Question 1

Considering the MIPS64 architecture presented in the following:

- Integer ALU: 1 clock cycle
- Data memory: 1 clock cycle
- FP multiplier unit: pipelined 8 stages
- FP arithmetic unit: pipelined 2 stages
- FP divider unit: not pipelined unit that requires 8 clock cycles
- branch delay slot: 1 clock cycle
- data forwarding is enabled
- it is possible to complete instruction EXE stage in an out-of-order fashion.

and using the following code fragment

```
; ********************** MIPS64 ************************
;for (i = 1; i <= 100; i++){
;          v5[i] = v1[i]*v2[i];
;          v6[i] = v3[i]/v4[i];
 ;}
                    .data
 V1:      .double “100 values”
 V2:      .double “100 values”
 V3:      .double “100 values”
 V4:      .double “100 values”
 V5:      .double “100 values”
 V6:      .double “100 values”
                    .text

 main:              daddui r1,r0,0
                    daddui r2,r0,100
 loop:              l.d       f1,v1(r1)
                    l.d       f2,v2(r1)
                    mul.d   f5,f1,f2
                    s.d       f5,v5(r1)
                    l.d       f3,v3(r1)
                    l.d       f4,v4(r1)
                    div.d    f6,f3,f4
                    s.d       f6,v6(r1)
                    daddui r1,r1,8
                    daddi r2,r2,-1
                    bnez r2,loop
                    halt
                                        total
```

comments	Clock cycles
r1← pointer	
r2← 100	
f1← v1[i]	
f2← v2[i]	
f5 ← v1[i]*v2[i]	
f3← v3[i]	
f4← v4[i]	
f6 ← v3[i]/v4[i]	
r1 ← r1 + 8	
r2 ← r2 – 1	

1. Show the timing of the presented loop-based program and compute how many cycles does this program take to execute?
2. Using all optimization techniques, re-write the developed code in order to eliminate the most data hazards.
    a. Compute once again the number of clock cycles needed to execute the new program