CDA5155/CDA4150 - Fall 2025
Assignment 2
Deadline: Oct 19, 2025 Sunday 11:59 pm

Copyright Statement

Notes

- Submit your report in PDF, and your source code to Canvas.
- This is **not** a group project, violation of the Academic Honor Policy will result in a failure in grade.

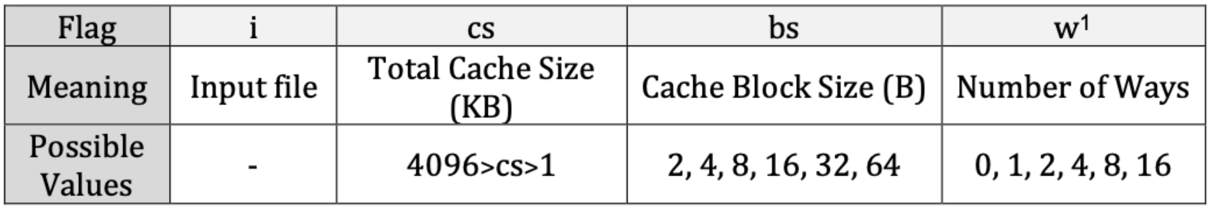
In this homework, you need to design and implement a simple cache simulator, with support for direct-mapped, set-associative, and fully associative mapping. You will implement from scratch with C/C++/Java/Python. As a cache simulator, your program is required to have the ability to collect cache misses and compute the cache miss rate.

To facilitate the testing of your cache simulator, we will utilize two sets of traces collected from a run of gcc. The traces gcc-10K.memtrace, and gcc-1M.memtrace are attached in Canvas. The traces contain ~10 thousand and ~1.5 million entries. A sample of the trace is given below:

```
L -200 7fffe7ff088
L 0 7fffe7fefd0
S 8 12ff228
S 8 12ff208
L 0 a295e8
```

Each line in the trace file contains the memory instruction type (L = load, S = store), the offset in decimal, and the memory address in hexadecimal. Note that this trace was obtained from an x86 machine, and thus the memory address is 44-bits. For this class, we can assume 32-bits and you can truncate the most significant 12 bits. LRU should be used as the cache replacement policy.

Your cache simulator is required to support the following input arguments (flags).



Flag	i	cs	bs	w[1]
Meaning	Input file	Total Cache Size (KB)	Cache Block Size (B)	Number of Ways
Possible Values	-	4096>cs>1	2, 4, 8, 16, 32, 64	0, 1, 2, 4, 8, 16

Requirements:

1. [80 points] For any given input arguments combination, your simulator should read the memory traces, simulate the cache behavior according to input arguments, and output the correct cache miss rate.

2. [20 points] Report the cache miss rate in the form below assuming we have a 512KB cache and 16B block size.

Trace	Direct	2-way	4-way	Fully Assoc.
gcc-10K				
gcc-1M				

3. [Bonus, 30 points] Add multi-level cache or victim cache to your cache simulator. You will need to explain your design and demonstrate the correctness of your code.

To help you verify the correctness of your program, we provide one additional input (gcc-1K.memtrace) and some cache results (gcc-1K_results.txt) for it. You can compare your cache miss rate with the one in the result file.