1. Why does direct mapping not need any replacement algorithm?
   Direct mapping doesn't need a replacement algorithm because each memory block maps to a unique cache line, so there is only one possible location for any block. When a new block is loaded, it automatically replaces the existing data in that specific location.
2. Explain the cache replacement algorithms:
   a. Least recently used - LRU
      The Least Recently Used (LRU) algorithm keeps track of the order in which cache blocks are accessed. When a cache miss occurs, and there is no free space, LRU selects the block that has not been used for the longest time and replaces it with the new block. This approach assumes that recent data is more likely to be used again soon, making it efficient for many access patterns.
   b. Most recently used - MRU
      In the Most Recently Used (MRU) algorithm, the cache block that was accessed most recently is considered the least useful for future accesses. When a new block needs to be loaded into a full cache, MRU selects the block that was accessed last for replacement. This approach assumes that the most recently used data is less likely to be reused soon compared to other blocks, making MRU useful in certain situations, such as when frequently used data is accessed in quick bursts.
   c. First in First out - FIFO
      The First In, First Out (FIFO) algorithm replaces the cache block that has been in the cache the longest, regardless of how recently it was accessed. When the cache becomes full and a new block needs to be loaded, the block that entered the cache first is evicted. FIFO operates like a queue: the oldest block is at the front, and the newest block is added at the back.
3. A cache memory can hold 8 frames from RAM. Consider the cache is initially EMPTY. RAM has 128 frames of 1 byte each in a byte-addressable system. The processor frames in the following sequence: 127,8,0,127,3,5,7,9,6,3,8,0,5,11,19,8. Calculate the no. of cache misses for Fully associative cache mapping in
   a. LRU
   b. MRU
   c. FIFO

   Cache Size: 8 frames.
   RAM Size: 128 frames (1 byte each).
   Access Sequence: 127, 8, 0, 127, 3, 5, 7, 9, 6, 3, 8, 0, 5, 11, 19, 8.
   Cache Mapping: Fully associative (any block can enter any cache frame).

   Least recently used

   - 127: Miss (cache: [127])
   - 8: Miss (cache: [127, 8])
   - 0: Miss (cache: [127, 8, 0])
   - 127: Hit
   - 3: Miss (cache: [8, 0, 127, 3])
   - 5: Miss (cache: [0, 127, 3, 5])
   - 7: Miss (cache: [127, 3, 5, 7])
   - 9: Miss (cache: [3, 5, 7, 9])
   - 6: Miss (cache: [5, 7, 9, 6])
   - 3: Hit
   - 8: Miss (cache: [7, 9, 6, 3, 8])
   - 0: Miss (cache: [9, 6, 3, 8, 0])
   - 5: Miss (cache: [6, 3, 8, 0, 5])

- 11: Miss (cache: [3, 8, 0, 5, 11])
- 19: Miss (cache: [8, 0, 5, 11, 19])
- 8: Hit

LRU Cache Misses: 12 misses.

Most recently used

- 127: Miss (cache: [127])
- 8: Miss (cache: [127, 8])
- 0: Miss (cache: [127, 8, 0])
- 127: Hit
- 3: Miss (cache: [8, 0, 3])
- 5: Miss (cache: [0, 3, 5])
- 7: Miss (cache: [3, 5, 7])
- 9: Miss (cache: [5, 7, 9])
- 6: Miss (cache: [7, 9, 6])
- 3: Hit
- 8: Miss (cache: [9, 6, 3, 8])
- 0: Miss (cache: [6, 3, 8, 0])
- 5: Miss (cache: [3, 8, 0, 5])
- 11: Miss (cache: [8, 0, 5, 11])
- 19: Miss (cache: [0, 5, 11, 19])
- 8: Miss (cache: [5, 11, 19, 8])

MRU Cache Misses: 13 misses.

First in first out

- 127: Miss (cache: [127])
- 8: Miss (cache: [127, 8])
- 0: Miss (cache: [127, 8, 0])
- 127: Hit
- 3: Miss (cache: [8, 0, 127, 3])
- 5: Miss (cache: [0, 127, 3, 5])
- 7: Miss (cache: [127, 3, 5, 7])
- 9: Miss (cache: [3, 5, 7, 9])
- 6: Miss (cache: [5, 7, 9, 6])
- 3: Hit
- 8: Miss (cache: [7, 9, 6, 3, 8])
- 0: Miss (cache: [9, 6, 3, 8, 0])
- 5: Miss (cache: [6, 3, 8, 0, 5])
- 11: Miss (cache: [3, 8, 0, 5, 11])
- 19: Miss (cache: [8, 0, 5, 11, 19])
- 8: Hit

FIFO Cache Misses: 12 misses.