МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова»

Факультет:
Кафедра
Направление

ОТЧЕТ
по лабораторным работам

по дисциплине: Структуры данных

Выполнил студент 1 курса, группы ПоВа-о24 Аканжи Аалия Омотайо

Фамилия, имя, отчество

Принял

Должность, звание Фамилия, имя, отчество

Работа принята «__06__»____2025_____ г. _____ Подпись

Новочеркасск 2025 г.

Задача;

Перемножить 2 квадратные матрицы размера 1024x1024 с элементами типа single complex (комплексное число одинарной точности).

Исходные матрицы генерируются в программе (случайным образом либо по определенной формуле) либо считываются из заранее подготовленного файла.

Оценить сложность алгоритма по формуле c = 2 n_3, где n - размерность матрицы.

Оценить производительность в MFlops, p = c/t*10_{-6}, где t - время в секундах работы алгоритма.

Выполнить 3 варианта перемножения и их анализ и сравнение:

1-й вариант перемножения - по формуле из линейной алгебры.

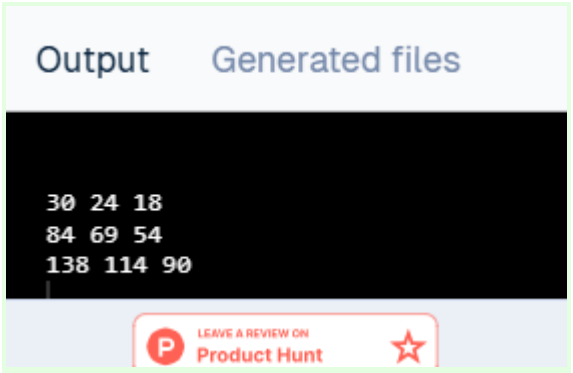
2-й вариант перемножения - результат работы функции cblas_cgemm из библиотеки BLAS (рекомендуемая реализация из Intel MKL)

3-й вариант перемножения - оптимизированный алгоритм по вашему выбору, написанный вами, производительность должна быть не ниже 30% от 2-го варианта

Код 1:

```
#include <iostream>


void naive_multiply(const double* A, const double* B, double* C, int n) {
   for (int i = 0; i < n; ++i) {
      for (int k = 0; k < n; ++k) {
         double tmp = A[i*n + k];
         for (int j = 0; j < n; ++j) {
            C[i*n + j] += tmp * B[k*n + j];
         }
      }
   }
}
```

```
int main() {
    const int n = 3;  // Example size
    double A[n*n] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    double B[n*n] = {9, 8, 7, 6, 5, 4, 3, 2, 1};
    double C[n*n] = {0};  // Initialize result matrix to zeros

    naive_multiply(A, B, C, n);

    // Print result
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cout << C[i*n + j] << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}
```



Код 2:

```
#include <cstddef>
#include <iostream>

void matrix_multiply(const double* A, const double* B, double* C,
int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            C[i*n + j] = 0;
            for (int k = 0; k < n; k++) {
                C[i*n + j] += A[i*n + k] * B[k*n + j];
            }
        }
    }
}

int main() {
    // Example usage of matrix_multiply
    const int n = 2;
    double A[] = {1, 2, 3, 4};
    double B[] = {5, 6, 7, 8};
    double C[n*n];

    matrix_multiply(A, B, C, n);

    // Print the result
    for (int i = 0; i < n; i++) {
```

```
        for (int j = 0; j < n; j++) {
            std::cout << C[i*n + j] << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}
```