Problem 1 (Assembly inlining - 30 points): Write a C function that adds integers between x and y, \( y \geq x \), using assembly inlining using the following format.

```c
/* returns x + (x+1) + (x+2) + ... + y */
int asm_sum(int x, int y) {
    ...
    __asm__ (    ...
    ...    )
}
```
Problem 2 (Module programming - 30 points): Write a module program that allows you to pass command line parameters to the kernel module. When you insmod your_mod.ko name="linux_kernel" number="323" street="central" city="newark" state="nj" zip="07102", your /var/log/messages would have

    cs433: name: linux_kernel
    cs433: number: 323
    cs433: street: central
    cs433: city: newark
    cs433: state: nj
    cs433: zip: 07102
Problem 3 (turn-on to start_kernel() - 20 points): Explain what happens (a) from the moment you turn on your Linux box until Grub finishes loading the kernel image vmlinux and (b) after Grub loads a kernel image till start_kernel() begins. Your answer must include the following steps in a correct sequence: Grub, POST, MBR, BIOS, CMOS, compressed and decompressed kernels. Be specific and concise. Write your answers in functions and the files that contain the functions. Writing a lot won’t necessarily help.
Problem 4 (20 points, the first three kernel threads):
What is the name, not number, of the very first kernel thread?

What is the name of the function that calls two functions of the same name but with different parameters to create the first two kernel threads?

What are the process numbers and names of the two kernel threads created by the very first kernel thread?

After the two kernel threads described above are created, what happens to the very first kernel thread and where?