|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 4 |

**Название:**

Массивы

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | С.В. Астахов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Задание**

Вводится матрица 4x4, найти элемент на главной диагонали, произведение которого на предыдущий по столбцу элемент максимально. Вывести его, а также все элементы из его строки и столбца и их адреса.

**Исходный код**

(файл Source.cpp)

*#include <iostream>*

*using namespace std;*

*int main()*

*{*

*puts("Enter elements for 4x4 matrix");*

*int matr[4][4];*

*for (int i = 0; i < 4; i++) {*

*for (int g = 0; g < 4; g++) {*

*scanf\_s("%d", &(matr[i][g]));*

*}*

*}*

*for (int i = 0; i < 4; i++) {*

*for (int g = 0; g < 4; g++) {*

*printf("%6d", matr[i][g]);*

*}*

*puts("\n");*

*}*

*int max = matr[1][1] \* matr[0][1];*

*int maxi = 1;*

*for (int i = 2; i < 4; i++) {*

*if (matr[i][i] \* matr[i - 1][i] > max) {*

*max = matr[i][i] \* matr[i - 1][i];*

*maxi = i;*

*}*

*}*

*printf("max elem: %d", matr[maxi][maxi]);*

*puts("\n His string: ");*

*for (int i = 0; i < 4; i++) {*

*printf("\n %6d %X", matr[maxi][i], &(matr[maxi][i]));*

*}*

*puts("\n His column: ");*

*for (int i = 0; i < 4; i++) {*

*printf("\n %6d %X", matr[i][maxi], &(matr[i][maxi]));*

*}*

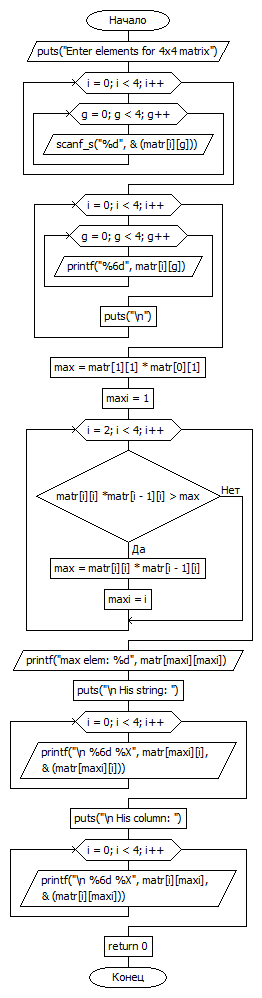
*return 0;*

*}*

**Тесты**

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| 1 2 3 4  1 2 3 4  1 2 3 4  1 2 3 4 | max elem: 4  His string:  1 93F7DC  2 93F7E0  3 93F7E4  4 93F7E8  His column:  4 93F7B8  4 93F7C8  4 93F7D8  4 93F7E8 | max elem: 4  His string:  1 93F7DC  2 93F7E0  3 93F7E4  4 93F7E8  His column:  4 93F7B8  4 93F7C8  4 93F7D8  4 93F7E8 |
| 1 2 3 4  4 3 2 1  5 6 7 8  8 7 6 5 | max elem: 5  His string:  8 3CF79C  7 3CF7A0  6 3CF7A4  5 3CF7A8  His column:  4 3CF778  1 3CF788  8 3CF798  5 3CF7A8 | max elem: 5  His string:  8 3CF79C  7 3CF7A0  6 3CF7A4  5 3CF7A8  His column:  4 3CF778  1 3CF788  8 3CF798  5 3CF7A8 |
| 1 1 1 1  1 7 1 1  1 1 1 1  1 1 1 1 | max elem: 7  His string:  1 137FB14  7 137FB18  1 137FB1C  1 137FB20  His column:  1 137FB08  7 137FB18  1 137FB28  1 137FB38 | max elem: 7  His string:  1 137FB14  7 137FB18  1 137FB1C  1 137FB20  His column:  1 137FB08  7 137FB18  1 137FB28  1 137FB38 |

**Схема алгоритма**



**Вывод**

* Массивы в C++ не имеют значительных отличий от массивов в Delphi. В отличие от Delphi в C++ не осуществляется контроль за размером массива, поэтому стоит быть внимательным при написании программы, чтобы не выйти за пределы заданной части массива.