**Branch Prediction Simulation Program**

For this project you will create a branch prediction simulator. We have looked at several algorithms in class. For this project you must implement the algorithms described in Shen and Lipasti's book. The algorithms discussed there include:

1. Smith Predictors
2. Global-History Two-level Predictors
3. Local-History Two-level Predictors
4. gshare Predictors

For this project you will have to implement one of these predictors in the order of complexity. The more complex predictors will receive more credit (see below).

The input for your program should be from a file named "branch\_trace.dat" which will be a text file. Each line of the file will represent a branch instruction. Each line will consist of an integer representing the address of the instruction, one or more blanks, and a single character either 'T' or 'N' indicating taken or not taken.   
My copy of branch\_trace.dat is available. Note that I am assuming this is simulating a modern RISC machine and that every instruction starts on a word boundary; that is all the addresses will be multiples of 4. (i.e. the last 2 bits will always be 00)

You are allotted 300 bits for your simulator. All predictors will be 2-bit Smith (aka saturating, four state) counters.

**Program output**

The output of your simulator should look like:

Number of branches: XXXX

Number of branches taken: XXXX

Number taken branches correctly predicted: XXXX

Number of branches not taken: XXXX

Number not taken branches correctly predicted: XXXX

Overall rate of correct predictions: YY.YY%

**Grading**

This project has 4 levels of complexity:

| Level | Predictor | Parameters | File Name to be used |
| --- | --- | --- | --- |
| 0 | 2-bit Smith counters | NA | smith.cc |
| 1 | Global-History Two-level Predictors | 3-bit BHR | globalHistory.cc |
| 2 | Local-History Two-level Predictors | 4-bit BHR 4 entry BHT | localHistory.cc |
| 3 | gshare Predictors | 6-bit BHR | gshare.cc |

NOTES:

1. Without prior approval, the program must be written in C++
2. It is expected that you will use multiple modules or classes for this assignment.
3. I require a class of SmithCounter. Others may be useful as well.
4. You should assume the values in the "branch\_trace.dat" file represent 16-bit addresses.