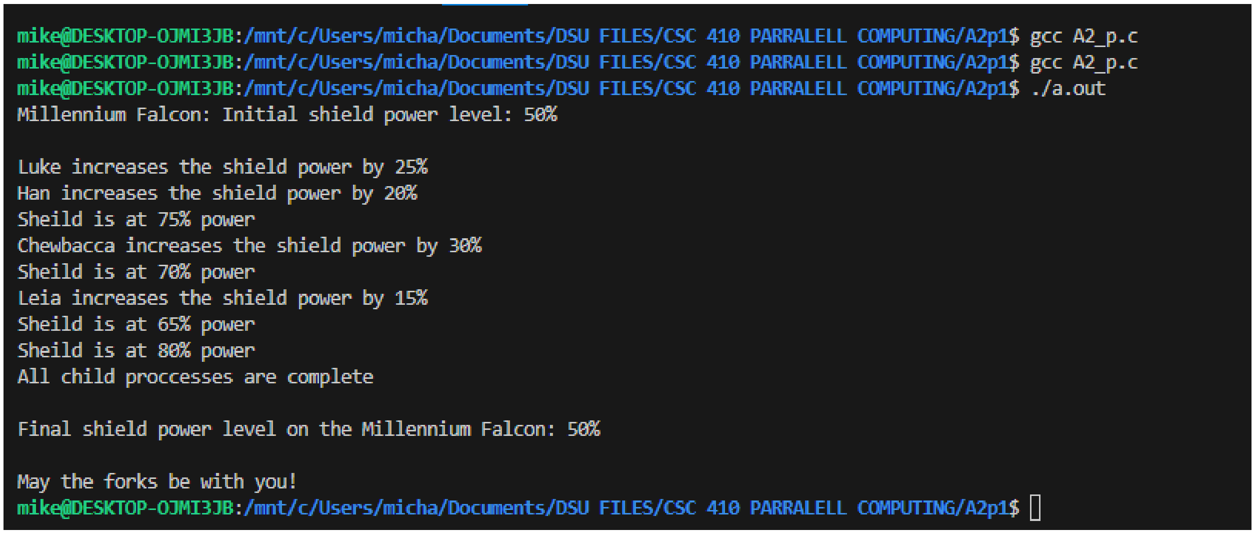

```
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p1$ gcc A2_p.c
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p1$ gcc A2_p.c
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p1$ ./a.out
Millennium Falcon: Initial shield power level: 50%

Luke increases the shield power by 25%
Han increases the shield power by 20%
Sheild is at 75% power
Chewbacca increases the shield power by 30%
Sheild is at 70% power
Leia increases the shield power by 15%
Sheild is at 65% power
Sheild is at 80% power
All child proccesses are complete

Final shield power level on the Millennium Falcon: 50%

May the forks be with you!
mike@DESKTOP-OJMI3JB:/mnt/c/Users/micha/Documents/DSU FILES/CSC 410 PARRALELL COMPUTING/A2p1$
```

Task 2:

1) The initial and final value of shield power remains unchanged because the child processes have their own copy of memory so because the parent process isn't changing the copy that gets printed, it remains unchanged.
2) The order changes based on how the child processes get scheduled by the CPU

Task 3:

Because memory is not shared, what you could do was break apart inventory into smaller chunks and let each child end when their allotted portion of inventory is exhausted. Then you could respawn the child process with a new chunk of inventory and have the parent process track how much is allotted to each child.