For this assignment you must create a C or C++ program that simulates a direct-mapped cache. Cache size options will be determined by command-line arguments, as will the input file name. The input file will then be a sequence of memory operations, one per line, each of which are of the format "0xhhhhhhhh OP", where the first is a hexadecimal 32-bit address and OP is "R" or "W", indicating a memory read or write operation. No actual data values are recorded in the file, since they don't really matter. This kind of input file is often called a **trace**, since it is a trace of some program's memory accesses, in sequence.

Each memory access is a 32-bit (4 byte) memory access, and you can assume that each address is a**word-aligned** address, meaning its lowest two bits are 0.

Your program will need to simulate the appropriate cache (from the configuration arguments) over the memory access trace, and at the end print out: the total number of reads, the total number of writes, the number of reads that resulted in cache misses, the number of writes that resulted in cache misses, the overall hit rate (as a fraction between 0 and 1.0), and the average memory access time (in cycles).

Your program's command line syntax is:

**progname [options] inputfilename**

The input filename is not optional. Your program must also take the following command line options (any order):

* "**-b #**" will set the block size in 32-bit words (legal values are positive powers of 2, default is 1).
* "**-n #**" will set the total number of data words (not blocks) in the cache (legal values are positive powers of 2, default is 1024).
* "**-m #**" will set the number of cycles it takes to satisfy a memory operation that misses the cache (default is 100).

For example, a 1KW (1024 data words) cache with a block size of 2 would have 512 entries. The number of cycles it takes to satisfy a cache hit is assumed to be 1 (for this program). Your program should print specific error messages and then exit if there is anything wrong with the command-line arguments.

In testing your program, you should probably construct some small traces by hand, and run your program with them on a small cache configuration, and verify by hand that your program is working correctly.

For larger tests, you need to modify your program from assignments 1/2/4 to be able to print its array access trace. This trace will be all memory read operations, but that should be Ok. Use the "%p" format to print the address as a hex value.

Submit the source code for the program, and your by-hand testing traces (but not any large traces generated from your other program). Good programming practices (good commenting, good program structuring (i.e., creating and using functions), good indentation, good function and variable naming) will contribute to your grade (as will poor programming practices, too!).

### Submission

**Turned In!**

Apr 2 at 10:44pm

[Submission Details](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007)

[Download cache.cpp](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044356)

[Download cache.h](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044358)

[Download test.cpp](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044359)

[Download test.c](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044360)

[Download test.txt](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044362)

[Download makefile](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044363)

[Download README](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044364)

[Download testing\_traces.JPG](https://nmsu.instructure.com/courses/1049933/assignments/4984078/submissions/3367007?download=107044365)

Grade: 10 (10 pts possible)

Graded Anonymously: no

#### **Comments:**

No Comments