1.)

	12	18	23	40	42	44	46	48	50
B1 Buffer	0	0	0	1	0	0	0	0	0
B1 Predicted	Not taken	Not taken	Not taken	Taken	Not taken	Not taken	Not taken	Not taken	Not taken
B1 Predictor	Correct	Correct	Incorrect	Incorrect	Correct	Correct	Correct	Correct	Correct
B2 Buffer	1	1	1	0	1	1	1	1	1
B2 Predicted	Taken	Taken	Taken	Not taken	Taken	Taken	Taken	Taken	Taken
B2 Predictor	Correct	Correct	Incorrect	Incorrect	Correct	Correct	Correct	Correct	Correct
Prediction Accuracy					14/18 = 0.7777 = **77.77%**				

2.)

	12	18	23	40	42	44	46	48	50
B1 Buffer	10	00	00	01	00	00	00	00	00
B1 Predicted	Taken	Not taken	Not taken	Not taken	Not taken	Not taken	Not taken	Not taken	Not taken
B1 Predictor	Incorrect	Correct	Incorrect	Correct	Correct	Correct	Correct	Correct	Correct
B2 Buffer	10	11	10	11	11	11	11	11	11
B2 Predicted	Taken	Taken	Taken	Taken	Taken	Taken	Taken	Taken	Taken
B2 Predictor	Correct	Correct	Incorrect	Correct	Correct	Correct	Correct	Correct	Correct
Prediction Accuracy					15/18 = 0.8333 = **83.33%**				

3.)

Iteration	1-Bit	Comment
1	0	Incorrect; Branch predicted not taken
2 - 98	1	Correct; Branch predicted taken
99	1	Incorrect; Branch predicted taken
Prediction Accuracy		97/99 = 0.9797 = **97.97%**

4.)

Iteration	1-Bit	Comment
1	00	Incorrect; Branch predicted not taken
2	01	Incorrect; Branch predicted not taken
3 - 98	11	Correct; Branch predicted taken
99	11	Incorrect; Branch predicted taken
Prediction Accuracy		96/99 = 0.9696 = **96.96%**

5.)

Prediction	aa = 3 bb = 2
L1 L2 L3 Prediction	01 01 01
L1 not taken	Incorrect
L2 not taken	Correct
L3 not taken	Correct

Prediction	aa = 4 bb = 5
L1 L2 L3 Prediction	11 00 11
L1 taken	Correct
L2 not taken	Incorrect
L3 taken	Incorrect

Prediction	aa = 2 bb = 2
L1 L2 L3 Prediction	11 01 11
L1 taken	Incorrect
L2 not taken	Correct
L3 taken	Correct

Prediction Accuracy	5/9 = 0.5555 = **55.55%**

6.)

[A]

Branch Address	Predicted PC	Predicted Taken bit
2012	2020	1
2036	2064	1

[B]

Branch Address	Predicted PC	Predicted Taken bit
2012	2020	1
2024	2032	1
2036	2064	1

7.)

[A]
Write register F2 to commit && remove instruction from ROB
[B]
Write register F2 to commit && remove instruction from ROB
[C]
Write memory at 0(R1) && remove instruction from ROB
[D]
Write register R3 to commit && remove instruction from ROB
[E]
Remove instruction from ROB
[F]
Clear ROB && restart at branch inheritor

8.)

With the latencies taken into account, then the baseline performance of the code sequence, assuming only one instruction can be issued per cycle, is 38 cycles per loop iteration.

Total Cycles = (Cycles per instruction + Stall Cycles per instruction) * Number of instructions in loop + Branch Delay Slot

```
Total Cycles = (1 + 0) * 1          # I0: fld f2, 0(Rx)
             + (1 + 0) * 1          # I1: fmul.d f2, f0, f2
             + (1 + 9) * 1          # I2: fdiv.d f8, f2, f0
             + (1 + 3) * 1          # I3: fld f4, 0(Ry)
             + (1 + 0) * 1          # I4: fadd.d f4, f0, f4
             + (1 + 2) * 1          # I5: fadd.d f10, f8, f2
             + (1 + 1) * 1          # I6: fsd f4, 0(Ry)
             + (1 + 0) * 1          # I7: addi Rx, Rx, 8
             + (1 + 0) * 1          # I8: addi Ry, Ry, 8
             + (1 + 0) * 1          # I9: sub x20, x4, Rx
             + (1 + 1) * 1          # Branch delay slot
             = 38 cycles
```

9.)

Loop:	ld	t9, 0(rx)
I0:	mult.d	t10, f0, f2
I1:	div.d	t11, t9, t10
I2:	ld	t12, 0(ry)
I3:	add.d	t13, f0, t12
I4:	sub.d	t14, t11, t13
I5:	sd	t14, 0(ry)