

1.

- a. The output starts with the parent id and prints the id and then prints the kid id and when it finishes printing the kid id then it prints the final statement no more kid!
- b. The process id numbers of the threads are the same. This is the case because the pthread create function just creates a new thread not a new process. Therefore, they will share the same ID.
- c. The parent thread finishes last because it waits for the kid thread to be finished before printing the no more kid print statement.

2.

- a. See threads.c file
- b. The output prints 0's and "."s through the threading process and it should be printed every other but sometimes the 0 can be printed twice and the "." can be printed twice in a row. Then it prints the total value of my global which always equals 21. The order in which 0 and "." are randomized depending on which thread takes priority in that moment.
- c. If the pthread_join function is removed this causes the main function to not wait for the threads to finish. This may cause the program to run the main function before the threads are finished so it won't complete the 21 0's and "."s