The program `pth_sem_bar.c` attempts to implement a barrier using two semaphores. However, it fails if a program uses the barrier more than once. One way to fix this is to replace the single semaphore, `barrier_sem`, with `thread_count` semaphores: one for each thread.

With these additional semaphores, each thread (except the last to enter the barrier) can block on its `barrier_sem` — e.g., `barrier_sem[my_rank]`. Then the last thread to enter the barrier can “signal” each thread by posting its semaphore:

```c
for (j = 0; j < thread_count; j++)
    if (j != my_rank) sem_post(&barrier_sem[j]);
```

For assignment 12 you should modify `pth_sem_bar.c` so that it uses this approach. Your documentation should include a brief discussion of the new approach: does it work with many threads? many barriers? many threads and many barriers?