1. As partes em **negrito** devem constar no relatório
2. Faça *download* do [arquivo](https://drive.google.com/file/d/0B4HCJCgCTQHYSlR5aVZfUVdTQWM/edit?usp=sharing) para a prática
3. Descompacte o arquivo
4. Observe o programa joinEx.c. Para entender o que ele faz, procure na documentação [API da biblioteca pthreads](https://computing.llnl.gov/tutorials/pthreads/#CreatingThreads) os tipos e as funções que você não conhece:
   * pthread\_t
   * pthread\_attr\_t
   * pthread\_attr\_init
   * pthread\_attr\_setdetachstate
   * pthread\_create
   * pthread\_attr\_destroy
   * pthread\_join
5. **Dê uma descrição do que faz cada um desses tipos ou funções.**
6. **Compile (gcc -o joinEx -pthread joinEx.c) e rode o programa. O que ele faz? Explique cada linha do resultado da saída do programa.**
7. Observe o programa simpleMutex.c. Para entender o que ele faz, procure na documentação [API da biblioteca pthreads](https://computing.llnl.gov/tutorials/pthreads/#CreatingThreads) os tipos e as funções que você não conhece:
   * pthread\_mutex\_t
   * pthread\_mutex\_lock
   * pthread\_mutex\_unlock
8. **Dê uma descrição do que faz cada um desses tipos ou funções.**
9. **Rode o programa. O que ele faz?**
10. **Retire as diretivas de sincronização "lock" e "unlock". O que acontece com o programa? Você saberia explicar esse comportamento?**
11. Observe o programa prodconsUnsync.c, que é uma implementação do problema do produtor/consumidor. Procure entender o código. Ele se comporta de maneira adequada? Responda às seguintes perguntas para avaliar o seu entendimento a respeito do programa:
    * **Qual o tamanho do *buffer*?**
    * **Pra que serve a variável *writeable* (dentro do buffer)?**
    * **Se o programa funcionasse corretamente, qual seria sua saída no final?**
12. Adicione algum código usando join() e exit(), como em joinEx.c para que o programa termine de maneira adequada (sem precisar de ctrl+C). Melhore o comportamento do programa usando mutexes. Siga as seguintes dicas:
    * Lembre-se que você deve usar mutexes tanto para garantir exclusão mútua quanto para evitar condições de *buffer* cheio ou vazio.
    * Use a variável *writeable* para garantir as condições de escrita no *buffer*.
    * As regiões críticas estão dentro das funções *store* e *retrieve*. É dentro delas que você deve adicionar o código dos mutexes.
    * **É muito provável que, após corrigir seu código, às vezes no terminal apareça que um número foi retirado antes de ser armazenado. Investigue e explique por que isso acontece.**
13. **Adicione seu código corrigido ao relatório.**