

## Задание 1.

Написать программу, выполняющую умножение матриц. Преподавателем выдается 2 файла – 2 большие матрицы из чисел типа `double`. Матрицы гарантированно являются квадратными – размер матрицы можно (и нужно) вычислить из размера файла. Демонстрацию корректности работы алгоритма проводить для матриц размера (10x10) с выводом результата на экран. Время работы замерять для выданных преподавателем матриц. Данную задачу решить без использования отдельных примитивов синхронизации.

## Задание 2. (средний уровень)

Написать программу, выполняющую поиск элементов с заданным значением в массиве элементов типа `int`. Результатом являются индексы всех элементов с заданным значением в порядке X. В качестве примитива синхронизации использовать Y. Помните про то, что в STL нет потокобезопасных коллекций, но с помощью блокировок с ними можно работать потокобезопасно.

Вариант	X	Y
3	возрастания	барьер

Команда запуска:

```
g++ -std=c++20 -o task2 task2.cpp -pthread
```

## Задание 3 (сложный уровень).

Напишите шаблон класса потокобезопасной очереди сообщений `threadsafe_queue`. Примитивы синхронизации – мьютекс и условная переменная. Полезно вспомнить про блокировку с двойной проверкой. Продемонстрировать работу с M потоками-писателями и N потоками-обработчиками. Способ демонстрации – на усмотрение студента. Самый простой способ – писатели пишут в очередь числа, читатели распечатывают числа из очереди (при этом порядок распечатанных элементов может отличаться от исходного, но сама очередь может работать корректно – если такое наблюдается, обосновать). Помните, что необходимо уведомлять потоки-читатели о том, что новых данных больше не будет. Данная задача может решаться и вне класса `threadsafe_queue` (отдельная переменная, сигнал, сообщение со специальным значением и пр.). Итоговый способ реализации также остается на усмотрение студента.