ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

имени В. И. ВЕРНАДСКОГО»

ТАВРИЧЕСКАЯ АКАДЕМИЯ

Факультет математики и информатики

**ОТЧЕТ**

**об учебной практике**

по направлению подготовки 01.03.02 Прикладная математика и информатика

образовательного уровня бакалавр

студента 2 курса \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Симферополь, 2019 г.

Содержание

[Общая информация 2](#__RefHeading___Toc13603026)

[Индивидуальное задание по программированию 2](#__RefHeading___Toc13603027)

[Постановка задачи 2](#__RefHeading___Toc13603028)

[Описание алгоритма 3](#__RefHeading___Toc13603029)

[Описание интерфейса пользователя программы 3](#__RefHeading___Toc13603030)

[Программа 3](#__RefHeading___Toc13603031)

[Тестирование 5](#__RefHeading___Toc13603032)

[Блок-схема 6](#__RefHeading___Toc13603033)

[Информационное обеспечение 8](#__RefHeading___Toc13603034)

# Общая информация

В соответствии с учебным планом подготовки бакалавров по направлению 01.03.02 Прикладная математика и информатика учебная практика проходила с 9 июля по 16 июля 2019 года в Центре компьютерных технологий Таврической академии (ТА КФУ) и Лаборатории программного обеспечения компьютерных систем факультета математики и информатики.

Целью учебной практики является закрепление теоретических знаний по программированию, приобретение профессиональных навыков и компетенций в области современных информационных технологий, получение опыта профессиональной деятельности в производственном коллективе.

Основные задачи учебной практики:

* выполнение индивидуального задания на разработку программы в среде С++.

# Индивидуальное задание по программированию

## Постановка задачи

Для заданной матрицы A(m,n) найти её произведение на транспонированную к ней AAT в упакованном виде (так как результат — симметричная матрица). Симметричная матрица упаковывается в одномерный массив, в котором последовательно записываются строки её части, находящейся не ниже главной диагонали.

## Описание алгоритма

Для решения задачи создаются следующие функции:

1. readNumberOfRaws считывает с клавиатуры количество строк матрицы.
2. readNumberOfСolumns считывает с клавиатуры колличество столбцов матрицы.
3. readMatrix считывает с клавиатуры заданную матрицу и записывает ее в глобальны массив векторов matrix.
4. vector <long long> getResult считает и возвращает ответ в упакованном виде.
5. printArray выводит сгенерированный массив на экран.
6. SetConsoleOutputCP(1251) для поддержки кириллицы .
7. system("pause") останавливает программу перед завершением, чтобы пользователь смог увидеть ответ.

В основной программе выполняются следующие действия:

* 1. Считывание с клавиатуры количество строк матрицы – readNumberOfRaws.
  2. Считывание с клавиатуры количество столбцов матрицы – readNumberOfСolumns.
  3. Считывание с клавиатуры самой матрицу и запись в глобальный массив векторов matrix – readMatrix.
  4. Считывание и возвращения ответа в упакованном виде - vector <long long> getResult.
  5. Вывод сгенерированного массива на экран – printArray.

## Описание интерфейса пользователя программы

* 1. Вывод на экран "Введите колличество строк матрицы: ".  
     Пользователь вводит количество строк в матрице в функции readNumberOfRaws.
  2. Вывод на экран "Введите колличество столбцов матрицы: ".  
     Пользователь вводит количество столбцов матрицы в функции readNumberOfСolumns.
  3. Вывод на экран "Введите значения матрицы:".  
     Пользователь вводит саму матрицу в функции readMatrix.

## Программа

#include <iostream>

#include <vector>

#include <windows.h> // библиотека для функции SetConsoleOutputCP()

using namespace std;

vector < vector <long long> > matrix;

// функция считывает с клавиатуры колличество строк матрицы

int readNumberOfRaws() {

int n;

cout << "Введите колличество строк матрицы: ";

cin >> n;

return n;

}

// функция считывает с клавиатуры колличество столбцов матрицы

int readNumberOfСolumns() {

int n;

cout << "Введите колличество столбцов матрицы: ";

cin >> n;

return n;

}

// функция считывает с клавиатуры заданную матрицу и записывает ее в глобальны массив векторов matrix

void readMatrix(int r, int c) {

cout << "Введите значения матрицы:" << endl;

matrix.reserve(r);

matrix.resize(r);

for (int i = 0, k = 0; i < r; i++) {

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

long long tmp;

cin >> tmp;

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

}

}

return;

}

// функция считает и возвращает ответ в упакованном виде

vector <long long> getResult(int r, int c) {

vector <long long> a;

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

for (int j = i; j < r; j++) {

long long ans = 0;

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

ans += matrix[i][k] \* matrix[j][k];

}

a.push\_back(ans);

}

}

return a;

}

// функция выводит на экран массив

void printArray(vector <long long> arr) {

cout << "Произведение данной матрицы на транспонированную в упакованном виде:" << endl;

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

cout << arr[i] << " ";

}

cout << endl;

return;

}

int main() {

int r, c;

vector <long long> result;

SetConsoleOutputCP(1251); // функция для поддержки Кириллицы

r = readNumberOfRaws();

c = readNumberOfСolumns();

readMatrix(r, c);

result = getResult(r, c);

printArray(result);

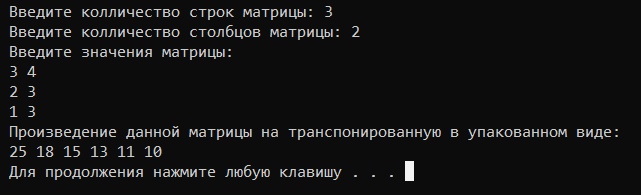
system("pause"); // функция останавливает программу перед завершением, чтобы пользователь смог увидеть ответ

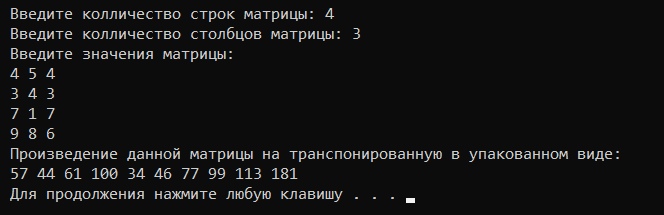
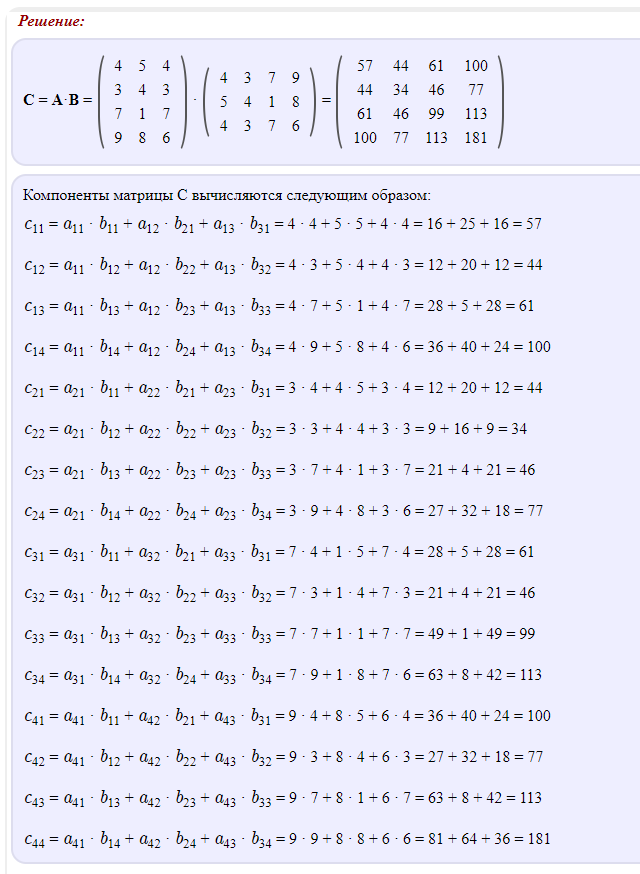
return 0; // завершение работы программы

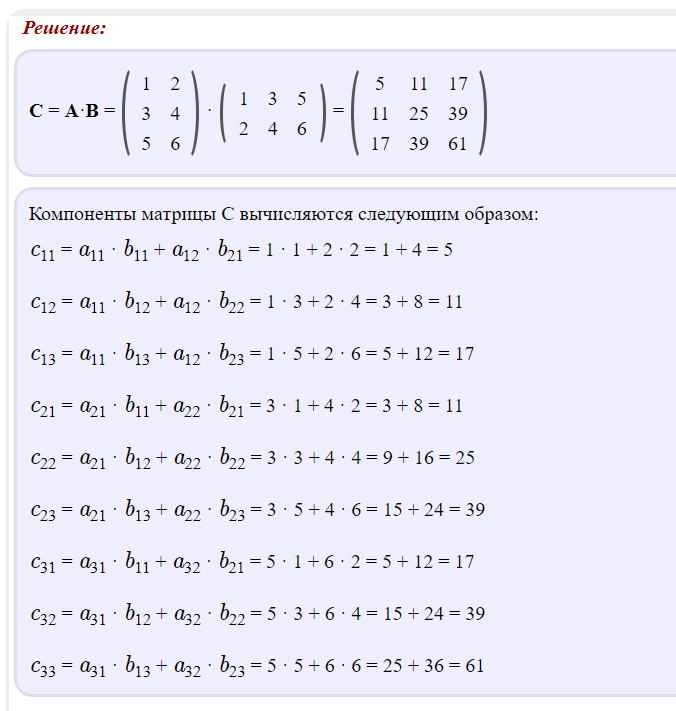
}

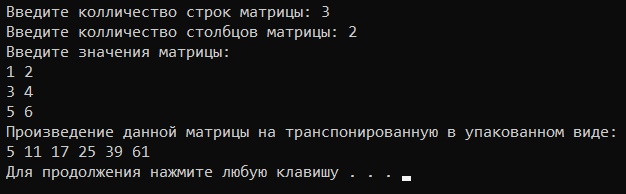
## Тестирование

Для визуализации результатов работы программы использовалась функция printArray, осуществляющая вывод элементов массива на экран.

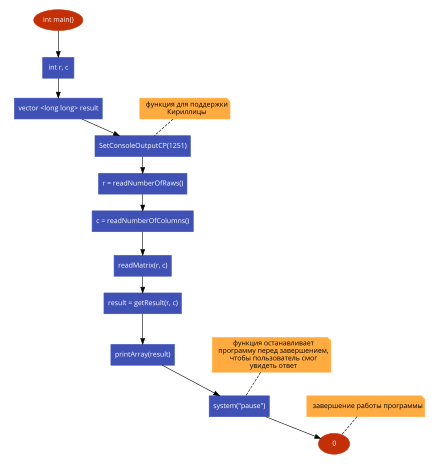
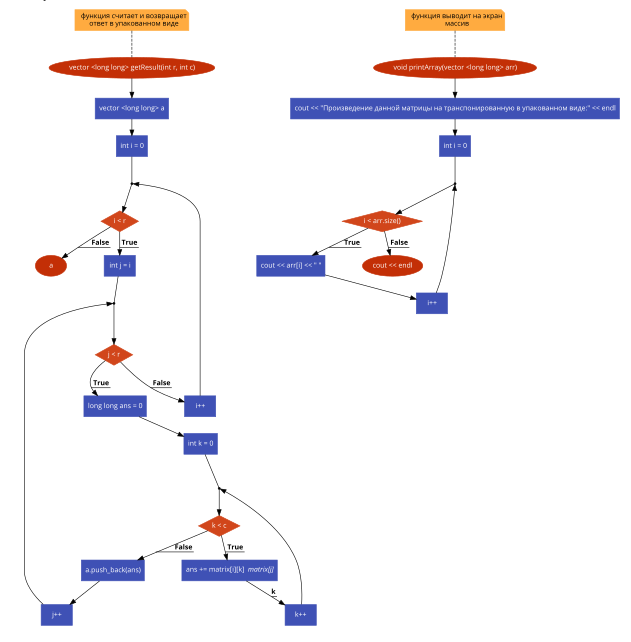
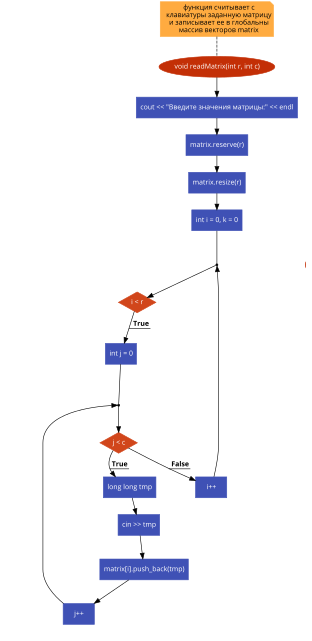
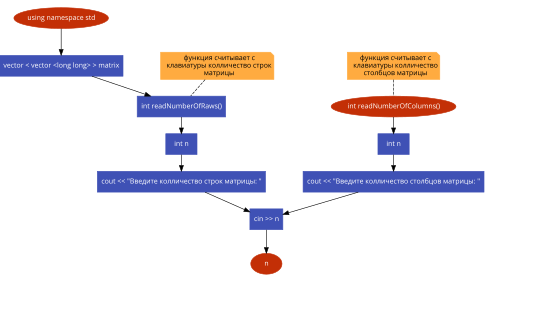


Анализ результатов





## Блок-схема



# Информационное обеспечение

Список литературы:

1. Дейтел Х. Как программировать на С++ : научное издание/ Х. Дейтел, П. Дейтел; Пер. В. Тимофеев, Ред. В. Тимофеев.  М. : Бином-Пресс, 2008.  1454 с.

Электронные учебники и справочники, форумы по программированию:

1. <https://en.cppreference.com/w/cpp/numeric/random/rand>
2. <https://codeforces.com/blog/entry/61587>
3. <http://code2flow.com/>
4. <https://www.wolframalpha.com/>
5. https://www.geogebra.org/classic