1. Consider the following C code:
   ```c
   int i = 0, sum_remainder_0 = 0, sum_remainder_1 = 0, sum_remainder_2 = 0;
   do {
       if(i % 3 == 0) // Branch X: taken if i % 3 != 0
           sum_remainder_0 += A[i];
       else
           {  
                if(i%3 == 1) // Branch Y: taken if i % 3 != 1
                    sum_remainder_1 += A[i];
                else
                    sum_remainder_2 += A[i];
           }
       i ++;
   } while (i < 10000); // Branch Z: taken if i < 10000
   
   Please answer the following questions:
   
   A. If your processor uses a “always-taken” predictor, what’s the branch prediction accuracy?
B. If your processor uses a “local 2-bit” predictor with unlimited BTB entries, what’s the branch prediction accuracy? Assume all counters are initialized as 0s.
C. If your processor uses a “global 2-bit” predictor with 4-bit GHR and unlimited BTB entries, what’s the branch prediction accuracy? Assume all counters are initialized as 0s.
D. What’s the speedup of C. over B.?