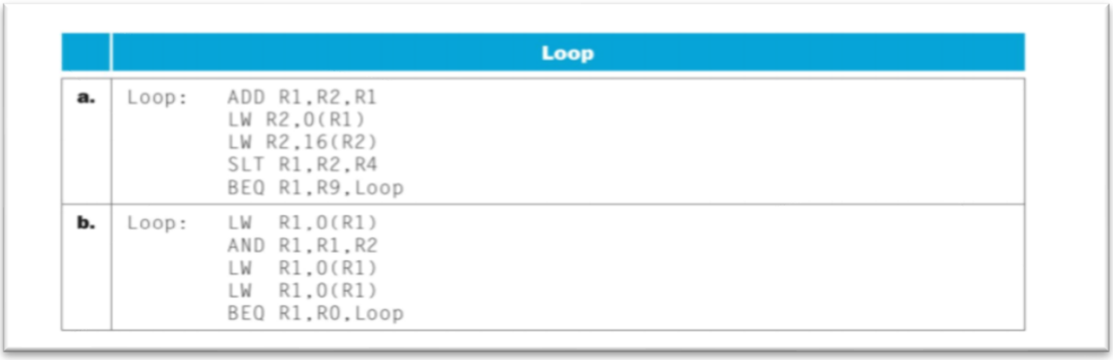
Exercise2



	Loop
a.	Loop: ADD R1,R2,R1 LW R2,0(R1) LW R2,16(R2) SLT R1,R2,R4 BEQ R1,R9,Loop
b.	Loop: LW R1,0(R1) AND R1,R1,R2 LW R1,0(R1) LW R1,0(R1) BEQ R1,R0,Loop

Execution has five stages IF, ID, MEM, WB during each clock cycle one instruction will complete a stage, the previous instruction will complete the next stage, and the following will complete the previous stage.

a)

For pipeline execution diagram for third iteration of the loop:

- Include the previous instructions from the 2nd iteration that are still executing when the 3rd iteration starts
- End the execution diagram on the clock cycle where the fourth iteration can start
- The LW instruction needs a stall since R1 which the previous add instruction uses
- The LW instruction needs a stall since R2 which the previous load instruction uses

MIPS CODE	1	2	3	4	5		
2: LW	WB						
2: SLT	EX	MEM	WB				
2: BEQ	ID	EX	MEM	WB			
	ID	EX	MEM	WB			
3: ADD	IF	ID	EX	MEM	WB		
3: LW		IF	ID	EX	MEM		
3: LW			IF	ID	STALL	EX	MEM
3: SLT				IF	STALL	ID	STALL
3: BEQ						IF	STALL

b)

For pipeline execution diagram for third iteration of the loop:

- Include the previous instructions from the 2nd iteration that are still executing when the 3rd iteration starts
- End the execution diagram on the clock cycle where the fourth iteration can start
- The AND instruction needs a stall since R1 which the previous load instruction uses
- Each instruction uses the R1 register so each instruction needs a stall so that next instruction does not use R1 until the current instruction is finished with it.

MIPS CODE	1	2	3	4	5	6	7	8	9
LW R1,0(R1)	WB								
LW R1,0(R1)	EX	MEM	WB						
BEQ R1, R0, LOOP	ID	STALL	EX	MEM	WB				
LW R1,0(R1)	IF	STALL	ID	EX	MEM	WB			
AND R1, R1,R2			IF	ID	STALL	EX	MEM	WB	
LW R1,0(R1)				IF	STALL	ID	EX	MEM	WB
LW R1,0(R1)						IF	ID	STALL	
BEQ R1,R0,LOOP							IF	STALL	