Branch Lab - Prelab

1) We use predictors with 2-bit saturating counters because we dont want the predictor to change its prediction on the first mispredict, and then mispredict again right after, as a one bit predictor will in the case of a nested loop.

2)
```
int a;
for (int i = 0; i < 100000; i++) { // conditional branch B1
    if ((i % 4) == 0) { // conditional branch B2
        a = 10;
    }
    a = 15;
}
```

a)

We assume that the taken path occurs when the if statement confition evaluates to false and the a = 10 is not executed.

i	Taken/not taken
0	Not taken
1	Taken
2	Taken
3	Taken
4	Not taken
5	Taken
6	Taken
7	Taken
8	Not taken
9	Taken
10	Taken

...

b)

We would need 3 history bits to correctly predict all of the possible outcomes of the B2 branch, since it keeps repeating every 4 iterations of the loop.

c)

The reason that this would be a bad idea, is that both branches B1 and B2, and every other case with multiple branches closer together would both index to the same entry in the Private history table, so both branch statements would be overwriting each others history(alias), and therefore the history would be wildly inaccurate, which is why it is best to use low order bits that change with consecutive PCs.