1. What are the different levels of memory? List the pros and cons of each level.

   Registers
   - Pros: Fastest, directly accessed by CPU.
   - Cons: Very small, expensive.

   Cache (L1, L2, L3)
   - Pros: Fast, improves access speed.
   - Cons: Limited size, costly.

   RAM
   - Pros: Fast, large capacity (GBs).
   - Cons: Volatile, slower than cache.

   Secondary Storage (SSD/HDD)
   - Pros: Non-volatile, large capacity.
   - Cons: Slower than RAM, higher latency.

   Tertiary Storage (Optical/Cloud)
   - Pros: Cheap, massive capacity.
   - Cons: Very slow, not easily accessible.

   Virtual Memory
   - Pros: Expands RAM using disk.
   - Cons: Slower, performance drops when overused.

2. What is a memory-addressing technique? Explain the various addressing modes.

   Immediate Addressing
   - Data is part of the instruction.
   - Pros: Fast access, simple.
   - Cons: Limited data size.

   Direct Addressing
   - The address of data is specified directly in the instruction.
   - Pros: Simple, and efficient for small programs.
   - Cons: Limited address range.

   Indirect Addressing
   - Address field points to a memory location that contains the actual address of data.
   - Pros: Can access large memory.
   - Cons: Slower due to extra memory access.

   Register Addressing
   - Data is stored in a CPU register.
   - Pros: Very fast.
   - Cons: Limited register availability.

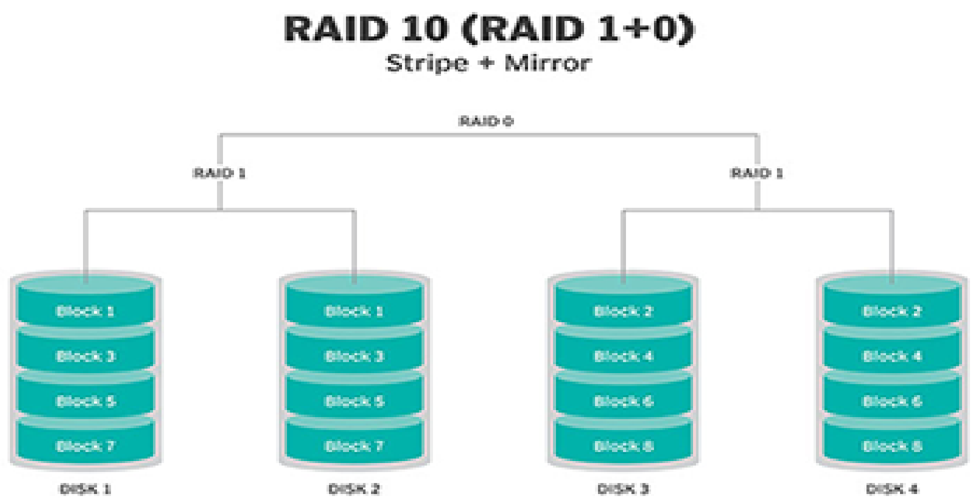
   Indexed Addressing
   - Base address is added to an index (usually stored in a register).
   - Pros: Good for array access.
   - Cons: More complex.

   Base-Relative Addressing
   - Address is computed by adding a constant (offset) to a base register.
   - Pros: Efficient for relocatable code.

- Cons: Slightly slower due to calculation.

3. Address bus consists of 20 bits in a byte-addressable system. Find the size of the memory.
   2^{20} bytes = 1MB of memory
4. Memory size is 8GB in a 4-byte addressable system. Find the number of address bits.
   31 bits
5. What is a RAID system? Explain with a diagram.
   A RAID system (Redundant Array of Independent Disks) uses multiple SSDs or HDDs to enhance data storage:
   - RAID 0: Stripes data across drives for improved performance, but no redundancy.
   - RAID 1: Mirrors data on two drives for redundancy but doesn't improve performance.
   - RAID 5: Uses striping with distributed parity for fault tolerance and improved performance.
   - RAID 6: Similar to RAID 5 but with double parity, allowing recovery from two drive failures.
   - RAID 10: Combines RAID 0 and RAID 1, offering both performance and redundancy.

   RAID improves speed, reliability, and data protection depending on the configuration.



6. Explain virtual memory with an example. Why do we need them?
   Virtual Memory extends RAM using disk space to handle more data. For example, if you have 8 GB of RAM and run a 12 GB program, virtual memory uses disk storage to accommodate the extra 4 GB.
   Why We Need It:
   - Allows more applications to run simultaneously.
   - Enables programs to use more memory than available RAM.
   - Provides process isolation and improves multitasking.

7. What is the wait state in memory? How can we deal with with state?
   A wait state occurs in a computer system when the CPU has to pause or wait because it can't access the memory or I/O device quickly enough. This usually happens due to slower memory or peripherals compared to the CPU's speed.
   - Increase Memory Speed: Use faster memory chips to reduce wait times.

- Cache Memory: Implement cache memory (L1, L2, L3) to store frequently accessed data and reduce wait states.
- Memory Interleaving: Distribute memory access across multiple modules to improve throughput and reduce wait times.
- Pipelining: Use pipelining techniques to overlap memory access and computation, reducing the impact of wait states.
- Bus Speed Upgrade: Increase the speed of the system bus to improve data transfer rates between the CPU and memory.
- Buffering and Prefetching: Use buffers and prefetch techniques to load data ahead of time, minimizing delays during data access.