My code uses hash tables as the given data structure, I used hash tables because it is more efficient than linked lists and is easy to understand code. The code takes 2 parameters and that is the value and associative that is given, I made a structure called cc and used that to make it store a data, value, and place so that it can store the tag index and validity of the cache. Essentially fifo , I used would take an missed address and it will shift the values to the right and make up room for the first and LRU changes and shifts if its hit

1. prefetcher allows for more change hits than misses, it increases memory reads as well, The prefetcher take sin more cache hits because it brings in data when it misses from nearby addresses. This is not very consistent and depends if the prefetcher hits or misses,.

2. We can add additional data structures such as linked lists so that a L2 cache could be added so that it can be used when we get a cache miss in L1 of the cache, we can essentially make many different data structure where we can link the 2 caches together and if L1 missed then we link it to L2
3. In order to not write the cache blocks back to memory we would have to edit the chase hits and not edit it back or write it into memory, you would also implement some sort of dirty bit that could be used to determine when to write back into memory