**Курсовая работа**

**по дисциплине: «Программирование и основы алгоритмизации»**

2019 г

Оглавление

[Введение 3](#_Toc12208639)

[Задание 4](#_Toc12208640)

[1 Описание данных 5](#_Toc12208641)

[2 Исходный код программы 6](#_Toc12208642)

[3 Тестирование 8](#_Toc12208643)

[Заключение 11](#_Toc12208644)

[Список литературы 12](#_Toc12208645)

# Введение

Целью курсовой работы является разработка в среде Microsoft Visual Studio на языке программирования C консольного приложения Windows для работы с квадратными матрицами.

Задачей курсовой работы является закрепление изученного материала и освоение методов обработки информации с использованием динамических массивов языка C, а также развитие практических навыков при выборе написании программ, тестировании, отладки и оформлении сопроводительной документации.

Данная работа выполнена в виде программы на языке программирования C. Программа предназначена для работы с квадратными матрицами, элементы которых хранятся в динамических массивах.

Элементы матриц вводятся пользователем с клавиатуры в интерактивном режиме. Пользовательский интерфейс реализован в виде текстовой консоли.

# Задание

Разработать консольное приложение Windows на языке C. В исходном коде программы необходимо выделить одну или несколько подпрограмм (функций). Требования к составляемым подпрограммам:

* каждая подпрограмма является функционально завершённой и должна вызываться более одного раза с разными фактическими параметрами;
* подпрограмма обработки не должна содержать ввод/вывод данных;
* в подпрограммах не рекомендуется использовать глобальные переменные.

Вариант 3.

Даны матрицы A(5x5), B(7x7), A(4x4). Вычислить значение выражения PA + PB - PC, где PA (PB, PC) - произведение положительных элементов главной диагонали матрицы A (B, C).

# Описание данных

Исходными данными в программе являются целочисленные элементы трёх квадратных матриц, вводимые пользователем с клавиатуры. Размерной исходных матриц указана в таблице 1.

Таблицы 1. Размерность исходных матриц

|  |  |
| --- | --- |
| A | 5 |
| B | 7 |
| C | 4 |

В качестве промежуточных данных используются локальные переменные P\_A, P\_B и P\_C, хранящие произведения положительных элементов главных диагоналей матриц A, B и C, соответственно.

Результатом выполнения программы является значение выражения P\_A + P\_B - P\_C.

# Исходный код программы

#include "pch.h"

#include <stdlib.h>

#include <stdio.h>

#define A\_SIZE 5

#define B\_SIZE 7

#define C\_SIZE 4

// Предварительное объявление функции

int main\_positive\_product(int\* matrix, int dimSize);

// Основная функция - точка входа программы

int main()

{

// Выделение памяти под массивы для

// хранения элементов исходных матриц

int\* A = (int\*)malloc(sizeof(int) \* A\_SIZE \* A\_SIZE);

int\* B = (int\*)malloc(sizeof(int) \* B\_SIZE \* B\_SIZE);

int\* C = (int\*)malloc(sizeof(int) \* C\_SIZE \* C\_SIZE);

// Ввод матрицы A

printf("A[%d][%d]:\n", A\_SIZE, A\_SIZE);

for (int i = 0; i < A\_SIZE; ++i)

{

for (int j = 0; j < A\_SIZE; ++j)

{

printf("input A[%d][%d]:", i, j);

// Смещаем При вводе указатель на &A[i][j]

scanf("%d", A + A\_SIZE \* i + j);

}

}

// Ввод матрицы B

printf("B[%d][%d]:\n", B\_SIZE, B\_SIZE);

for (int i = 0; i < B\_SIZE; ++i)

{

for (int j = 0; j < B\_SIZE; ++j)

{

printf("input B[%d][%d]:", i, j);

// Смещаем При вводе указатель на &B[i][j]

scanf("%d", B + B\_SIZE \* i + j);

}

}

// Ввод матрицы C

printf("C[%d][%d]:\n", C\_SIZE, C\_SIZE);

for (int i = 0; i < C\_SIZE; ++i)

{

for (int j = 0; j < C\_SIZE; ++j)

{

printf("input C[%d][%d]:", i, j);

// Смещаем При вводе указатель на &C[i][j]

scanf("%d", C + C\_SIZE \* i + j);

}

}

// Вычисляем произведения положительных элементов

// главной диагонали каждой из исходных матриц

int P\_A = main\_positive\_product(A, A\_SIZE);

int P\_B = main\_positive\_product(B, B\_SIZE);

int P\_C = main\_positive\_product(C, C\_SIZE);

// Вычисляем результат выражения

int P = P\_A + P\_B - P\_C;

// Выводим результат

printf("%d + %d - %d = %d\n", P\_A, P\_B, P\_C, P);

// Ожидаем реакции пользователя

printf("Press any key to finish\n");

getchar();

// Освобождаем выделенную память

free(A);

free(B);

free(C);

}

/\* Возвращает произведение положительных

элементов главной диагонали квадратной матрицы

Параметры:

matrix - указатель на первый элемент массива,

содержащего элементы квадратной матрицы

dimSize - количество строк/столбцов матрицы

\*/

int main\_positive\_product(int\* matrix, int dimSize)

{

int result = 1;

for (int i = 0; i < dimSize; ++i)

{

int m = matrix[dimSize \* i + i];

if (m > 0)

{

result \*= m;

}

}

return result;

}

# Тестирование

Для тестирование программы выбраны исходные данные таким образом, чтобы на главных диагоналях матриц содержались не только положительные элементы - чтобы убедиться, что программа не учитывает их при вычислении произведений на главной диагонали.

Содержимое тестовой матрицы A приведено в таблице 2.

Таблица 2. Элементы матрицы A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | -2 | 3 | 4 | 5 |
| 1 | 2 | -3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 |

Содержимое тестовой матрицы B приведено в таблице 3.

Таблица 3. Элементы матрицы B

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | -3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 3 | 4 | 5 | 6 | 7 | -7 |

Содержимое тестовой матрицы C приведено в таблице 4.

Таблица 4. Элементы матрицы C

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |
| 1 | 2 | 3 | 4 |

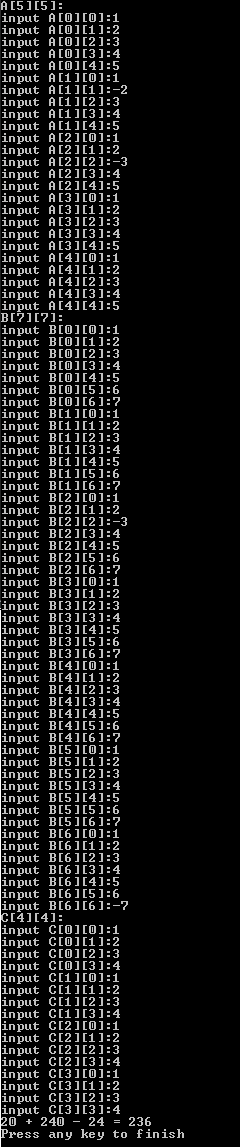


Рисунок 1 - Тестирование программы

Из рисунка 1 видно, что результат вычисляется верно:

PA = 1 \* 4 \* 5 = 20

PB = 1 \* 2 \* 4 \* 5 \* 6 = 240

PC = 1 \* 2 \* 3 \* 4 = 24

PA + PB - PC = 20 + 240 - 24 = 236

# Заключение

В результате работы была спроектирована и разработана программа, позволяющая работать с квадратными матрицами, вводимыми пользователем и вычислять произведение положительных элементов их главных диагоналей.

В ходе работы над проектом получены практические навыки разработки программ на языке высокого уровня C. Изучено множество аспектов разработки программ на этом языке – в первую очередь, организация динамических структур данных и управление памятью.

На базе практического применения, закреплены базовые знания, полученные ранее, при изучении теории.

# Список литературы

1. Стивен Прата Язык программирования C. Лекции и упражнения / Стивен Прата. - М.: Вильямс, 2015. - 928 c.

2. Подбельский В.В, Фомин С.С. Программирование на языке C / М.: «Финансы и статистика». 2003.

3. Дейтел Х.М., Дейтел П.Дж. Как программировать на C / М: «БИНОМ», 2000.

4. Шилдт, Г. Справочник по C. Диасофт М.: 2000.

5. ГОСТ 2.105-95 Общие требования к текстовым документам.