Q4. (25 pts) Use the following MIPS code fragment:

```
        I1:   ADDI  $3, $0, 100       # $3 = 100
        I2:   ADD   $4, $0, $0        # $4 = 0
   Loop:
        I3:   LW      $5, 0($1)         # $5 = MEM[$1]
        I4:   ADD   $4, $4, $5        # $4 = $4 + $5
        I5:   LW      $6, 0($2)         # $6 = MEM[$2]
        I6:   SUB   $4, $4, $6        # $4 = $4 - $6
        I7:   ADDI  $1, $1, 4         # $1 = $1 + 4
        I8:   ADDI  $2, $2, 4         # $2 = $2 + 4
        I9:   ADDI  $3, $3, -1        # $3 = $3 - 1
        I10:  BNE   $3, $0, Loop      # if ($3 != 0) goto Loop
```

a)

(10 pts) Show the timing of one loop iteration on the 5-stage MIPS pipeline **without forwarding hardware**. Complete the timing table, showing all the stall cycles. Assume that the register write is in the first half of the clock cycle and the register read is in the second half. Also assume that the branch will stall the pipeline for 1 clock cycle only. Ignore the "startup cost" of the pipeline.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1	20	21	22	2	24	25
I1	IF	ID	E	MEM	WB																				
I2		IF	I	EX	MEM	WB																			
I3			I	ID	EX	MEM	WB																		
I4				IF	Stall	Stall	ID	EX	ME	WB															
I5							IF	ID	EX	MEM	WB														
I6								IF	St	Sta	ID	EX	MEM	WB											
I7											IF	ID	EX	MEM	WB										
I8												IF	ID	EX	MEM	WB									
I9													IF	ID	EX	MEM	WB								

I10													IF	Stal	Sta	ID	EX	M	WB					
I3																IF	IF	I	EX	MEM	WB			
I4																		I	Stal	Sta	ID	E	MEM	WB

b)

According to the timing diagram of part (a), compute the number of clock cycles and the average CPI to execute ALL the iterations of the above loop.

There are 100 iterations

Each iteration requires 15 cycles =

8 cycles to start the 8 instructions in loop body + 7 stall cycles

There are 2 additional cycles to start the first 2 instructions before the loop.

Therefore, total cycles = 100 * 15 + 2 (can be ignored) = 1502 cycles ≈ 1500 cycles

Total instruction executed = 2 + 8 * 100 = 802 instructions (counting first two)

Average CPI = 1502 / 802 = 1.87

If we ignore first two instructions and the time to terminate last iteration then

Average CPI = 1500/800 = 1.88 (almost same answer)

c)

Redo part (a) to show the timing of one loop iteration with **full forwarding** hardware. If forwarding happens, please show how the data is forwarded with an arrow.

	1	2	3	4	5	6	7	8	9	10	1	12	13	14	15	16	17	18	19	20
I1	IF	ID	EX	MEM	WB															
I2		IF	ID	EX	MEM	WB														
I3			IF	ID	EX	MEM	W													
I4				IF	Stal	ID	E	MEM	WB											
I5						IF	I	EX	ME	WB										
I6							I	Stall	ID	EX	M	WB								
I7									IF	ID	E	MEM	WB							
I8										IF	I	EX	ME	WB						
I9											I	ID	EX	MEM	WB					
I10												IF	ID	EX	ME	WB				
I3													IF	IF	ID	EX	MEM	WB		
I4															IF	Stal	ID	EX	MEM	WB

d)

Reorder the instructions of the above loop to fill the load-delay and the branch delay slots, without changing the computation. Write the code of the modified loop.

```
ADDI $3, $0, 100     # $3 = 100
```

```
        ADD $4, $0, $0        # $4 = 0
Loop:
        LW $5, 0($1)          # $5 = MEM[$1]
        LW $6, 0($2)          # Moved earlier to avoid load-delay
        ADDI $3, $3, -1       # Moved earlier
        ADD $4, $4, $5        # $4 = $4 + $5
        ADDI $1, $1, 4        # $1 = $1 + 4
        ADDI $2, $2, 4        # $2 = $2 + 4
        BNE $3, $0, Loop      # if ($3 != 0) goto Loop
        SUB $4, $4, $6        # Fills branch delay slot
```

e)

(5 pts) Compute the number of cycles and the average CPI to execute ALL the iteration of the modified loop. What is the speedup factor?

There are 100 iterations

Each iteration requires 8 cycles =

8 cycles to start the 8 instructions in loop body + 0 stall cycles

There are 2 additional cycles to start the first 2 instructions before the loop

+ 4 additional cycles to terminate the ADDI instruction in the last iteration.

Therefore, total cycles = 100 * 8 + 6 (can be ignored) = 806 cycles ≈ 800 cycles

Total instruction executed = 2 + 8 * 100 = 802 instructions (counting first two)

Average CPI = 806 / 802 = 1.00

If we ignore first two instructions and the time to terminate last iteration then

Average CPI = 800/800 = 1.00 (almost same answer)

Speedup Factor = CPIpart-b/CPIpart-d = 1.88/1.00 = 1.88