

CSARCH2-MP2 User's Manual

The application simulates a Full Associative Cache Mapping Function with MRU (Most Recently Used) as the Replacement Algorithm. It is able to support the following:

- Sequential that runs a single time
- Sequential with loop
- Sequential with varying loops
- Sequential with single-level nested loops
- Sequential with disjointed sequences (e.g. blocks 0-5, then blocks 10-15)

Furthermore, only the following inputs are supported:

- **Block Size:** Numbers only (words)
- **Cache Memory Size:** Numbers only (blocks or words)
- **Main Memory Size:** Numbers only (blocks or words)
- **Cache Access Time:** Numbers only (nanoseconds)
- **Main Memory Access Time:** Numbers only (nanoseconds)
- **Read Type:** Load Through / Non-load Through only
- **Sequence:** Comma-separated values (x,y,z), Range values (x-y), or a combination of both (x-y, z). Note that range values must not have space before and after the dash (blocks or addresses). Note that hex inputs are **NOT** supported.

Pages

A. Input Page

FA - MRU Simulator

Block Size ① 16 words

Main Memory Size ② 32 Blocks

MM Access Time ③ 10 ns

Cache Memory Size ④ 32 Blocks

Cache Access Time ⑤ 1 ns

Read Type ⑥ Load Through

Next ⑦

The user is expected to input the following information to perform the cache simulation:

1. **Block Size:** Size of a block in words.
2. **Main Memory Size:** Size of the Main Memory. May be either in blocks or words.
3. **Main Memory Access Time:** Main Memory Access Time represented in nanoseconds.
4. **Cache Memory Size:** Size of the Cache Memory Size. May be either in blocks or words.
5. **Cache Access Time:** Cache Access Time represented in nanoseconds.
6. **Read Type:** Read type of the cache. May either be Load-Through or No Load-Through.
 - **Load-Through:** The desired information can optionally be sent to the processor prior to completion of the cache line fill.
 - **No Load-Through:** The information is read from the main memory and a block or cache line is copied from the main memory onto the cache (cache line fill). The data is then transferred from the cache to the CPU.
7. **Next Button:** Go to the Sequence Page (Enabled once all the fields are filled)

B. Sequence Page

The screenshot shows the 'FA - MRU Simulator' window. At the top, there is a 'Number of Sequence Groups' input field with the value '3' (callout 1), a 'Create' button (callout 2), and a 'Blocks' dropdown menu (callout 3). Below this is a large container for 'Sequences'. Inside, there are three sequence groups. Each group has a '+' button (callout 4) and a '-' button (callout 5) to the left of a 'Group' label. The first group has a 'Sequence' input field with '1, 3, 5' (callout 7) and a 'Repetitions' input field with '1' (callout 6). The second group has a 'Sequence' input field with '9-15' and a 'Repetitions' input field with '1'. The third group has two 'Sequence' input fields: the first contains '1, 3-10, 5' and the second contains '6'. The 'Repetitions' input field for the third group is '4'. Below the sequence groups is a 'Begin Simulate' button (callout 9).

1. **Number of Sequence Groups:** This determines the number of sequence groups in the simulation. Generally, this is useful for creating a group of sequences (where each can have varying number of loops) that will be looped n times; this allows support for nested loop.
2. **Create Button:** Creates/updates number of sequence groups in the sequence group input area of the page.
3. **Input Type:** Determines whether sequence input is in terms of Blocks or Addresses
4. **Add Sequence Button:** Adds a new sequence field in the sequence group.
5. **Remove Sequence Button:** Removes endmost sequence field in the sequence group.
6. **Number of Group Repetitions:** Determines the number of times a sequence group will loop.
7. **Sequence:** A data sequence is represented as a string of characters. One inner loop is also supported per data sequence. **Repetitions** refer to the number of times the line sequence will be

looped/repeated.

Examples of sequences with loops:

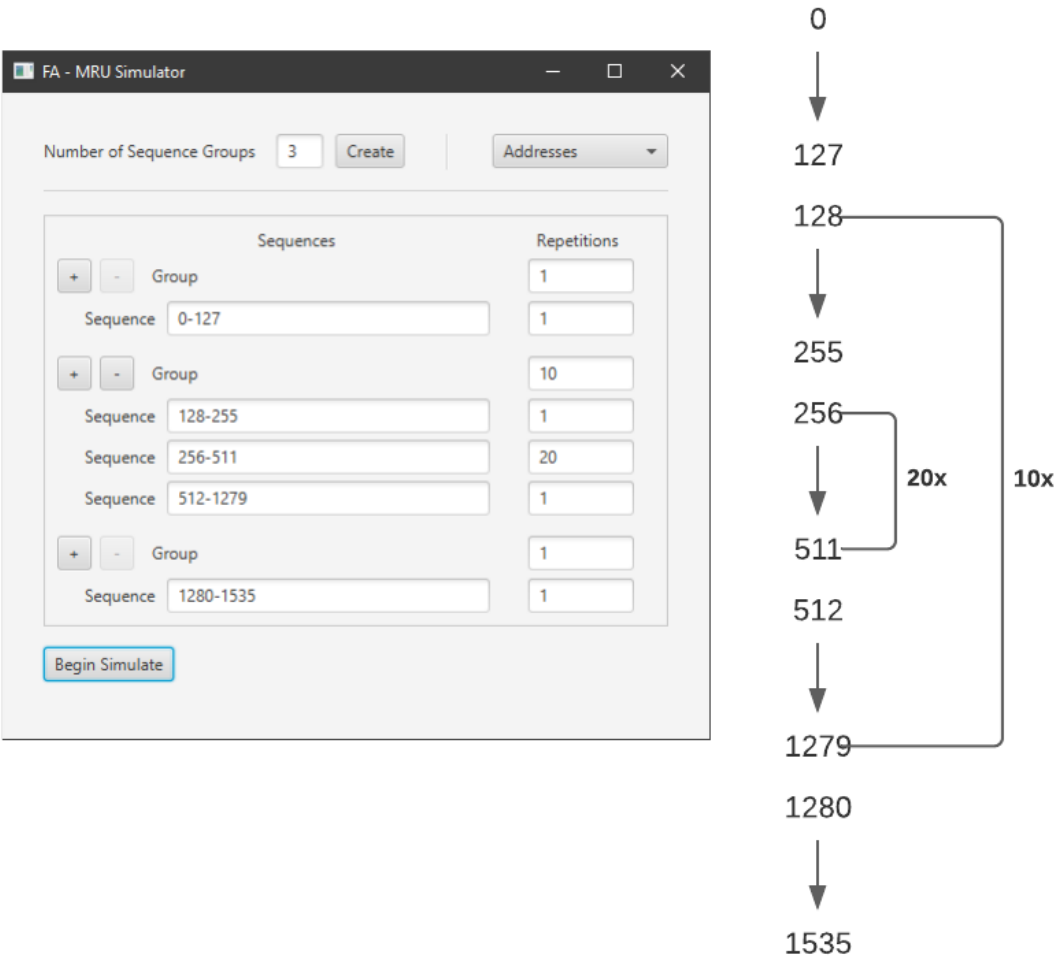


Figure 1: Sample sequence pattern with one loop

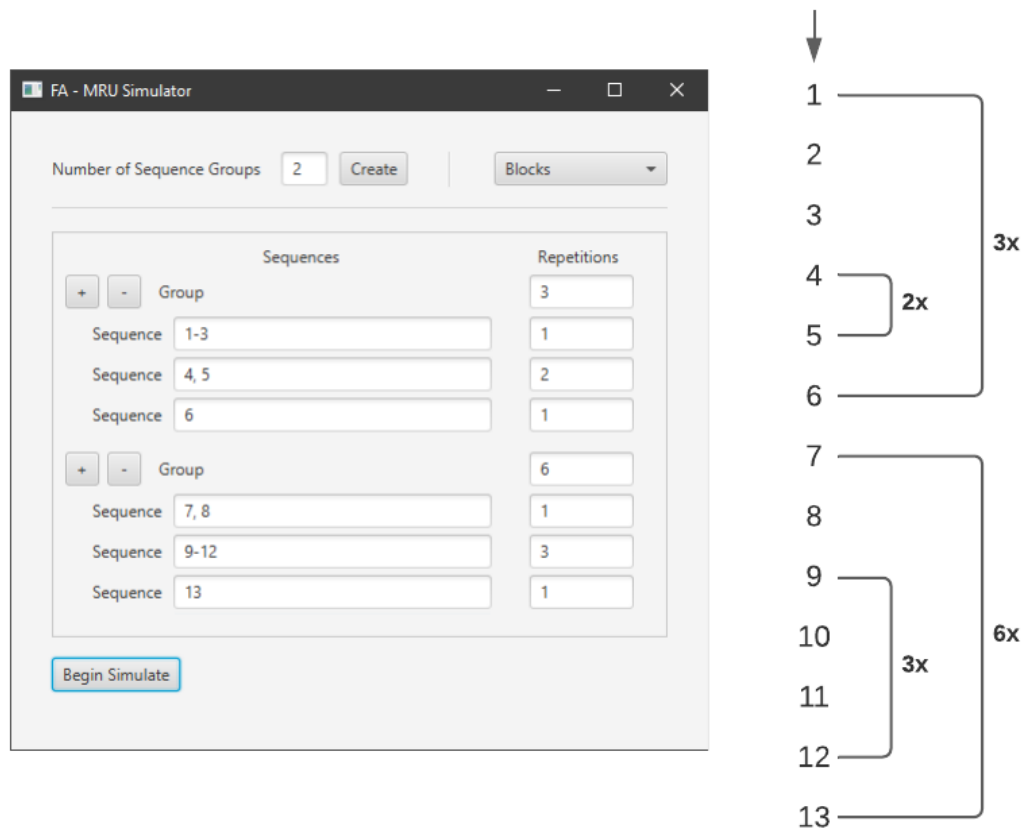


Figure 2: Sample sequence pattern with two loops

8. **Number of Sequence Repetitions:** Determines the number of times a sequence will loop.
9. **Begin Simulate:** Run simulation.

C. Output Page

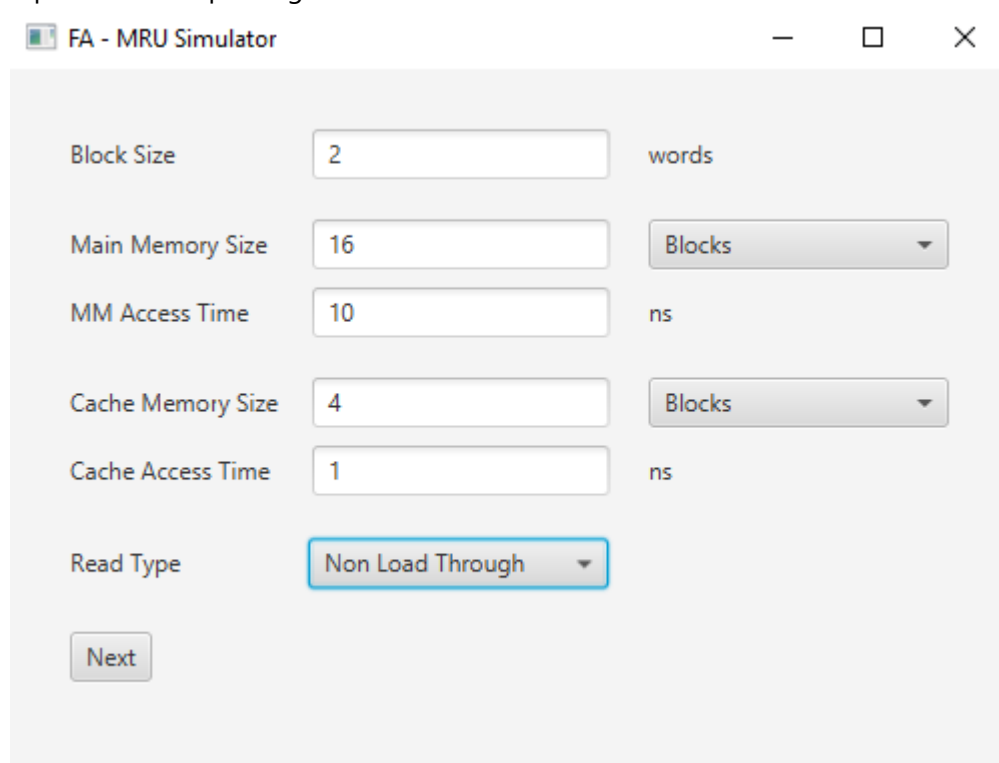
Once the necessary fields have been filled up, the application will display the following information:

1. **Hit Rate:** Percentage of memory accesses found in the cache memory.
2. **Miss Rate:** Percentage of memory accesses not found in the cache.
3. **Miss Penalty:** Extra time required to fetch a block from the Main Memory into the cache memory.
4. **Average Access Time:** Average memory access time retrieved by the processor.
5. **Total Access Time:** Total memory access time with cache of the program.
6. **Cache Contents:** The representation of the cache memory (and its contents) in table form.
7. **Export Results Button:** Exports the result to a text file.

Sample Inputs

A. Memory Size in Blocks

1. Inputs for the Input Page:

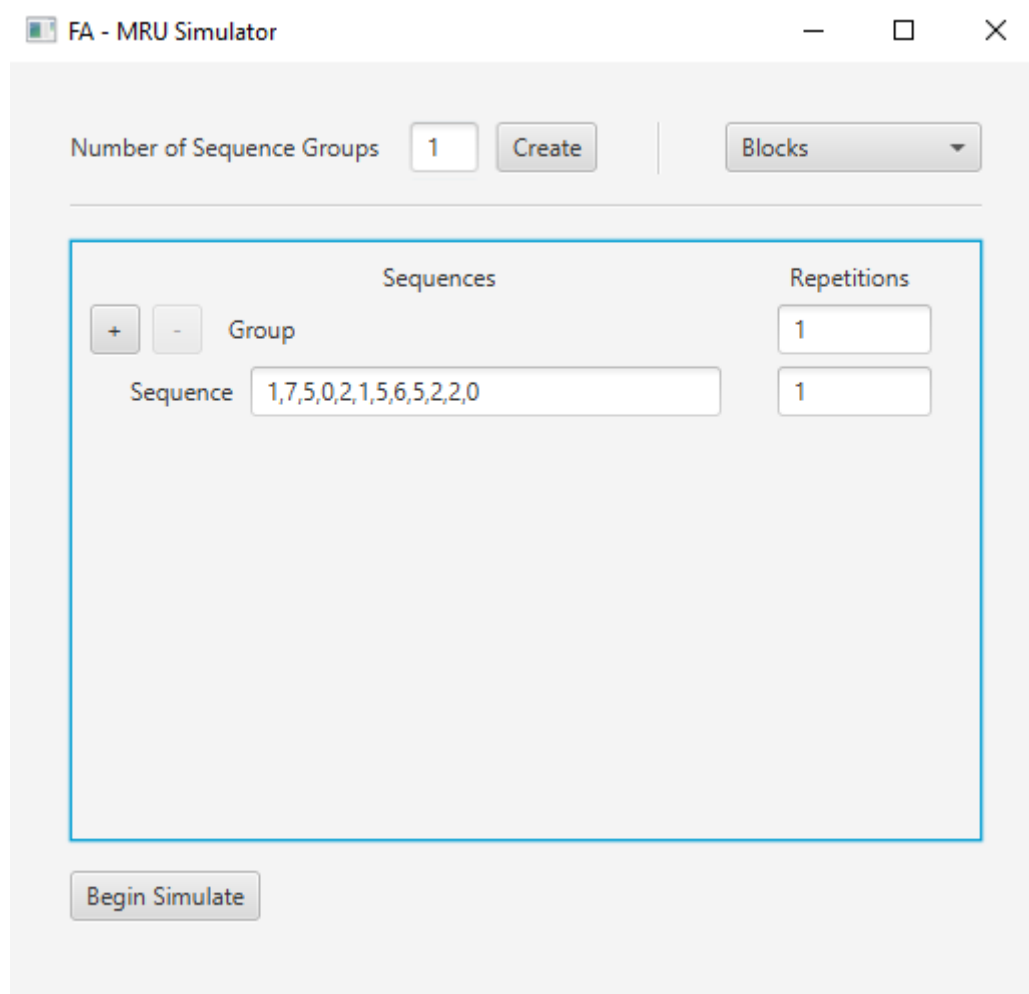


The screenshot shows the 'FA - MRU Simulator' window with the following inputs:

Parameter	Value	Unit/Type
Block Size	2	words
Main Memory Size	16	Blocks
MM Access Time	10	ns
Cache Memory Size	4	Blocks
Cache Access Time	1	ns
Read Type	Non Load Through	

A 'Next' button is located at the bottom left of the input area.

2. Inputs for the Sequence Page:



The screenshot shows the 'FA - MRU Simulator' window with the following inputs:

Number of Sequence Groups: 1 [Create] [Blocks]

Sequences	Repetitions
Group: + - Sequence: 1,7,5,0,2,1,5,6,5,2,2,0	1 1

A 'Begin Simulate' button is located at the bottom left of the sequence area.

3. Output:

FA - MRU Simulator

Block	Data
0	1
1	7
2	5
3	0

Cache Hits4

Cache Misses8

Miss Penalty22.0

Average Memory Access Time

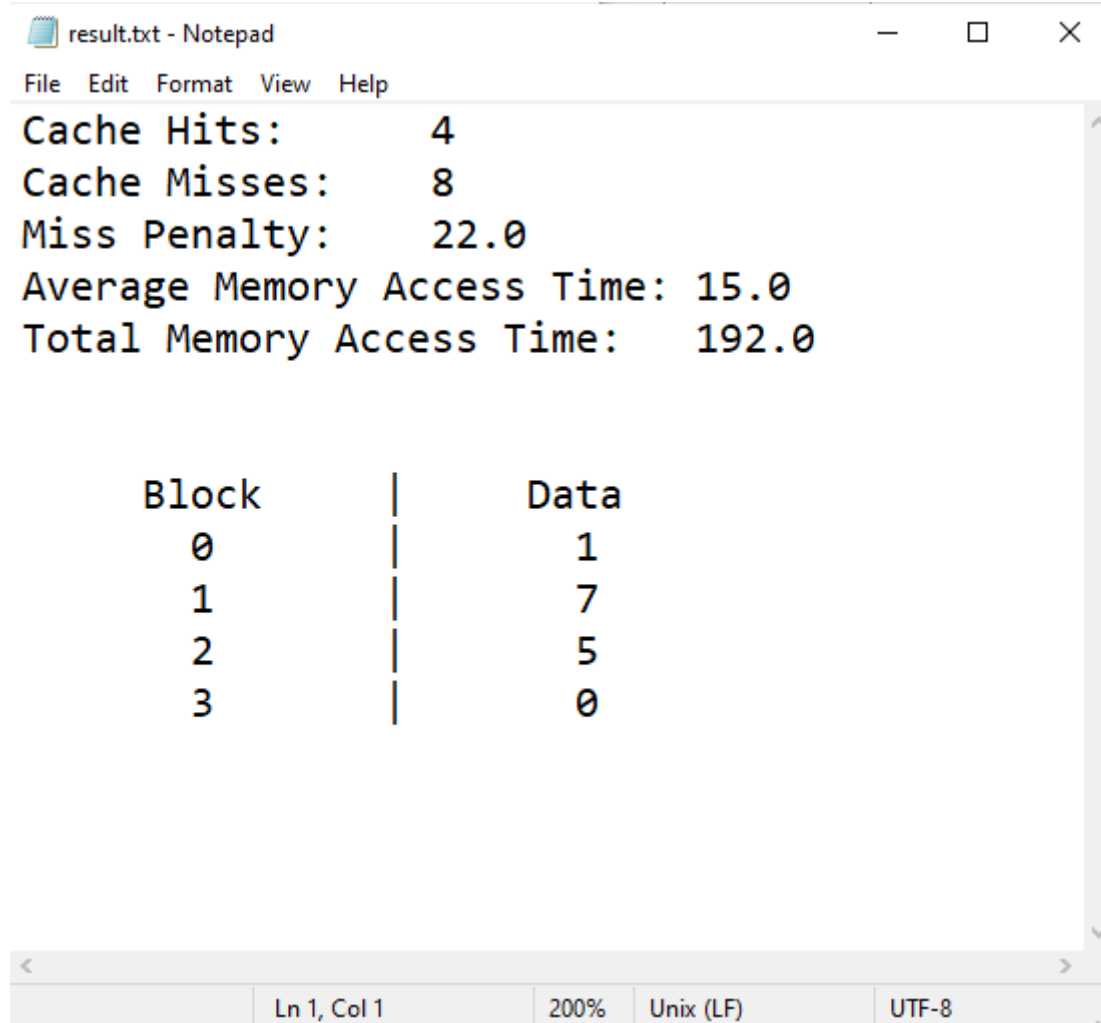
15.0

Total Memory Access Time

192.0

Export Results

4. By clicking the Export Results button, a text file containing the results will be exported.



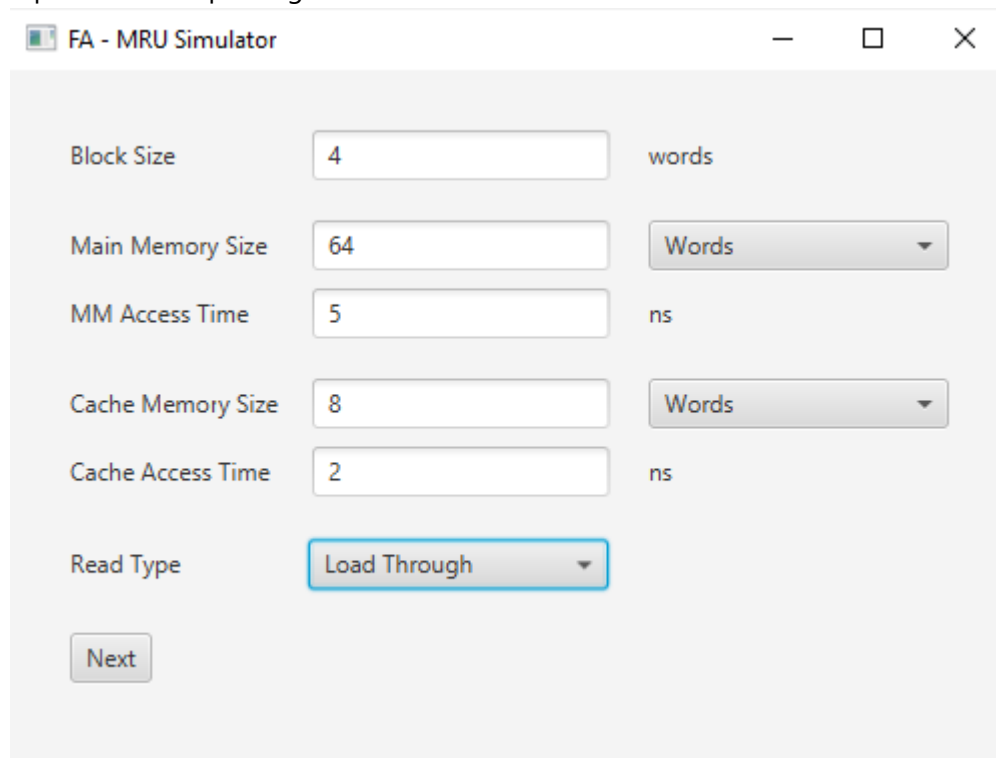
```
Cache Hits:      4
Cache Misses:    8
Miss Penalty:    22.0
Average Memory Access Time: 15.0
Total Memory Access Time: 192.0
```

Block	Data
0	1
1	7
2	5
3	0

Ln 1, Col 1 200% Unix (LF) UTF-8

B. Memory Size in Words

1. Inputs for the Input Page:



FA - MRU Simulator

Block Size	<input type="text" value="4"/>	words
Main Memory Size	<input type="text" value="64"/>	<input type="text" value="Words"/>
MM Access Time	<input type="text" value="5"/>	ns
Cache Memory Size	<input type="text" value="8"/>	<input type="text" value="Words"/>
Cache Access Time	<input type="text" value="2"/>	ns
Read Type	<input type="text" value="Load Through"/>	

2. Inputs for the Sequence Page:

FA - MRU Simulator

Number of Sequence Groups

2

Create

Blocks

Sequences		Repetitions
<div><div>+</div><div>-</div></div> Group		3
Sequence	3-5	1
Sequence	6,7	2
<div><div>+</div><div>-</div></div> Group		1
Sequence	9-11,15	3

Begin Simulate

3. Output:

FA - MRU Simulator

Block	Data
0	15
1	5

Cache Hits5

Cache Misses28

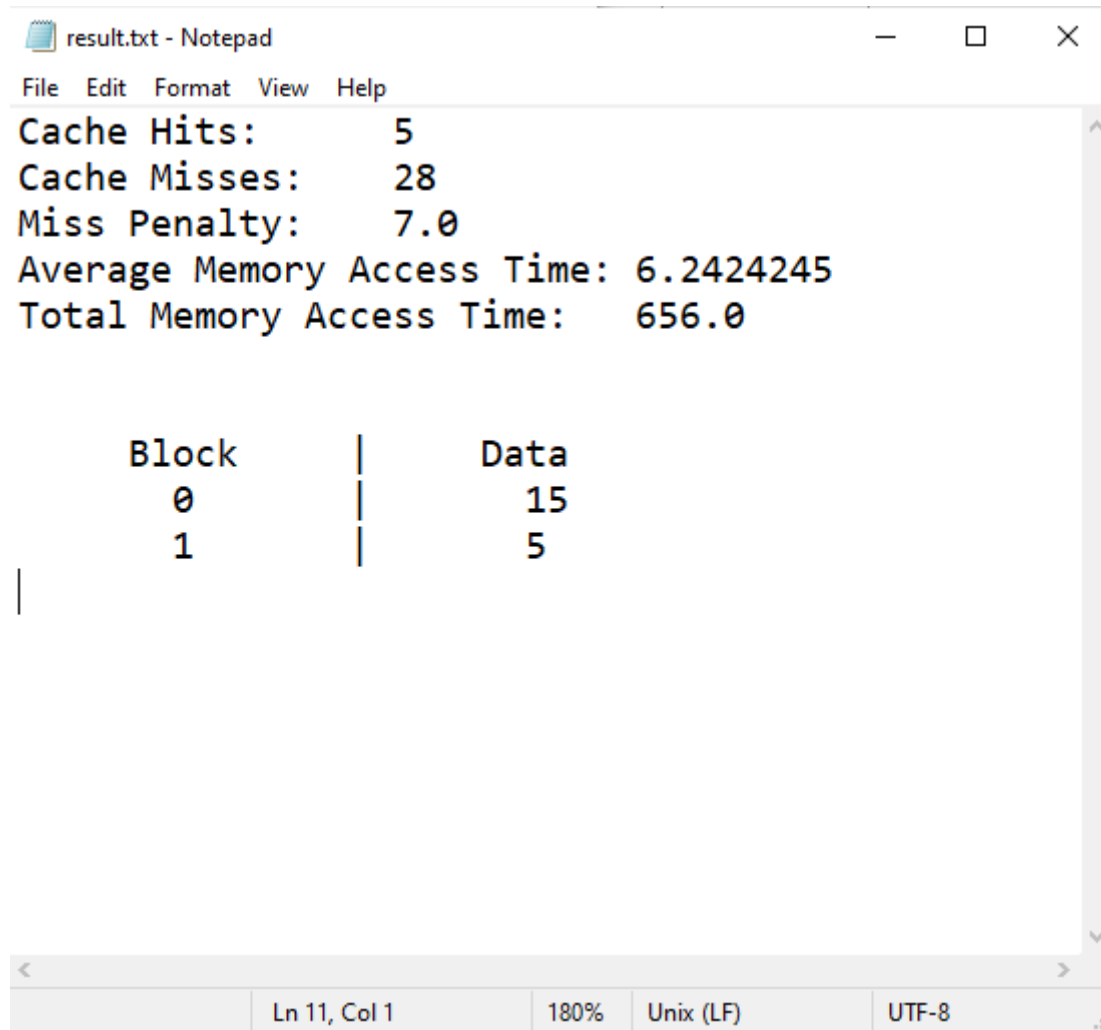
Miss Penalty7.0

Average Memory Access Time6.2424245

Total Memory Access Time656.0

Export Results

4. Contents of text file:



```
result.txt - Notepad
File Edit Format View Help
Cache Hits:      5
Cache Misses:    28
Miss Penalty:    7.0
Average Memory Access Time: 6.2424245
Total Memory Access Time:  656.0

      Block      |      Data
      0          |      15
      1          |      5
```

Ln 11, Col 1 180% Unix (LF) UTF-8

Authors

- [Bianca Joy Benedictos](#)
- [Kenneth Loquinte](#)
- [Regina Masilang](#)
- [Mikayla Tejada](#)