ECE552 – Lab 2 Prelab
Bing Li – 1004191910

1. A 2-bit counter (2bc) is a branch predictor that uses a (strongly taken, taken, not taken, strongly not taken) scheme for prediction. This forces the predictor to mis-predict twice in a row before "changing" its mind, this is somewhat of an improvement to a 1-bit counter, which follows whichever direction the branch last took, which will mis-predict twice in a row in nested looping situations, whereas the 2bc will only mis-predict once and then be back on track again
2. a) The taken/not taken sequence is as follows:

i	0	1	2	3	4	5	6	7	8	…
Outcome	T	N	N	N	T	N	N	N	T	…

This is assuming that the assembly code sees the if statement as:

- MOD i, 4 -> r3
- BEQZ r3, targ

b) The smallest number of history bits needed would be 3, and the patterns used would be: TNN, NNN, NNT, and NTN. Since the repeating sequence has a length of 4, we would need the previous 3 historical outcomes to properly predict the next one

c) By indexing the BHT with the 3 most significant bits of the PC, it splits the program into large contiguous chunks, and the first 3 bits of the PC for B1 is more likely to be the same as for B2, making the per address history table no different than a global history table (i.e. making the Pag predictor into a Gag predictor). In doing this, the previously determined patterns for B2 would no longer apply since B1 and B2 are included in the same history entry