CENG204 - Computer Organization
Homework 4
Deadline: 23/05/2024 23:59 (Late Submissions are not allowed)

In this homework, you will simulate a direct map cache by filling ***LookforCache*** method (homework4.c). Delete the comment in the ***LookforCache*** method and develop your own code according to the given rules. It is better to read slides and understand the rules of E=2 associated cache.

```
int lookforCache(unsigned char memIndex, unsigned char *memory)

{

    // student code

    // You need to fill the cache according to the rules given in the document.

    return 0; //  Return 0 -> miss , Return 1 -> hit

}
```

Cache Map Policies:

- Cache and line structs are given. Do not change them.
- The number of set, the number of lines in a set and size for blocks are predefined and NOT going to change during the evaluation.
- memIndex is the index of the wanted data in the memory array.
  You need to separate this data to reach set number, tag and block offset.
- Memory is an unsigned char array. (You can assume that this is the main memory)
- myCache is our cache. It has 4 sets, 2 lines in each set and 8 blocks in each line.
  - Each block is 1 byte
  - You are going to store the values inside the memory array inside the blocks
  - Example outputs are given below.
- If all the lines are valid in a set and you need to remove one line, if the memory index is even you are going to remove the line in the left (line 0). If the memory index is odd, you are going to remove the line in the right (line 1).
- You need to return 1 for Cache Hit and return 0 for Cache Miss.

General Policies:

- You will do this homework alone.
- Your code will be tested with a different main method.
- Solution must be in a single C file `homework4.c` file but don't forget to write your name, surname and student number as a comment to the beginning of your file.
- Submit only `homework4.c` file.

```
(Example Output)

Cache at the begining

-------------------------------------------------------------------------------------------------------------

Set 0 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 1 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 2 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 3 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

-------------------------------------------------------------------------------------------------------------

Cache Miss For 9

-------------------------------------------------------------------------------------------------------------

Set 0 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 1 --> Line 1: v: 1 - tag: 0 - block: 146 63 228 77 122 108 100 53 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 2 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 3 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

-------------------------------------------------------------------------------------------------------------

Cache Hit For 12

-------------------------------------------------------------------------------------------------------------

Set 0 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 1 --> Line 1: v: 1 - tag: 0 - block: 146 63 228 77 122 108 100 53 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 2 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 3 --> Line 1: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

-------------------------------------------------------------------------------------------------------------

Cache Miss For 33

-------------------------------------------------------------------------------------------------------------

Set 0 --> Line 1: v: 1 - tag: 1 - block: 240 186 199 163 213 103 12 131 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 1 --> Line 1: v: 1 - tag: 1 - block: 146 63 228 77 122 108 100 53 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 2 --> Line 1: v: 0 - tag: 1 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |

Set 3 --> Line 1: v: 0 - tag: 1 - block: 0 0 0 0 0 0 0 0 | Line 2: v: 0 - tag: 0 - block: 0 0 0 0 0 0 0 0 |
```

Penalty for cheating:

- If cheating is detected **once**, students will **loose half of their all homeworks grades** (Hw1, Hw2, Hw3, Hw4 and Hw5).
- If cheating is detected **twice**, students **will loose the grades of all homeworks**.
- If cheating is detected more than two times, students will get FF and failed from the course.
- Detection of cheating **will not be announced** until at the end of the semester.

What is **cheating**?

- Sharing code: by copying, retyping, looking at, or supplying a file
- Describing: verbal description of code from one person to another.
- Coaching: helping your friend to write a homework, line by line
- Searching the Web for solutions
- Copying code from a previous course or online solution
- You are only allowed to use code we supply

What is **NOT cheating**?

- Explaining how to use systems or tools
- Helping others with high-level design issues

You can ask your questions via yusufsevkigunaydin@aybu.edu.tr with the mail subject "***About CENG204 Hw4***".