Cache Simulator (800 points):

For this assignment you will write a C++ program to simulate caches of different configurations.

We will make the following assumptions about all the caches we simulate.

1. The replacement strategy is LRU (Least Recently Used), if needed.
2. The caches are initially empty.
3. The caches use “allocate on write” policy, i.e. we treat stores similar to loads and fetch a block from memory if the store results in a miss.

We simulate caches with different settings in terms of total size, block size, and mapping type. These different setting are enumerated below.

Cache Size (in terms of bytes):

1. 1K
2. 4K
3. 64K
4. 128K

Block Size (in terms of bytes):

1. 8
2. 16
3. 32
4. 128

Mapping Type:

1. Direct Mapped
2. 2-way set associative
3. 4-way set associative
4. Fully associative

Write Policy:

1. Write back
2. Write through

Therefore, we will simulate 4x4x4x2, i.e. 128 different cache configurations.

Input Files Format: We will use trace files to simulate our cache configurations. Each trace file is a sequence of memory accesses, one on each line, with the format:

```
read/write  addressHex
```

where *read/write* is either R or W and address is a byte address written in hex, and is in the range 0 to (2^32) - 1. An example file is provided under the assignment link.

Output Files Format: Your program should summarize the result of simulation for different cache configurations for a given trace file in a text file called `*.result`, where * is the name of the trace. For example for `mm1.trace` the output is `mm1.result`. An example file is provided under the assignment link. You can view this using wordpad or other text editors.

There are 128 lines in each output file. Each line represents result for one specific cache configuration and is formatted as follows:

```
1024 8 FA WB 0.91 1024 512 64
```

The above numbers represent the following:

1. Cache size (in bytes)
2. Block size (in bytes)
3. Mapping type (DM, 2W, 4W, and FA for direct-mapped, 2-way and 4-way set associative, and fully-associative caches respectively)
4. Write policy (WB, WT)
5. Hit rate (two decimal points)
6. Total bytes transferred from memory to the cache
7. Total bytes transferred from cache to memory
8. Number of comparisons to identify a hit

Execution Format: Give your executable the name “simulate”. The trace that is simulated should follow the name of executable w/o the extension. In other words, in order to simulate the cache for the test.trace, you need to type the following in the console:

```
simulate test
```

Trace Files (120 points):

As part of this project, you will have to generate two traces to test your cache simulator. Here are the traces you need to create:

1. Create a trace that shows the effect of associatively. In other words, create a trace that results in higher miss rate in direct mapped cache and lower miss rate in set associative caches. Name this trace associative.trace.
2. Create a trace that shows the effect of increasing the block size. Specifically, your trace should show an improvement in hit rate when the block size is increased from 8 byte to 16 bytes. However, you have to show to the improvement stops at some point and increasing the block size hurts hit rate beyond that point. Use the 1K byte direct-mapped cache as the basis for this example. Name this trace block.trace.

Report (80 points):

Write a short report (between 200 and 300 words) that describes your approach in implementing the project. Also, include two graphs representing the two traces that you have created. The first one should show miss rate with respect to associativity and the second one should show miss rate with respect to block size. Describe each graph in a few sentences.

Getting help: As mentioned in syllabus, you are strongly encouraged to use the forum on scholar. Please send us an email after you post a question on forum, so that we accidentally do not miss the question. You should expect to get a response from the instructor or TA within 24 hours. Meanwhile, other students are encouraged to respond to posts, if they think they know the answer. ***You can submit traces (other than the two requested trace above) and their corresponding result file to the forum.*** You are also encouraged to start the project early and go to OH to get help. Questions sent to the TA or instructor in the last 24 hours before the deadline, may not be answered.