3. Branch Prediction [40 points]

Assume the following piece of code that iterates through two large arrays, j and k, each populated with completely (i.e., truly) random positive integers. The code has two branches (labeled B1 and B2). When we say that a branch is *taken*, we mean that the code *inside* the curly brackets is executed. Assume the code is run to completion without any errors (there are no exceptions). For the following questions, assume that this is the only block of code that will ever be run, and the loop-condition branch (B1) is resolved first in the iteration before the if-condition branch (B2). N and X are unspecified non-zero integers.

```
 for (int i = 0; i < N; i++) {      /* B1 */
                                     /* TAKEN PATH for B1 */
     if (i % X == 0) {               /* B2 */
         j[i] = k[i] - i;            /* TAKEN PATH for B2 */
     }
}
```

You are running the above code on a machine with a two-bit global history register (GHR) shared by all branches, which starts at *Strongly Not Taken* (2'b00). Each pattern history table entry (PHTE) contains a 2-bit saturating counter, which is initialized to *Strongly Taken* (2'b11)

The saturating counter values are as follows:
2'b00 - Strongly Not Taken
2'b01 - Weakly Not Taken
2'b10 - Weakly Taken
2'b11 - Strongly Taken

(a) Assuming that N is larger than 10 (ten), after running the loop for 10 iterations, you observe that the branch predictor mispredicts 0% of the time. What is the value of X? Explain your reasoning.

$X = 1$
In this case, the branch predictor is always right. Because all the counters in the PHTE start at Strongly Taken, it will have to mispredict at least once if the branch B2 is not taken at least once. This means B2 has to always result in a taken outcome.

(b) What is the prediction accuracy of the branch predictor if $N = 20$ and $X = 2$? Explain your answer. You can leave your answer as a fraction.

Mispredict 11 times out of 41 branches (The first ten times when there is a not taken occurs in B2, GHR entries for 11 got updated to Weakly taken, and one more mispredict branch when B1 terminates).
Accuracy = 30/41 = 73.2%

(c) Assuming that N is larger than 10 (ten), after running this code until it exits the loop, you observe that the state of the branch predictor is as follows:

GHR: 2'b00

PHTE:
NN:2'b11
NT:2'b00
TN:2'b10
TT:2'b11

What can you tell about the value of X and N? Explain your reasoning.

$X >= N$ or $N \mod X != 1$
If $X > 1$, the program will always execute with the following pattern: TTTNTNTNT...TNTTTNTNTNTN... where the TN pattern will happen based on the value of X, and at the exit of the loop the pattern will end with N. In this question, the only way the state of NT to have a value greater than zero is when the last iteration B2 is not taken.