

COSC 121
Project 6
Due April 23, 2018

The programs for this project may be written in any language of your choice (such as C++, C, Java, etc.).

The goal of this project is to simulate two different types of caches, with parameters specified below. You will write separate programs for each cache.

The program should take plain-text input in the form below

```
1 402A
0 4004
1 20F8
1 40F4
0 4025
1 4005
1 202A
-1
```

In the above input, each line represents one memory operation, with each line consisting of two fields. The first field being 0 represents a “memory load” operation, and 1 represents a “memory store” operation. The second field represents a 16-bit memory address in **hexadecimal** format.

-1 at the start of a line indicates that the input has terminated.

Simulate the following two caches:

- Cache 1: A direct-mapped cache consisting of 4 blocks, with each block consisting of 16 words.
- Cache 2: A fully associative cache consisting of 4 blocks, with each block consisting of 16 words. Use a **first-in-first-out** cache replacement policy, where the oldest cache block is replaced if necessary.

For each operation in the input, your program should print “hit” or “miss” depending on whether a hit or miss occurred.

After the entire input has been processed, the program should print the tag for each block in hexadecimal format, one block per line. For instance, in the output below, first number in each line is a block index, and second number represents a tag in hexadecimal format.

```
0 40A
1 2B0
2 325
3 401
```

Submit the following:

- Your programs for the two caches, named appropriately.
- Include a README file to briefly describe the contents of each file you submit.
- A short report describing which bits of the address are used as the tag for each cache.

You may be asked to demonstrate your code to the T.A., for the purpose of grading.