1. Raspberry Pi B+ components:
   1. quad-core ARM CPU
   2. 1GB RAM
   3. Ethernet connector and controller
   4. 4 USB 2.0 ports
   5. GPIO pins
   6. HDMI port
   7. Camera and Display interfaces
   8. microSD slot
   9. Micro USB slot for power
2. The Raspberry Pi B+ contains four cores
3. Three main differences between x86 (Complex Instruction Set Computer) and ARM (Reduced Instruction Set Computer) architectures;
   1. CISC executes “complex instructions” with hardware logic capable of understanding and executing multiple instructions. It mimics a high-level language by being able to understand multiple processes from a single instruction (ex: a = a + b). This allows code to use less RAM, and a compiler has to do less work when translating high-level code to assembly.
   2. RISC executes simple instructions (explicit load-store) that are executed in one clock cycle. This means more code has to be written both by the compiler and when coding in assembly itself. It also uses more RAM since there’s more instruction to be stored. Also ARM produces less heat and uses less power when compared to x86.
4. Difference between sequential and parallel computation:
   1. A single core only operates on one process at a time; having more cores means the OS can execute more processes concurrently. The OS sees the cores as separate and schedules threads for each one when available.
5. Identify the basic form of data and task parallelism in computational problems.
   1. Task parallelism: carrying out multiple functions across the same or different datasets simultaneously on multiple cores
   2. Data parallelism: carrying out a single function across the elements of a dataset on multiple cores.
6. A process is the abstraction of a running program and does not share memory with other processes. Threads are lightweight processes that allows a single process to be broken into smaller parts. This allows multi-core computers to run several threads at once completing a process in less time.
7. OpenMP is a multi-core assembly language that has been industry standard since the late 1990s. OpenMP pragmas are compiler directives allowing the compiler to generate threaded code implicitly without the threads having to be hard-coded by the programmer.
8. Database servers, Web servers, Multimedia applications, Compilers
9. Why Multicore?
   1. Difficult to increase clock frequencies of single-core CPUs
   2. Most new applications are multithreaded allowing for the utilization of multicore
   3. Current trend in computer architecture points towards more parallelism
   4. Today’s most-used Operating Systems (Mac OS, Windows, Linux) are multicore.