CS503 Spring 2019
Quiz 1 (10 pts)

Answer the questions in the spaces provided. If you run out of room, continue on the back.
Provide clear answers and justify statements where required.
Unless otherwise noted, assume that questions refer to C code, AT&T assembly syntax and the x86 architecture.

1. (2 points) Name:

2. (6 points) State whether the following statements are true of false. **Explain the false statements.**
   (a) (1 point) The C statement "`char *i;`" is valid and declares a pointer to an integer.
   - ◯ True
   - ◯ False. Explanation:

   Solution: False. It declares a pointer to a char (or memory location).

   (b) (1 point) The assembly statement "`pop %eax`" moves the value of the EAX register to an implicit register.
   - ◯ True
   - ◯ False. Explanation:

   Solution: False. Read values from stack and puts it in the EAX register. (No need to discuss the stack pointer.)

   (c) (1 point) One of the roles of the operating system is to be a referee.
   - ◯ True
   - ◯ False. Explanation:

   Solution: True.

   (d) (1 point) The stack grows "downwards"
   - ◯ True
   - ◯ False. Explanation:

   Solution: True.

   (e) (1 point) Dynamically linked libraries are linked with applications during run-time.
   - ◯ True
   - ◯ False. Explanation:

   Solution: True.

(f) (1 point) There is no difference between the BSS section and the data section in a binary.

◯ True

◯ False. Explanation:

Solution: False. BSS stores uninitialized (or zeroed) data and the data section stores initialized data.

3. (2 points) Explain at a high-level the concept of privileged instructions and *CPU ring level* in the x86 architecture and how they can be used by kernel developers.

Solution: Privileged instructions configure the system and can interfere with the normal behavior of the system (i.e., any application) and thus are generally only executable by kernel code. (+1 point) The CPU ring levels specify different modes of CPU execution that have different levels of access to privileged instructions. (+1 point) Not required: In OSs with protection, user programs generally run in ring 3 and kernel code in ring 0.

4. (2 points (bonus)) In a C program the first function to run is the main() function but the compiled binary will also contain a *_start* routine. Provide a high-level description of how this routine is generated and at least two tasks that it performs.

Solution: The routine is generated by the C compiler and static linker (+1 point). Tasks: call main(), set up the arguments, set up the environment variables, initialize C libraries. (Each +0.5 points, up to +1 point for two correct).