

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Естественнонаучный институт

Механико-математический факультет

Отчёт

Лабораторная работа №1

«Высокопроизводительные вычисления в механике»

Выполнил: Амиров Э.А.

Группа 4446-010303D   
 Проверил: Неженский М.С.

      «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

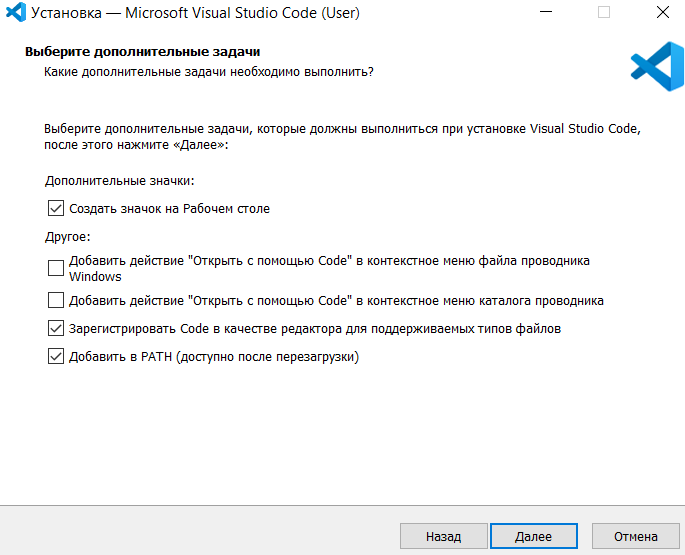
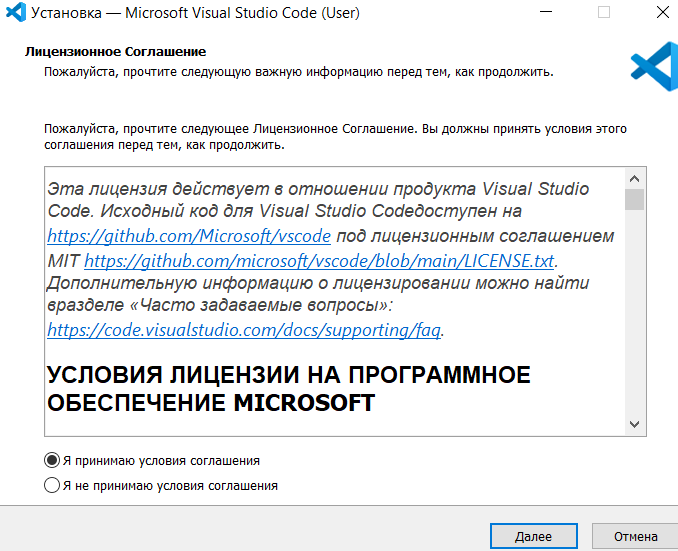
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2025

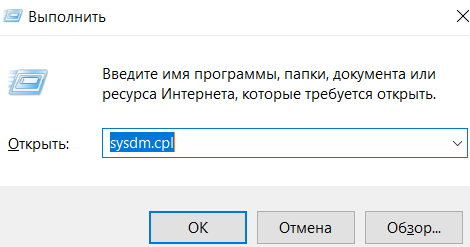
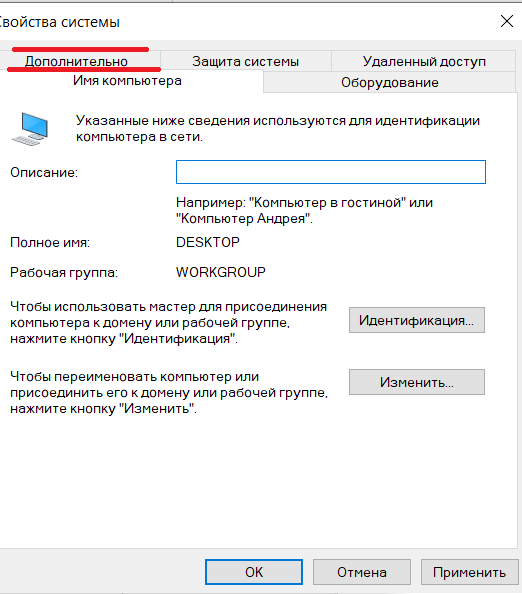
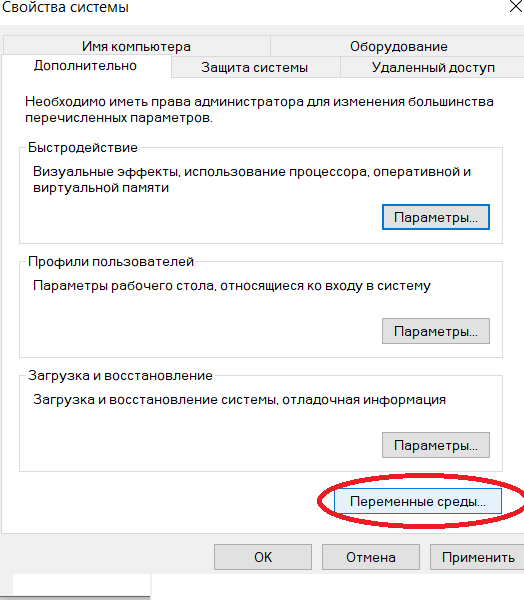
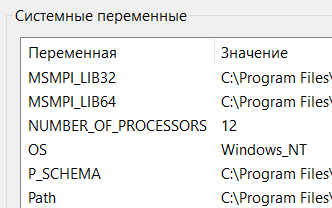
# Задание

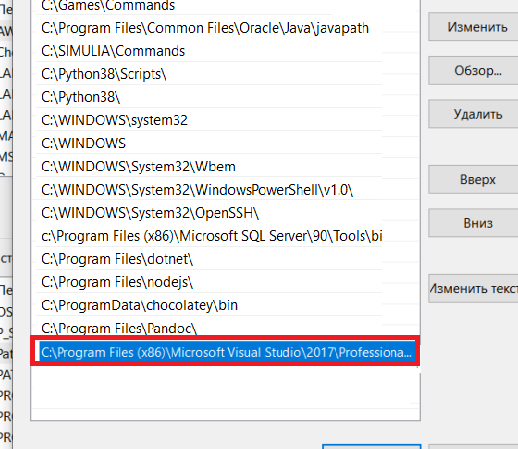
1. Исследовать программу на языке C/C++ для перемножения двух матриц.
2. matMulSeq.cpp – исходники прилагаются  
   **Порядок выполнения л/р:**

В данном задании сделать всю необходимую подготовку к работе с C/C++ со скринами, включая изменение системных переменных среды.  
  
Ход работы:

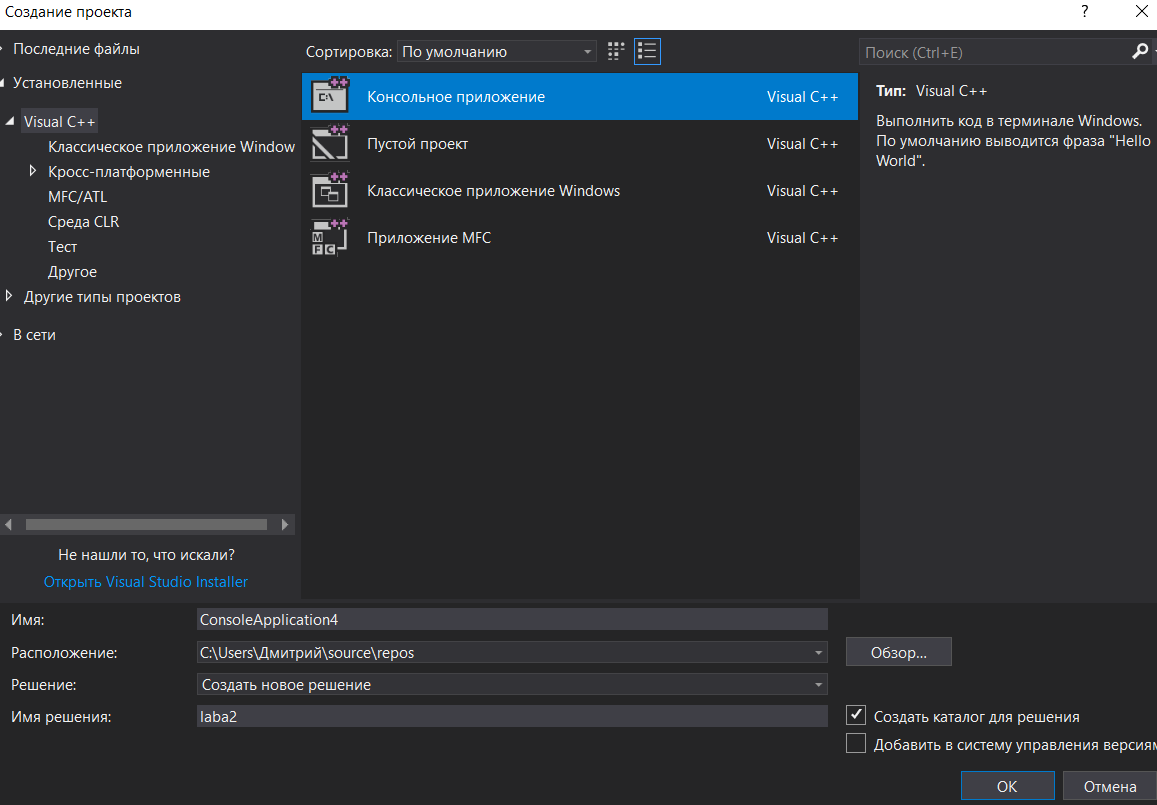
Установка visualstudio <https://code.visualstudio.com/>  
Добавление путей компилятора и библиотек в PATH

Открываем редактор переменных среды:

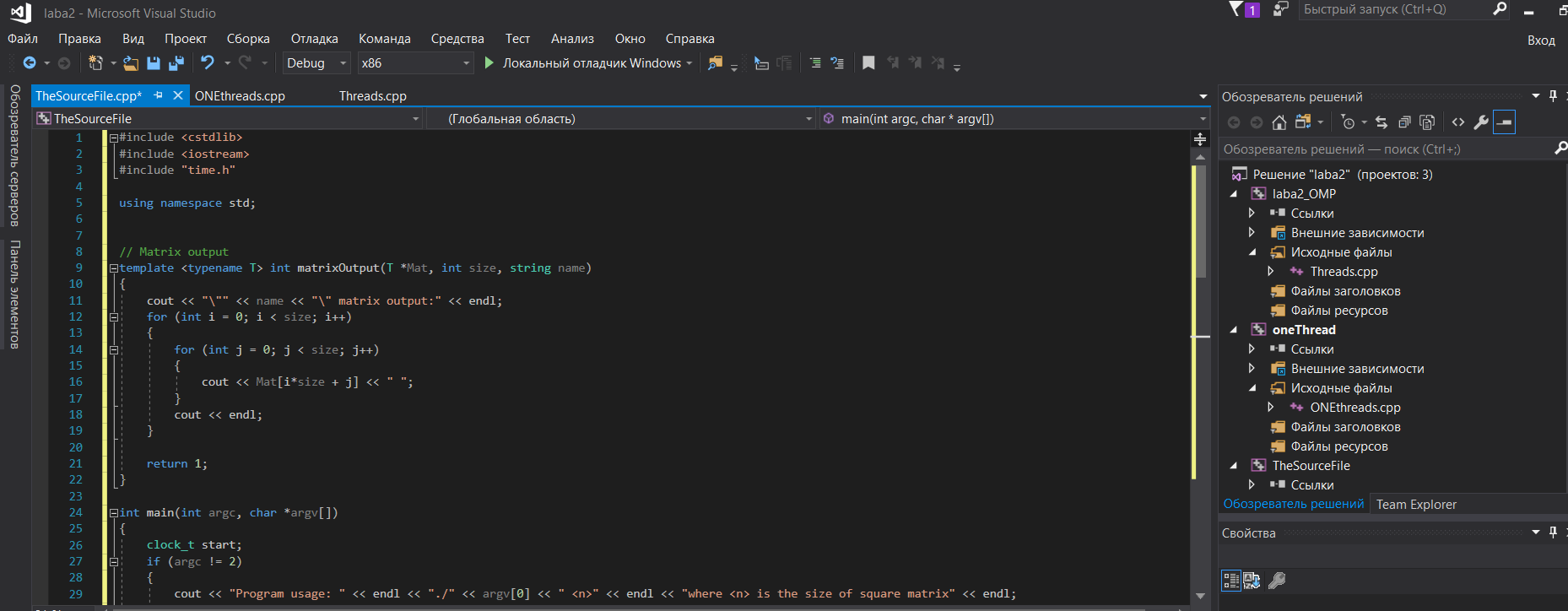
Нажимаем Win + R, вводим sysdm.cp  
  
 → вкладка Дополнительно  
  
 → Переменные среды.  
  
В разделе Системные переменные находим Path → Изменить → Создать.  




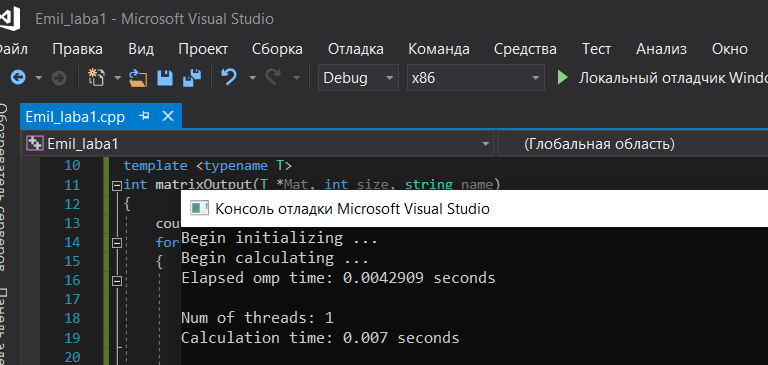
Для начала работы создаём проект в Microsort Visual Studio 2019 Professional:



Переносим код matMulSeq для перемножения матриц



Запускаем программу с размерностью матриц N=100, получаем результат следующий:



Всё успешно расчитано. Сам код и его описание расписаны в приложении 1.

# Создаем новый репозиторий на github и загружаем файл matMulSeq

# Приложение 1. Исходный файл matMulSeq.cpp

#include <cstdlib> // Для atoi() - преобразования строки в число

#include <iostream> // Для ввода/вывода

#include "time.h" // Для работы с временем (лучше использовать <ctime>)

using namespace std; // Использование стандартного пространства имен

// Функция вывода матрицы

template <typename T>

int matrixOutput(T \*Mat, int size, string name) {

cout << "\"" << name << "\" matrix output:" << endl;

for (int i=0; i < size; i++) {

for (int j = 0; j < size; j++) {

cout << Mat[i\*size + j] << " "; // Линейный доступ к элементам матрицы

}

cout << endl;

}

return 1;

}

int main(int argc, char \*argv[]) {

clock\_t start; // Переменная для замера времени

// Проверка аргументов командной строки

if (argc != 2) {

cout << "Использование: " << argv[0] << " <размер>" << endl;

return -1; // Выход при ошибке

}

const int N = atoi(argv[1]); // Получаем размер матрицы

cout << "Инициализация матриц..." << endl;

// Выделение памяти под матрицы

float \*A = new float[N\*N]; // Матрица A

float \*B = new float[N\*N]; // Матрица B

// Заполнение матриц значениями

for (int i=0; i < N; i++) {

for (int j = 0; j < N; j++) {

A[i\*N + j] = (i+1) \* (j+1); // Формула для Aij

B[i\*N + j] = (i+1) + 2\*(j+1); // Формула для Bij

}

}

float \*C = new float[N\*N]; // матрица размерностью N

cout << "Вычисление произведения..." << endl;

start = clock(); // Начало замера времени

// Умножение матриц

for (int i=0; i < N; i++) {

for (int j = 0; j < N; j++) {

C[i\*N + j] = 0; // Обнуление элемента

for (int k = 0; k < N; k++) {

C[i\*N + j] += A[i\*N + k] \* B[k\*N + j]; // Вычисление суммы произведений

}

}

}

// Вывод времени выполнения

cout << "Время вычислений: " << double(clock() - start)/CLOCKS\_PER\_SEC

<< " секунд" << endl;

// Освобождение памяти

delete [] A;

delete [] B;

delete [] C;

return 0; // Успешное завершение

}