Bilal Quadri
CS211
pa3

sim

Design:
At the top of the program, variable declarations are made as per ansi specifications. The program begins by reading in the trace file specified in the arguments line by line. This string is passed to the `htoi(char* ptr)` method. This method returns an `unsigned int` conversion of the hex number on the far right of each line. This `unsigned int` is the memory address, and so the cache index and tag value are calculated using this number. Error-checking is, of course, done before-hand to ensure that the proper number of arguments are present, that a proper write policy is specified, and that the trace file exists.

A `struct` was made for each line in the cache. This `struct` contained characters for the valid bit and dirty bit, as well as an unsigned int for the tag. Having these fields in the struct allowed it to work universally with both write-through and write-back. In the main method of the program, an array of these lines was made to represent the cache.

The program is designed primarily using nested `if` and `else` statements to navigate through execution. Characters were used as flags to designate which branch to follow, so as to save memory. Such flags included `'t'` and `'b'` for write-through/write-back respectively, and `'r'` and `'w'` for read and write respectively. Areas where instructions repeat were taken into account to avoid unnecessary code repetition. A clean miss occurred when the valid bit was `'0'` at the index of the tag, and a cold miss occurred if there was a tag with a different value in that index.