Programming Project #1
Cache Simulator

How to compile the code:
This command compiles **cachesim.c** source file and generates a binary, executable file, cachesim. Or, you can use the make utility.

$ gcc cachesim.c -o cachesim

How to run the code:
To run the sample simulator, please use a command like below. You need to use the **argument** direct to let simulator know it is the **direct-mapped cache (direct), n-way associative (n-way), fully associative (fully)** and the argument **tracefile** represents the **path** of the file that contains the memory traces.

$ ./cachesim direct tracefile
$ ./cachesim n-way tracefile
$ ./cachesim fully tracefile

How to Change Properties for Execution:
In **cachesim.h** change the **BLOCK_SIZE, WAY_SIZE** and **CACHE_SIZE** values, so that we can get the implementation of n-way and fully for **each case hit rate** and **miss rate**.

Part 1. Direct-mapped cache
Requirements of Part 1:
1. The provided sample simulator only measures the total cache hits and misses. Please make modifications to the cachesim.c source code to calculate the cache miss rate and hit rate and use the print statements to print out the results.

```
==================================
Cache type:    Direct-Mapped Cache
==================================
Cache Hits:    7
Cache Misses:  5
==================================
Cache Hit Rate: 0.58
Cache Miss Rate: 0.42
```

2. Please compile and run the modified simulator and report the hit and miss rate for each given trace.

Trace	Block Size	Cache Size	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	32768	7	5	0.58	0.42
trace.hpcg	64	32768	1301327	198673	0.87	0.13
trace.stream	64	32768	1059918	2174137	0.33	0.67
trace.stream_10	64	32768	0	10	0	1
trace.stream_20	64	32768	4	16	0.2	0.8

Note: Output file for the test cases for each part was in ./cs5375/Output files/Part 1

Part 2. Fully associative and n-way set associative cache

Develop the simulator to simulate fully associative cache and n-way set associative cache with random replacement policy.

Requirements of Part 2:
1. Given a fixed cache size of 32KB, test the fully associative, 8-way set associative, 4-way set associative, and 2-way set associative cache with cache line size of 16 bytes, 32 bytes, and 128 bytes, respectively. Please report the hit and miss rate in each case.

Fully Associative

Trace	Block Size	Cache Size	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	32768	7	5	0.58	0.42
hw2_q4_memoryaddr	32	32768	5	7	0.42	0.58
hw2_q4_memoryaddr	16	32768	3	9	0.25	0.75
hw2_q4_memoryaddr	128	32768	7	5	0.58	0.42
trace.hpcg	64	32768	1321345	178655	0.88	0.12
trace.hpcg	32	32768	1299306	200694	0.87	0.13
trace.hpcg	16	32768	1288250	211750	0.86	0.14
trace.hpcg	128	32768	1370169	129831	0.91	0.09
trace.stream	64	32768	1075745	2158310	0.33	0.67
trace.stream	32	32768	1069028	2165027	0.33	0.67
trace.stream	16	32768	105441	2179638	0.33	0.67
trace.stream	128	32768	1078587	2155468	0.33	0.37
trace.stream_10	64	32768	0	10	0	1
trace.stream_10	32	32768	0	10	0	1
trace.stream_10	16	32768	0	10	0	1
trace.stream_10	128	32768	0	10	0	1
trace.stream_20	64	32768	4	16	0.2	0.8
trace.stream_20	32	32768	4	16	0.2	0.8
trace.stream_20	16	32768	3	17	0.15	0.85
trace.stream_20	128	32768	4	16	0.2	0.8

8-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	32768	8	7	5	0.58	0.42
hw2_q4_memoryaddr	32	32768	8	5	7	0.42	0.58
hw2_q4_memoryaddr	16	32768	8	3	9	0.25	0.75
hw2_q4_memoryaddr	128	32768	8	7	5	0.58	0.42
trace.hpcg	64	32768	8	1302854	197146	0.87	0.13
trace.hpcg	32	32768	8	1288887	211113	0.86	0.14

trace.hpcg	16	32768	8	1278517	221483	0.85	0.15
trace.hpcg	128	32768	8	1345457	154543	0.9	0.1
trace.stream	64	32768	8	1060143	2173912	0.33	0.67
trace.stream	32	32768	8	1062536	2171519	0.33	0.67
trace.stream	16	32768	8	1070159	2163896	0.33	0.67
trace.stream	128	32768	8	1061353	2172702	0.33	0.67
trace.stream_10	64	32768	8	0	10	0	1
trace.stream_10	32	32768	8	0	10	0	1
trace.stream_10	16	32768	8	0	10	0	1
trace.stream_10	128	32768	8	0	10	0	1
trace.stream_20	64	32768	8	4	16	0.2	0.8
trace.stream_20	32	32768	8	4	16	0.2	0.8
trace.stream_20	16	32768	8	3	17	0.15	0.85
trace.stream_20	128	32768	8	4	16	0.2	0.8

4-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	32768	4	7	5	0.58	0.42
hw2_q4_memoryaddr	32	32768	4	5	7	0.42	0.58
hw2_q4_memoryaddr	16	32768	4	3	9	0.25	0.75
hw2_q4_memoryaddr	128	32768	4	7	5	0.58	0.42
trace.hpcg	64	32768	4	1304159	195841	0.87	0.13
trace.hpcg	32	32768	4	1286974	213026	0.86	0.14
trace.hpcg	16	32768	4	1276998	223002	0.85	0.15
trace.hpcg	128	32768	4	1348278	151722	0.9	0.1
trace.stream	64	32768	4	1061857	2172198	0.33	0.67
trace.stream	32	32768	4	1062593	2171462	0.33	0.67
trace.stream	16	32768	4	1048050	2186005	0.32	0.68
trace.stream	128	32768	4	1062271	2171784	0.33	0.67
trace.stream_10	64	32768	4	0	10	0	1
trace.stream_10	32	32768	4	0	10	0	1
trace.stream_10	16	32768	4	0	10	0	1
trace.stream_10	128	32768	4	0	10	0	1
trace.stream_20	64	32768	4	4	16	0.2	0.8
trace.stream_20	32	32768	4	4	16	0.2	0.8
trace.stream_20	16	32768	4	3	17	0.15	0.85
trace.stream_20	128	32768	4	4	16	0.2	0.8

2-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	32768	2	7	5	0.58	0.42
hw2_q4_memoryaddr	32	32768	2	5	7	0.42	0.58
hw2_q4_memoryaddr	16	32768	2	3	9	0.25	0.75
hw2_q4_memoryaddr	128	32768	2	7	5	0.58	0.42
trace.hpcg	64	32768	2	1297785	202215	0.87	0.13
trace.hpcg	32	32768	2	1281081	218919	0.85	0.15
trace.hpcg	16	32768	2	1271696	228304	0.85	0.15
trace.hpcg	128	32768	2	1341692	158308	0.89	0.11
trace.stream	64	32768	2	1062079	2171976	0.33	0.67
trace.stream	32	32768	2	1062355	2171700	0.33	0.67
trace.stream	16	32768	2	1047845	2186210	0.32	0.68
trace.stream	128	32768	2	1059491	2174564	0.33	0.67
trace.stream_10	64	32768	2	0	10	0	1
trace.stream_10	32	32768	2	0	10	0	1
trace.stream_10	16	32768	2	0	10	0	1
trace.stream_10	128	32768	2	0	10	0	1
trace.stream_20	64	32768	2	4	16	0.2	0.8
trace.stream_20	32	32768	2	4	16	0.2	0.8
trace.stream_20	16	32768	2	3	17	0.15	0.85
trace.stream_20	128	32768	2	4	16	0.2	0.8

Note: *Output file for the test cases for each part was in ./cs5375/Output files/Part 2*

2. Given a fixed cache line size of 64 bytes, test the fully-associative, 8-way set associative, 4-way set associative, and 2-way set associative cache with the cache size of 16KB, 32KB and 64KB, respectively. Please report the hit and miss rate in each case.

Fully Associative

Trace	Block Size	Cache Size	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	16384	7	5	0.58	0.42
hw2_q4_memoryaddr	64	32768	7	5	0.58	0.42
hw2_q4_memoryaddr	64	65536	7	5	0.58	0.42
trace.hpcg	64	16384	1313891	186109	0.88	0.12
trace.hpcg	64	32768	1321345	178655	0.88	0.12
trace.hpcg	64	65536	1325026	174974	0.88	0.12
trace.stream	64	16384	1074086	2159969	0.33	0.67
trace.stream	64	32768	1075745	2158310	0.33	0.67

trace.stream	64	65536	1076596	2157459	0.33	0.67
trace.stream_10	64	16384	0	10	0	1
trace.stream_10	64	32768	0	10	0	1
trace.stream_10	64	65536	0	10	0	1
trace.stream_20	64	16384	4	16	0.2	0.8
trace.stream_20	64	32768	4	16	0.2	0.8
trace.stream_20	64	65536	4	16	0.2	0.8

8-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	16384	8	7	5	0.58	0.42
hw2_q4_memoryaddr	64	32768	8	7	5	0.58	0.42
hw2_q4_memoryaddr	64	65536	8	7	5	0.58	0.42
trace.hpcg	64	16384	8	1294583	205417	0.86	0.14
trace.hpcg	64	32768	8	1302854	197146	0.87	0.13
trace.hpcg	64	65536	8	1E+06	2E+06	0.33	0.67
trace.stream	64	16384	8	1059701	2174354	0.33	0.67
trace.stream	64	32768	8	1060143	2173912	0.33	0.67
trace.stream	64	65536	8	1070159	2163896	0.33	0.67
trace.stream_10	64	16384	8	0	10	0	1
trace.stream_10	64	32768	8	0	10	0	1
trace.stream_10	64	65536	8	0	10	0	1
trace.stream_20	64	16384	8	4	16	0.2	0.8
trace.stream_20	64	32768	8	4	16	0.2	0.8
trace.stream_20	64	65536	8	4	16	0.2	0.8

4-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	16384	4	7	5	0.58	0.42
hw2_q4_memoryaddr	64	32768	4	7	5	0.58	0.42
hw2_q4_memoryaddr	64	65536	4	7	5	0.58	0.42
trace.hpcg	64	16384	4	1268456	231544	0.85	0.15
trace.hpcg	64	32768	4	1304159	195841	0.87	0.13
trace.hpcg	64	65536	4	1316872	183128	0.88	0.12
trace.stream	64	16384	4	1037054	2197001	0.32	0.68
trace.stream	64	32768	4	1061857	2172198	0.33	0.67
trace.stream	64	65536	4	1070139	2163916	0.33	0.67
trace.stream_10	64	16384	4	0	10	0	1
trace.stream_10	64	32768	4	0	10	0	1

trace.stream_10	64	65536	4	0	10	0	1
trace.stream_20	64	16384	4	4	16	0.2	0.8
trace.stream_20	64	32768	4	4	16	0.2	0.8
trace.stream_20	64	65536	4	4	16	0.2	0.8

2-way set associative

Trace	Block Size	Cache Size	N- Way	Hit	Miss	Hit Rate	Miss Rate
hw2_q4_memoryaddr	64	16384	2	7	5	0.58	0.42
hw2_q4_memoryaddr	64	32768	2	7	5	0.58	0.42
hw2_q4_memoryaddr	64	65536	2	7	5	0.58	0.42
trace.hpcg	64	16384	2	1267562	232438	0.85	0.15
trace.hpcg	64	32768	2	1297785	202215	0.87	0.13
trace.hpcg	64	65536	2	1317758	182242	0.88	0.12
trace.stream	64	16384	2	1044099	2189956	0.32	0.68
trace.stream	64	32768	2	1062079	2171976	0.33	0.67
trace.stream	64	65536	2	1070189	2163866	0.33	0.67
trace.stream_10	64	16384	2	0	10	0	1
trace.stream_10	64	32768	2	0	10	0	1
trace.stream_10	64	65536	2	0	10	0	1
trace.stream_20	64	16384	2	4	16	0.2	0.8
trace.stream_20	64	32768	2	4	16	0.2	0.8
trace.stream_20	64	65536	2	4	16	0.2	0.8

Note: *Output file for the test cases for each part was in ./cs5375/Output files/Part 2*

Part 3. Two-level cache simulation

Simulation of two-level cache

Requirements of Part 3:
Please simulate a two-level cache. L1 is a 2-way, 64KB cache, with 64B block size. L2 is an 8-way, 1MB cache, with 64B block size. The replacement policy is always a random policy. Please conduct the simulation using the provided trace files and report the hit and miss rate of L1 and L2, in each case.

We are using

(i) **L-1 2-way** associative with **Cache Size 64KB** and **Block Size 64B**
(ii) **L2 8-way** associative with **Cache Size 1MB** and **Block Size 64B**

Trace	hw2_q4_memoryaddr	trace.hpcg	trace.stream	trace.stream_10	trace.stream_20

L1 Hit	4	1219785	982082	0	1
L1 Miss	8	280215	2251973	10	19
L1 Hit Rate	0.33	0.81	0.3	0	0.05
L1 Miss Rate	0.67	0.19	0.7	1	0.95
Total CPU Access	20	1780215	5486028	20	39
L2 Hits	3	102444	94027	0	3
L2 Misses	5	177771	2157946	10	16
L2 Local Hit Rate	0.38	0.37	0.04	0	0.16
L2 Local Miss Rate	0.62	0.63	0.96	1	0.84
L2 Global Hit Rate	0.58	0.88	0.33	0	0.2
L2 Global Miss Rate	0.42	0.42	0.67	1	0.8

Note: *Output file for the test cases for each part was in ./cs5375/Output files/Part 3*

Discuss your findings/insights (by all the graphs from Excel Project1_dataset.xlsx**)**

1. High Volume of Unique Values = High Miss rate.
2. High Volume of Duplicate Values = High Hit rate.
3. Random replacement policy plays a vital role, because of it the miss rate and hit rate changes every time.