Notes

- The memory should be byte addressing. You can think about it as an array of bytes where each element is 8 bits/1 byte.
- A **compulsory miss** occurs when we access an address in a block and that block is not available in the cache.
- When we access an address in a block and that block is available in the cache, it is a hit.
- If you are loading a word then you get 4 bytes from the cache if the block is there. If the block is not there, then you should get the block from the memory and put it in the cache and then you read the 4 bytes of this block.
    - Example: LW R1, 100.
        1. You will first compute the cache address where address 100 is mapped onto. If its block is available then it hits. Otherwise it misses.
        2. If it hits, then you get the 4 bytes with memory addresses 100, 101, 102, 103 from the cache.
        3. If it misses, then you will get its corresponding block from the memory and put it in the cache. Then get addresses 100-103.
        4. Similarly the rest of the loads.
- For the L.S and L.D for the single and double precision floating point operation, they will be handled similarly to LW and LD for the integer operation.
- No need to handle the floating point numbers aka mantissa, exponent and so on. Handle it as you handle the integer values.
- The size of both the integer, floating point registers can be assumed to have the same size as a block of memory.
- For the branches BEQ or BNE, assume that they take two source registers and the address it will branch onto.
    - Example 1: BEQ R1, R2, 0
    - Example 2: BNE R1, R2, 0

Test Cases

Test Case 1: Sequential Code

```
L. D F6, 0
L. D F2, 4
MUL. D F0, F2, F4
SUB. D F8, F2, F6
DIV. D F10, F0, F6
ADD. D F6, F8, F2
S.D F6, 0
```

Test Case 2: Sequential Code

```
L. D F6, 0
ADD. D F7, F1, F3
L. D F2, 20
MUL. D F0, F2, F4
SUB. D F8, F2, F6
DIV. D F10, F0, F6
S.D F10, 0
```

Test Case 3: Loop Code

```
DADDI R1, R1, 24
DADDI R2, R2, 0
LOOP:L.D   F0, R1
     MUL.D F4, F0, F2
     S.D   F4, R1
     DSUBI R1, R1, 8
     BNE  R1, R2, LOOP
```

Where the word LOOP can be replaced with the address of the L.D instruction.