Given a 64-bit x86_64 Architecture where the page size is set at 4096, the number of physical address size in memory is 2^{46}, and the virtual memory address size is 2^{48}, derive the number of pages in the disk and draw the representation of the Virtual-to-Physical address translation.

For a single level Page Table with 32-Bit machine with 4-KB pages, define the size of a single Page Table size. Then, what is the smallest amount of page table data we need to keep in memory for n-32 bit applications in a two-level Hierarchical Page Table? And how many applications can be implemented in this two-level Hierarchical Page Table in the equivalent memory space?

Given the following parameters for memory

- L1 cache hit = 1 clock cycles, hit 95% of accesses
- L2 cache hit = 10 clock cycles, hit 60% of L1 misses
- DRAM = 200 clock cycles (≈100 nanoseconds)
- Disk = 20,000,000 clock cycles (≈10 milliseconds)

a) Derive the Average Memory Access Time without paging

b) Derive the Average Memory Access Time with paging for a Memory Hit Rate of 99%?

c) Derive the Average Memory Access Time with paging for a Memory Hit Rate of 99.9999%?