Additional questions:

1. Would it be sufficient just to add to the old algorithm from task5 additional mutex variable to organize critical sections in functions grab\_forks() and put\_away\_forks() for making changes to values of two mutexes indivisably?  If not, why?

Answer:

In our lab5 task is sleeping barber task, it is similar to philosopher task, I think if we just add an additional mutex variable to organize critical section it will be not sufficient, cause the mutex cannot change more than one mutex value per one call.

1. Why m mutex is initialized with 1 and mutexes from the array s are initialized with 0's?

Mutex m is initialized with 1 because we just want to change or compare philosopher state by themselves thread rather than other thread to check or change it when we enter a critical section. The other hands because we want let thread accessible and active since very beginning after initialization.

Array s [] is initialized with 0 because we try to avoid leading deadlock, the shared memory will be locked, until it is necessary at proper moment to let access them by expecting them process.