Quiz 2 micro

Exercise 1

Consider two caches C1 and C2. C1 is direct mapped and C2 is two-way set associative. Both caches are unified (instructions + data). Memory stalls are the only source of stalls in the system.

- The miss rate was found to be 6% in C1 while it's 5% in C2.
- The hit time is 1 clock cycle in C1, but due to the tag comparison approach employed in C2, the average hit time was found to be 10% longer than that of C1.
- The miss penalty is 48 ns in both caches.
- The load and store instructions constitute 20% of our codes.
- The clock rate is 0.5 Ghz

A) Compare C1 and C2 in terms of the average memory access time.

```
Avg mem access time = hit time + miss rate x miss penalty.

frequency is 0.5 GHZ which means clock cycle = 2 ns.
C1 mem access time  = 2 ns + 0.06 x 48 = 4.88 ns
C2 mem access time = 2 ns *1.1 + 0.05 x 48 = 4.6 ns
C2 is slightly faster.
```

B) Compare C1 and C2 in terms of execution time.

```
Execution time = IC x CPI x CCT

Penalty is given in ns = 48 ns which is equivalent to 24 cycles, since
CCT = 2ns.

CPI = ideal + memory stalls = ideal + misses/instruction  x penalty
 = 1 + (mem access/instruction x miss rate) x penalty
 = 1 + 1.2 * miss rate * 24

For C1: CPI_1 = 1+1.2*0.06*24 = 2.728
For C2: CPI_2 = 1+1.2*0.05*24 = 2.44
Since same IC and same CCT, C2 is also better.
```

C) How would the two caches perform against each other if they were enlarged enough so that we never miss in them?

```
Without misses: mem access time of C1 = 2 while C2 is 2.2 ns so C1
becomes better.
Execution time: they exhibit the same performance since CPI = 1 + 0
```

Exercise 2

Compare the number of misses when we use a *write-allocate* strategy versus a *no-write allocate* strategy, when executing the following pseudo-code, using a direct-mapped cache of size 128 bytes and block size 64 bytes. Cache starts empty.

```
# all numbers are block addresses.

WRITE block 9
READ  block 9
WRITE block 8
WRITE block 9
READ  block 8
READ  block 6
WRITE block 8
```

Solution:

Write allocate: 4 misses, 3 hits.

WRITE block 9 miss, bring block 9 to SET 1
READ block 9 hit
WRITE block 8 miss, bring block 8 to SET 0
WRITE block 9 hit
READ block 8 hit
READ block 6 miss, bring it to SET 0, remove block 8
READ block 8 miss

No- Write allocate: 6 misses and 1 hit

WRITE block 9 miss
READ block 9 miss, bring it to set 1
WRITE block 8 miss
WRITE block 9 hit
READ block 8 miss
READ block 6 miss, bring it to SET 0, remove block 8
READ block 8 miss