Q.1. Write an implementation LRU cache in python?

Ans :-

LRU (Least Recently Used) Cache discards the least recently used items first. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure the algorithm always discards the least recently used item. General implementations of this technique require keeping “age bits” for cache-lines and track the “Least Recently Used” cache-line based on age-bits.

The LRU caching scheme is to remove the least recently used frame when the cache is full and a new page is referenced which is not there in the cache. There are generally two terms use with LRU Cache.

Page hit: If the required page is found in the main memory then it is a page hit.

Page Fault: If the required page is not found in the main memory then page fault occurs.

When a page is referenced, the required page may be in the memory. If it is in the memory, we need to detach the node of the list and bring it to the front of the queue.

If the required page is not in memory, we bring that in memory. In simple words, we add a new node to the front of the queue and update the corresponding node address in the hash. If the queue is full, i.e. all the frames are full, we remove a node from the rear of the queue, and add the new node to the front of the queue.