Topic 1-1: Basic Cache Simulator (One-Person Group)

In this project you will design a one-level blocking cache simulator with the following **configurations**:

1. Cache Line Size;
2. Associativity;
3. Data Cache Size;
4. Cache Replacement Policy;
5. Write-Though/Write-Back;
6. Write Allocation/Write Around;
7. Write Buffer Size;
8. Miss Penalty;
9. Hit Time.

The parameter settings should be reasonable and justification should be given in the report.

To test your simulator, you need to design two **benchmarks** (choose one ISA from RISC-V; MIPS; ARM, etc.) that can result in cache hits, misses, and evicts.
The outputs of the simulator should include the following information:

1. Cache architecture configuration.
2. Indicate whether each instruction results in cache operation. If so, please also indicate whether it is a cache hit, miss, or evict.
3. Show the # of cache hits, misses, evicts. Calculate the hit rate, miss rate.
4. Screenshots of the results and the output files are both required.

Topic 1-2: Intermediate Cache Simulator (Two-Person Group)

1. Finish the design of the above Basic Cache Simulator.
2. Design another Replacement Policy (If the basic one is The Lease Recently Used, you can explore The Most Recently Used, etc.).
3. Compare the benchmark results of different policies.

Topic 1-3: Advanced Cache Simulator (Three-Person Group)

1. Finish the design of the above Intermediate Cache Simulator.
2. Design Write-Back and Write Through cache separately.
3. Compare the benchmark results of different policies.

Bonus (10%):

Modify your original design including configuration and benchmark to realize a non-blocking cache. Show the improvements over the original cache.

Programming:

1. Choose a popular programming language for the project (C, C++, C#, Python, Java, etc.)
2. UI is unnecessary. Command window and logged data for output will be enough.