7. (6 points) A friend suggests adding a cache system to speed up the execution of MIPS assembly code on the micro-architecture from question 6 (slow load word (`lw`) instruction that requires 10 clock cycles for memory access).

   Which of the two cache design options would make the program in question 5 run faster? Briefly explain why.

   Solution A A direct-mapped cache with a capacity of 8 words and an access time of $t_{cache} = 1$ *clock cycle*

   Solution B A two-way set associative cache with the same capacity of 8 words, but with twice the access time $t_{cache} = 2$ *clock cycles*.

   Note: The question is specific to the program in question 5 and not about speeding up any arbitrary program.

   Solution:

   In a direct mapped cache, there is only one location where data can be placed in a cache. In a direct mapped cache with 8 locations, the address `0x0000 0004` and `0x0000 0024` would map to the same cache location. If this cache is used, every memory access would result in a cache miss. Using solution A would make the program run even slower, as every memory access would have first a cache miss (1 clock cycle) followed by the actual memory access (10 cycles).
   $N_{cycles} = 2 + (8 \times (5 + 2 \times 11)) = 218$

   Solution A can not be used to speed up the program.

   In a 2-way set associative memory, there are two locations where data can be placed in a cache. Although the addresses `0x0000 0004` and `0x0000 0024` still map to the same set, they can both be placed in the cache. The first two memory accesses would be compulsory misses requiring 12 clock cyles (2 for the cache miss plus 10 for the actual memory access). But for the following 7 iterations, every memory access would be a cache hit, requiring only 2 clock cycles.
   $N_{cycles} = 2 + (7 \times (5 + 2 \times 2)) + (5 + 2 \times 12) = 94$

   Solution B is the only viable alternative, that would speed up the program by more than 2x.

   Note that the program used is virtually identical to the one used in class notes to explain the problems with Direct-mapped cache.