COMP 222 Computer Organization
Assignment #2—Cache Memory

Objective:

To simulate reading and writing to a custom-sized direct-mapped cache, involving a custom-sized main memory.

Inputs:

The total size of accessible main memory (in words)
The total size of the cache (in words)
The block size (words/block)
A signal to read (r) or write (w) to the cache
The main memory address to read from/write to
The contents of the address for writing to the cache

Outputs:

The corresponding cache tag, block, and word for a main memory address
The contents of the address resulting from reading/writing to the cache
A message indicating either a hit or a miss to the cache

Specification:

The program simulates reading from and writing to a cache based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:

a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit program

Notes:

- Use a structure (struct) to represent a cache line consisting of a tag (integer) and a block (integer pointer). Define the cache to be a pointer to the struct.
- Upon entering the parameters, the main memory and cache are to be dynamically allocated based on their respective total sizes. Each word *i* of main memory is initialized with the value *M*–*i*, where *M* is the size of main memory in words. For example, if the memory size is 16384, then word 10 will initially contain the value 16374 (which is 16384-10).
- Reading/writing from/to a new block in the cache results in dynamically allocating the block based on the block size.

What to turn in:

The source code as a single C file uploaded to Canvas (http://canvas.csun.edu) by the deadline (-20% per consecutive day for late submissions, up to the 4th day).

Sample test run

```
% ./a.out

Cache memory allocation and mapping:
-----------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: a
Enter main memory size (words): 65536
Enter cache size (words): 1024
Enter block size (words/block): 16


Cache memory allocation and mapping:
-----------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: b
Select read (r) or write (w): w
Enter main memory address to write to: 65535
Enter value to write: 14
Write miss!
-------------------------------------------
| Tag: 63   | Block: 63 | Word: 15 (14)   |
-------------------------------------------


Cache memory allocation and mapping:
-----------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: b
Select read (r) or write (w): r
Enter main memory address to read from: 65535
Read hit!
-------------------------------------------
| Tag: 63   | Block: 63 | Word: 15 (14)   |
-------------------------------------------


Cache memory allocation and mapping:
-----------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: b
Select read (r) or write (w): w
Enter main memory address to write to: 65534
Enter value to write: 512
Write hit!
-------------------------------------------
| Tag: 63   | Block: 63 | Word: 14 (512)  |
-------------------------------------------
```

```
Cache memory allocation and mapping:
-------------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: b
Select read (r) or write (w): r
Enter main memory address to read from: 1023
Read miss!
-----------------------------------------
| Tag: 0    | Block: 63 | Word: 15 (64513)|
-----------------------------------------


Cache memory allocation and mapping:
-------------------------------------
a) Enter parameters
b) Access cache for reading/writing and transfer data
c) Quit

Enter selection: c
%
```