**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра безопасности информационных систем**

**ОТЧЁТ**

по практической работе работе №1 на тему:   
**«Массивы данных»**

по дисциплине «Алгоритмы и структуры данных»

Выполнил: студент группы ИСТ-114,Константинов А.Д.

«21» сентября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/А.Д. Константинов/

Принял: к.ф.-м.н., доцент, И.А. Моисеев

«05» октября 2022 г. \_\_\_\_\_\_\_\_\_\_\_/ И.А. Моисеев /

**Содержание:**

[Цель работы: 3](#_Toc114692066)

[Листинг кода: 3](#_Toc114692067)

[Результат работы программы: 5](#_Toc114692068)

[Вывод: 6](#_Toc114692069)

Цель работы:

Требуется вычислить матрицу, которая является результатом перемножения двух действительных матриц размерности m×n и n×m

# Листинг кода:

#include <iostream>

#include <fstream>

#include <vector>

void printMatrix(std::vector<std::vector<int>> first){

std::cout<< std::endl <<"Result matrix array"<< std::endl;

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

for (int j = 0; j < first[i].size(); ++j) {

std::cout << first[i][j] << " ";

}

std::cout << std::endl;

}

std::cout << std::endl;

}

void printMatrix2(std::vector<std::vector<int>> first,std::vector<std::vector<int>> second){

std::cout<< std::endl <<"First array"<< std::endl;

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

for (int j = 0; j < first[i].size(); ++j) {

std::cout << first[i][j] << " ";

}

std::cout << std::endl;

}

std::cout<<"Second array"<< std::endl;

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

for (int j = 0; j < second[i].size(); ++j) {

std::cout << second[i][j] << " ";

}

std::cout << std::endl;

}

std::cout << std::endl;

}

//Метод решение со сложностью O(n^3) - "в лоб"

std::vector<std::vector<int>> Multipy(std::vector<std::vector<int>> first,std::vector<std::vector<int>> second){

std::vector<std::vector<int>> result;

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

result.push\_back(std::vector<int>());

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

//новый элемент есть сумма умножения столбца А на строку Б

int tmp =0;

for (int k = 0; k < second.size(); ++k) {

tmp += first[i][k]\*second[k][j];

}

result[i].push\_back(tmp);

}

}

return result;

}

int main() {

std::ifstream firstMatrix;

std::ifstream secondMatrix;

firstMatrix.open("a.txt");

secondMatrix.open("b.txt");

if(!firstMatrix || !secondMatrix)

{

std::cout << "Files is closed!" << std::endl;

return -1;

}

//Objects for data

std::vector<std::vector<int>> first;

std::vector<std::vector<int>> second;

//Парсинг данных из файлов

//Возможно сделано не эффективно

// Но работает, так что пусть будет оставаться так

//Reading matrix from files

int currentLine = 0;

while(!firstMatrix.eof()) {

std::string a;

getline(firstMatrix,a);

std::string tmpS;

first.push\_back(std::vector<int>());

for (int i = 0; i < a.length(); ++i) {

if(a[i] != 32) {

tmpS+=a[i];

}

else{

first[currentLine].push\_back(atoi(tmpS.c\_str()));

tmpS="";

}

}

first[currentLine].push\_back(atoi(tmpS.c\_str()));

currentLine++;

}

currentLine = 0;

while(!secondMatrix.eof()) {

std::string a;

getline(secondMatrix,a);

std::string tmpS;

second.push\_back(std::vector<int>());

for (int i = 0; i < a.length(); ++i) {

if(a[i] != 32) {

tmpS+=a[i];

}

else{

second[currentLine].push\_back(atoi(tmpS.c\_str()));

tmpS="";

}

}

second[currentLine].push\_back(atoi(tmpS.c\_str()));

currentLine++;

}

printMatrix2(first,second); //Выводим данные, полученные в файлах

printMatrix(Multipy(first,second));

firstMatrix.close();

secondMatrix.close();

return 0;

}

Вводные данные располагаются в файлах a.txt и b.txt

# Результат работы программы:

/Users/andrew/Projects/testproj/cmake-build-debug/testproj

First array

1 2 3

4 5 6

Second array

1 2

3 4

5 6

Result matrix array

22 28

49 64

Process finished with exit code 0

# Вывод:

В ходе выполнения практической работы была написана программа, которая считывает две матрицы из файлов a.txt и b.txt при помощи ifstream, парсит полученные данные и сохраняет в динамическую структуру данных – vector. После этого матрицы перемножаются по формуле и получившаяся матрица выводится на экран. После чего программа завершается

Приложение:

Тестовые файлы(a.txt;b.txt). и файл исходного кода main.cpp можно найти по адресу <https://github.com/StSiRe/Algoritms/tree/master/Algorithms%20and%20Data%20Structures/Matrix%20multipy/first>