# SimpleScalar Code Guide

All the parameters we changed can be seen from the PowerPoint.
Below is some example on how to run the simulation.

1. When we use sim-cache as simulator and use test-math as benchmark:

```
./sim-cache -cache:il1 il1:32:16:1:l -cache:il2 il2:32:32:1:l ./tests-pisa/bin.little/test-math
```

*-cache:il1 il1:32:16:1:l :*

*S*et the parameters of cache L1. The number of sets is 32, the block size is 16, and it is a one-way associativity cache, LRU.

*-cache:il2 il2:32:32:1:l*

Set the parameter of cache L2. The number of sets is 32, the block size is 32, and it is a one-way associativity cache, LRU.

*./test-pisa/bin.little/test-math*

Enables the test-math benchmark.

2. When we use sim-cache as simulator and use anagram as benchmark:

```
./sim-cache -cache:il1 il1:32:16:1:l -cache:il2 il2:32:32:1:l ./tests-pisa/bin.little/anagram
```

During the experience, we change the parameters manually and get the corresponding miss rate.

3. When we use sim-cheetah as simulator and use test-math as benchmark

```
./sim-cheetah -l 5 -a 4 -b 10 -n 3 -refs unifed ./tests-pisa/bin.little/test-math
```

1. Unified cache (Reference stream to analyze) == -refs unified.
2. Least-recently-used (LRU) replacement policy == Default using (LRU).
3. 16 to 1024 sets == (-a 4 -b 10 = where 16=2^4 then a=4 and 1024 =2^10 then b=10).
4. 1-way to 8-way associativity ==-(n 3 where 8=2^3 is the max range).
5. 32-byte cache line size == (-l 5 where 32= 2^5 then l=5).

4. When we use sim-cheetah as simulator and use anagram as benchmark

```
./sim-cheetah -l 5 -a 4 -b 10 -n 3 -refs unifed ./tests-pisa/bin.little/anagram
```

We change the parameters manually and get the corresponding miss rate.