1. Написать метод, который производит умножение матриц (*даёт 2.0 балла*).

**Вход**:

1. A:double[,], B:double[,] — перемножаемые матрицы
2. m:int, n:int, k:int — размеры матриц (m×n, n×k);

**Выход:** double[,] — произведение.

1. Написать методы для приближённого нахождения максимума и минимума *производной* от заданной функции *f*(*x*) на *n* равномерно распределённых узлах из заданного интервала [*a*, *b*] (*даёт 3.0 балла*).

**Вход**:

1. f:(double → double) — исследуемая функция;
2. a:double, b:double — границы интервала для исследования;
3. n:double — количество узлов.

**Выход:** double — найденный максимум/минимум.

**Дополнительные требования**: поиски максимума и минимума должны находиться внутри директив **section**. Внутри обеих секций должно работать *более одного* потока одновременно (*nested*).

1. Написать программу для численного вычисления интеграла заданной функции *f* (*x*) на интервале [*a*, *b*] по квадратурному интерполяционному правилу Ньютона – Котеса на подынтервалах. (*даёт 3.0 балла*)

**Вход:**

1. f:(double → double) — подынтегральная функция;
2. a:double, b:double — начало и конец интервала интегрирования;
3. m:double — количество подынтервалов;

**Выход**: double — значение интеграла.

**Дополнительные требования**: используйте директивы for и reduce.

1. \*Напишите программу, которая решает систему линейных уравнений методом Гаусса. (*даёт 2.0 балла*)

**Вход:**

1. A:double[,] — матрица коэффициентов;
2. n:int — количество переменных;
3. b:double[ ] — вектор значений

**Выход**: double[] — вектор-решение.