Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Белгородский государственный технологический университет им. В.Г. Шухова»

(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №16

По дисциплине: «Основы программирования»

Тема: «Работа с многомерными массивами»

Выполнил: студент группы ВТ-231

Борченко Александр Сергеевич

Проверили:

Черников Сергей Викторович

Новожен Никита Викторович

Белгород 2024

Цель работы: получение навыков работы с многомерными массивами.

Содержание работы:

Ссылка на репозиторий: https://github.com/ImperialEQZ/LAB15Matrix

Реализуемые функции:

```
#include "matrix.h"
#include "matrix.c"
#include <math.h>

/*Дана квадратная матрица, все элементы которой различны. Поменять местами
строки, в которых находятся максимальный и минимальный элементы */
void Lab16_task1(matrix *m) {
    int min = getMinValuePos(*m).rowIndex;
    int max = getMaxValuePos(*m).rowIndex;
    swapRows(m, max, min);
}
//Номер 2 лаб 16 (a)
int getMax(int *a, int n) {

    int max = a[0];
    for(size_t i = 0; i < n; i++) {
        if(max < a[i])
            max = a[i];
    }

    return max;
}
//Номер 2 лаб 16 (b)
void sortRowsByMinElement(matrix m) {
    insertionSortRowsMatrixByRowCriteria(&m, getMax);
}
//номер 3 лаб 16 (a)
int getMin(int *a, int n) {

    int min = a[0];
    for (size_t i = 1; i < n; i++) {
        if(a[i] < min)
            min = a[i];
    }

    return min;
}
//номер 3 лаб 16 (b)
void sortColsByMinElement(matrix *m) {
    selectionSortColsMatrixByColCriteria(*m, getMin);
}
//Задание 4 (1)
matrix mulMatrices(matrix m1, matrix m2) {

    if(m1.nRows * m1.nCols == m2.nRows * m2.nCols){
    matrix result = getMemMatrix(m1.nRows, m2.nCols);
    for (int i = 0; i < m1.nRows; i++) {
        for (int j = 0; j < m2.nCols; j++) {
            result.values[i][j] = 0;
            for (int k = 0; k < m1.nCols; k++) {
                result.values[i][j] += m1.values[i][k] * m2.values[k][j];
                }
            }
        }

        return result;
    }
}
//Задание 4 (2)
void getSquareOfMatrixIfSymmetric(matrix *m) {
    if(isSymmetricMatrix(m)){
```

```
    matrix temp = mulMatrices(*m, *m);
    freeMemMatrix(m);
    *m = temp;
    }
}
//Задание 5 (1)
bool isUnique(long long *a, int n) {
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (a[i] == a[j] && i != j)
                return false;
        }
    }
    return true;
}
//Задание 5 (2)
//getSum из библиотеки matrix lab 15:
/*long long getSum(int *a, int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += a[i];
    }
    return sum;
}*/
//Задание 5 (3)
void transposeIfMatrixHasNotEqualSumOfRows(matrix *m) {
    long long temp_sum[m->nRows];
    for (int i = 0; i < m->nRows; i++) {
        temp_sum[i] = getSum(m->values[i], m->nCols);
    }

    if (isUnique(temp_sum, m->nRows))
        transposeSquareMatrix(m);
}
//Задание 6 (a)
bool isMutuallyInverseMatrices(matrix m1, matrix m2) {
    if (m1.nCols != m2.nCols || m1.nRows != m2.nRows)
        return false;
    else {
        matrix mul_m = mulMatrices(m1, m2);
        return isEMatrix(&mul_m);
    }
}
//Задание 7 (a)
int max(int a, int b){
    return a > b ? a : b;
}
//Задание 7 (b)
long long findSumOfMaxesOfPseudoDiagonal(matrix m) {
    long long max_sum_S = 0;
    int max_number = 0;
    for (int i = 1; i < m.nCols; i++) {
        int i_rows = 0;
        int i_cols = i;
        max_number = m.values[i_rows][i_cols];
        while (i_cols < m.nCols && i_rows < m.nRows) {
            max_number = max(max_number, m.values[i_rows] [i_cols]);
            i_rows++;
            i_cols++;
        }
        max_sum_S += max_number;
    }
    for (int i = 1; i < m.nRows; i++) {
```

```
        int i_row = i;
        int i_col = 0;
        max_number = m.values[i_row][i_col];
        while (i_col < m.nRows && i_row < m.nRows) {
            max_number = max(max_number, m.values[i_row][i_col]);
            i_row++;
            i_col++;
        }
        max_sum_S += max_number;
    }
    return max_sum_S;
}
//Задание 8 (а)
int getMinInArea(matrix m) {
    position max_pos = getMaxValuePos(m);

    int min = m.values[max_pos.rowIndex][max_pos.colIndex];
    int row = max_pos.rowIndex - 1;
    int elementary_col;

    if(max_pos.colIndex - 1 >= 0) {
        elementary_col = max_pos.colIndex - 1;
    } else {
        elementary_col = 0;
    }

    int col = elementary_col;
    int col_final;

    if(max_pos.colIndex + 1 <= m.nCols - 1) {
        col_final = max_pos.colIndex + 1;
    } else {
        col_final = m.nCols - 1;
    }

    for (int i = row; i >= 0; i--) {
        for (int j = col; j <= col_final; j++)
            min = min < m.values[i][j] ? min : m.values[i][j];

        if(elementary_col - 1 >= 0) {
            elementary_col = elementary_col - 1;
        } else {
            elementary_col = 0;
        }

        col = elementary_col;

        if(col_final + 1 <= m.nCols - 1) {
            col_final = col_final + 1;
        } else {
            col_final = m.nCols - 1;
        }
    }

    return min;
}
//Задание 9 (1)
float getDistance(int *a, int n) {

    float dist = 0;
    for (int i = 0; i < n; i++) {
        dist += a[i] * a[i];
    }
```

```
    dist = sqrt(dist);
    return dist;
}
//Задание 9 (2)
void insertionSortRowsMatrixByRowCriteriaF(matrix *m,
                                           float (*criteria)(int *, int)) {
    float temp[m->nRows];
    float mem_num;
    for (int i = 0; i < m->nRows; i++) {
        float res = criteria(m->values[i], m->nCols);
        temp[i] = res;
    }
    int min_id;
    for (int j = 0; j < m->nRows; j++) {
        min_id = j;
        for (int i = j + 1; i < m->nRows; i++) {
            if (temp[i] < temp[min_id]) {
                min_id = i;
            }
        }
        if (min_id != j) {
            mem_num = temp[j];
            temp[j] = temp[min_id];
            temp[min_id] = mem_num;
            swapRows(m, j, min_id);
        }
    }
}
//Задание 9 (3)
void sortByDistances(matrix *m) {
    insertionSortRowsMatrixByRowCriteriaF(m, getDistance);
}
//Задание 10 (a)
int cmp_long_long(const void *pa, const void *pb) {

    if (*(long long int *) pa - *(long long int *) pb < 0)
        return -1;
    if (*(long long int *) pa - *(long long int *) pb > 0)
        return 1;
    return 0;
}
//Задание 10 (b)
int countNUnique(long long *a, int n) {
    int count = 0;
    int is_unique = 0;
    for (int i = 0; i < n - 1; i++) {
        if (!is_unique && a[i] == a[i + 1]) {
            count += 1;
            is_unique = 1;
        } else
            is_unique = 0;
    }
    return count;
}
//Задание 10 (c)
int countEqClassesByRowsSum(matrix m) {
    long long temp[m.nRows];
    for (int i = 0; i < m.nRows; i++) {
        temp[i] = getSum(m.values[i], m.nCols);
    }

    qsort(temp, m.nRows, sizeof(long long int), cmp_long_long);
```

```
    return countNUnique(temp, m.nRows);
}
//Задание 11 (a)
int getNSpecialElement(matrix m) {

    int sum;
    int max;
    int counter = 0;

    for (int i = 0; i < m.nCols; i++) {
        max = m.values[0][i];
        sum = max;
        for (int j = 1; j < m.nRows; j++) {

            if(max > m.values[j][i]) {
                max = max;
            } else {
                max = m.values[j][i];
            }

            sum += m.values[j][i];
        }

        sum -= max;

        if(sum < max) {
            counter = counter + 1;
        } else {
            counter = counter;
        }
    }
    return counter;
}
//Задание 12 (a)
position getLeftMin(matrix m) {
    return getMinValuePos(m);
}
//Задание 12 (b)
void swapPenultimateRow(matrix *m, int n) {
    if (isSquareMatrix(m)) {
        for (int i = m->nRows - 1; i >= 0; i--)
            m->values[m->nRows - 2][i] = m->values[i][n];
    }
}
//Задание 13 (a)
bool isNonDescendingSorted(int *a, int n) {
    for (int i = 0; i < n - 1; i++)
        if (a[i] > a[i + 1])
            return false;
    return true;
}
//Задание 13 (b)
bool hasAllNonDescendingRows(matrix m) {
    for (int i = 0; i < m.nRows; i++)
        if (!isNonDescendingSorted(m.values[i], m.nCols))
            return false;
    return true;
}
//Задание 13 (c)
int countNonDescendingRowsMatrices(matrix *ms, int nMatrix) {
    int count = 0;
    for (int i = 0; i < nMatrix; i++)
        if (hasAllNonDescendingRows(ms[i]))
```

```
            count++;
    return count;
}
//из лаб 15. Задание 14 (a)
/*int countValues(const int *a, int n, int value) {
    int count = 0;
    for (size_t i = 0; i < n; i++) {
        if (a[i] == value) {
            ++count;
        }
    }

    return count;
}*/
//из лаб 15. Задание 14 (b)
/*int countZeroRows(matrix m) {
    int result = 0;
    for (size_t i = 0; i < m.nRows; i++) {
        int count = countValues(m.values[i], m.nCols, 0);
//Если количество найденных нулевых элементов = общему количеству столбцов в
матрице, то увеличивается result
        if (count == m.nCols)
            ++result;
    }

    return result;
}*/
//Задание 14 (c)
void printMatrixWithMaxZeroRows(matrix *ms, int nMatrix) {

    int arr_count[nMatrix];

    int max;
    int count;

    for (int i = 0; i < nMatrix; i++) {
        count = countZeroRows(ms[i]);
        max = max > count ? max : count;
        arr_count[i] = count;
    }
    for (int i = 0; i < nMatrix; ++i) {
        if (arr_count[i] == max)
            outputMatrix(ms[i]);
    }
}
//Задание 15
void Lab16_task15(matrix *ms, int nMatrix) {
    int temp_mem[nMatrix];
    int abs;
    int max = 0;
    for (int i = 0; i < nMatrix; i++) {
        for (int j = 0; j < ms->nRows; j++)
            for (int k = 0; k < ms->nCols; k++) {
                abs = ms[i].values[j][k] > 0 ? ms[i].values[j][k] :
                      -1 * ms[i].values[j][k];
                max = max > abs ? max : abs;
            }
        temp_mem[i] = max;
        max = 0;
    }

    int min = temp_mem[0];
    for (int i = 1; i < nMatrix; i++)
```

```
        min = min < temp_mem[i] ? min : temp_mem[i];
    for (int i = 0; i < nMatrix; i++)
        if (temp_mem[i] == min)
            outputMatrix(ms[i]);
}
```

Тесты:

```
//тест
void test_Lab16_task1() {
    matrix m = createMatrixFromArray((int[]) {
        3, 3, 3,
        4, 4, 4,
        5, 5, 5},
                                     3, 3);
    matrix m_test = createMatrixFromArray((int[]) {
        5, 5, 5,
        4, 4, 4,
        3, 3, 3},
                                         3, 3);
    Lab16_task1(&m);

    assert(areTwoMatricesEqual(&m, &m_test));

    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}

void Lab16_task2(matrix *m) {
    sortRowsByMinElement(*m);
}
//тест
void test_Lab16_task2() {
    matrix m = createMatrixFromArray((int[]) {
        23, 7, 7,
        21, 7, 7,
        19, 7, 7},
                                     3, 3);
    matrix m_test = createMatrixFromArray((int[]) {
        19, 7, 7,
        21, 7, 7,
        23, 7, 7},
                                         3, 3);
    Lab16_task2(&m);

    assert(areTwoMatricesEqual(&m, &m_test));

    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
void Lab16_task3(matrix *m) {
    sortColsByMinElement(m);
}
//тест (с пособия)
void test_Lab16_task3() {
    matrix m = createMatrixFromArray((int[]) {
        3, 5, 2, 4, 3, 3,
        2, 5, 1, 8, 2, 7,
        6, 1, 4, 4, 8, 3},
                                     3, 6);
    matrix m_test = createMatrixFromArray((int[]) {
        5, 2, 3, 3, 3, 4,
        5, 1, 2, 2, 7, 8,
        1, 4, 6, 8, 3, 4},
                                         3, 6);
    Lab16_task3(&m);

    assert(areTwoMatricesEqual(&m, &m_test));
```

```
    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
//тесты
void Lab16_task4(matrix *m) {
    getSquareOfMatrixIfSymmetric(m);
}
void test_Lab16_task4() {
    matrix m = createMatrixFromArray((int[]) {
        3, -2, 1,
        -2, 0, -4,
        1, -4, 2,},
                                        3, 3);
    matrix m_test = createMatrixFromArray((int[]) {
        14,      -10,     13,
        -10,     20,      -10,
        13,      -10,     21
                                             },
                                             3, 3);
    Lab16_task4(&m);
    assert(areTwoMatricesEqual(&m, &m_test));
    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
//тесты
void Lab16_task5(matrix *m) {
    transposeIfMatrixHasNotEqualSumOfRows(m);
}
void test_Lab16_task5() {
    matrix m = createMatrixFromArray((int[]) {
        10, 9, 8,
        7,  6, 5,
        4,  3, 2},
                                       3, 3);
    matrix m_test = createMatrixFromArray((int[]) {
        10, 7, 4,
        9,  6, 3,
        8,  5, 2},
                                             3, 3);
    Lab16_task5(&m);

    assert(areTwoMatricesEqual(&m, &m_test));

    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
//тесты
bool Lab16_task6(matrix m1, matrix m2) {
    return isMutuallyInverseMatrices(m1, m2);
}
void test_Lab16_task6() {
    int data1[] = {3,   4,
                   5,   7,};

    int data2[] = {7, -4,
                    -5,3};


    matrix m1 = createMatrixFromArray(data1, 2, 2);
    matrix m2 = createMatrixFromArray(data2, 2, 2);

    assert(Lab16_task6(m1, m2));
```

```
    freeMemMatrix(&m1);
    freeMemMatrix(&m2);
}
//тесты
long long Lab16_task7(matrix m) {
    return findSumOfMaxesOfPseudoDiagonal(m);
}

void test_Lab16_task7() {
//Пример из пособия
    matrix m = createMatrixFromArray((int[]) {
        3, 2, 5, 4,
        1, 3, 6, 3,
        3, 2, 1, 2},
                                     3, 4);
    assert(Lab16_task7(m) == 20);
    freeMemMatrix(&m);
}
//тесты
int Lab16_task8(matrix m) {
    return getMinInArea(m);
}

void test_Lab16_task8() {
//Пример из пособия
    matrix m = createMatrixFromArray((int[]) {
        6, 8, 9, 2,
        7, 12, 3, 4,
        10, 11, 5, 1},
                                  3, 4);
    assert(Lab16_task8(m) == 6);
    freeMemMatrix(&m);
}
//тесты
void Lab16_task9(matrix *m) {
    sortByDistances(m);
}
void test_Lab16_task9() {
    matrix m = createMatrixFromArray((int[]) {
        5, -20,
        10, -10,
        -8, -25},
                                     3, 2);
    matrix m_test = createMatrixFromArray((int[]) {
        10, -10,
        5, -20,
        -8, -25},
                                          3, 2);
    Lab16_task9(&m);

    assert(areTwoMatricesEqual(&m, &m_test));

    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
//тесты
int Lab16_task10(matrix m) {
    return countEqClassesByRowsSum(m);
}

void test_Lab16_task10() {
    matrix m = createMatrixFromArray((int[]) {7, 1,//8 (1)
                                              2, 7,//9 (2)
```

```
                                                5, 4,//9
                                                4, 3,//7 (3)
                                                1, 6,//7
                                                8, 0},//8
                                        6, 2);
//кол-во уникальных строк, с одинаковой суммой
    assert(Lab16_task10(m) == 3);
    freeMemMatrix(&m);
}
//тесты
int Lab16_task11(matrix m) {
    return getNSpecialElement(m);
}
void test_Lab16_task11() {
    matrix m = createMatrixFromArray((int[]) {
        3, 5, 5, 4,
        2, 3, 6, 7,
        12, 2, 1, 2},
                                        3, 4);
    assert(Lab16_task11(m) == 2);
    freeMemMatrix(&m);
}
//тесты
void Lab16_task12(matrix *m, int n) {
    swapPenultimateRow(m, n);
}
void test_Lab16_task12() {
//Пример из пособия
    matrix m = createMatrixFromArray((int[]) {
        1, 2, 3,
        4, 5, 6,
        7, 8, 1},
                                        3, 3);

    Lab16_task12(&m, getLeftMin(m).colIndex);

    matrix m_test = createMatrixFromArray((int[]) {
        1, 2, 3,
        1, 4, 7,
        7, 8, 1},
                                            3, 3);

    assert(areTwoMatricesEqual(&m, &m_test));

    freeMemMatrix(&m);
    freeMemMatrix(&m_test);
}
//тесты
int Lab16_task13(matrix *ms, int nMatrix) {
    countNonDescendingRowsMatrices(ms, nMatrix);
}

void test_Lab16_task13() {
    matrix *ms = createArrayOfMatrixFromArray((int[]) {
        //1 матрица
        7, 1,
        1, 1,
        //2 матрица
        1, 6,
        2, 2,
        //3 матрица
        5, 4,
        2, 3,
```

```
        //4 матрица
        1, 3,
        7, 9},
                                                4, 2, 2);
    assert(Lab16_task13(ms, 4) == 2);

    freeMemMatrices(ms, 4);
}
//тесты
int Lab16_task14(matrix m) {
    return countZeroRows(m);
}
void test_Lab16_task14() {
    matrix *ms = createArrayOfMatrixFromArray((int[]) {
//Пример с пособия
        //1 матрица
        0, 1,
        1, 0,
        0, 0,//ряд 0, в сумме 1
        //2 матрица
        1, 1,//нет 0
        2, 1,
        1, 1,
        //3 матрица
        0, 0,//ряд 0
        0, 0,//ряд 0
        4, 7,//в сумме 2
        //4 матрица
        0, 0,//ряд 0
        0, 1,
        0, 0,//ряд 0, в сумме 2
        //5 матрица
        0, 1,//нет 0
        0, 2,
        0, 3},
                                                5, 3, 2);
    assert(Lab16_task14(ms[0]) == 1);
    assert(Lab16_task14(ms[1]) == 0);
    assert(Lab16_task14(ms[2]) == 2);
    assert(Lab16_task14(ms[3]) == 2);
    assert(Lab16_task14(ms[4]) == 0);
    freeMemMatrices(ms, 5);
}
//тесты
void test_Lab16_task15() {
    matrix *ms = createArrayOfMatrixFromArray((int[]) {
        //1 матрица
        10, -15,
        20, 30,
        //2 матрица
        5, -40,
        -70, 32,
        //3 матрица
        2, 3,
        7, 15,
        //4 матрица
        1, -5,
        8, 25},
                                                4, 2, 2);
    Lab16_task15(ms, 4);
    assert(42 == 42);
    freeMemMatrices(ms, 4);
}
```

```
void test_lab16() {
    test_Lab16_task1();
    test_Lab16_task2();
    test_Lab16_task3();
    test_Lab16_task4();
    test_Lab16_task5();
    test_Lab16_task6();
    test_Lab16_task7();
    test_Lab16_task8();
    test_Lab16_task9();
    test_Lab16_task10();
    test_Lab16_task11();
    test_Lab16_task12();
    test_Lab16_task13();
    test_Lab16_task14();
    test_Lab16_task15();
}

int main() {

    test_lab16();
}
```

Результат тестирования:

```
"C:\Users\Александр\CLionProjects\LAB15Matrix\Lab 16 task.exe"
2       3
7       15

Process finished with exit code 0
```

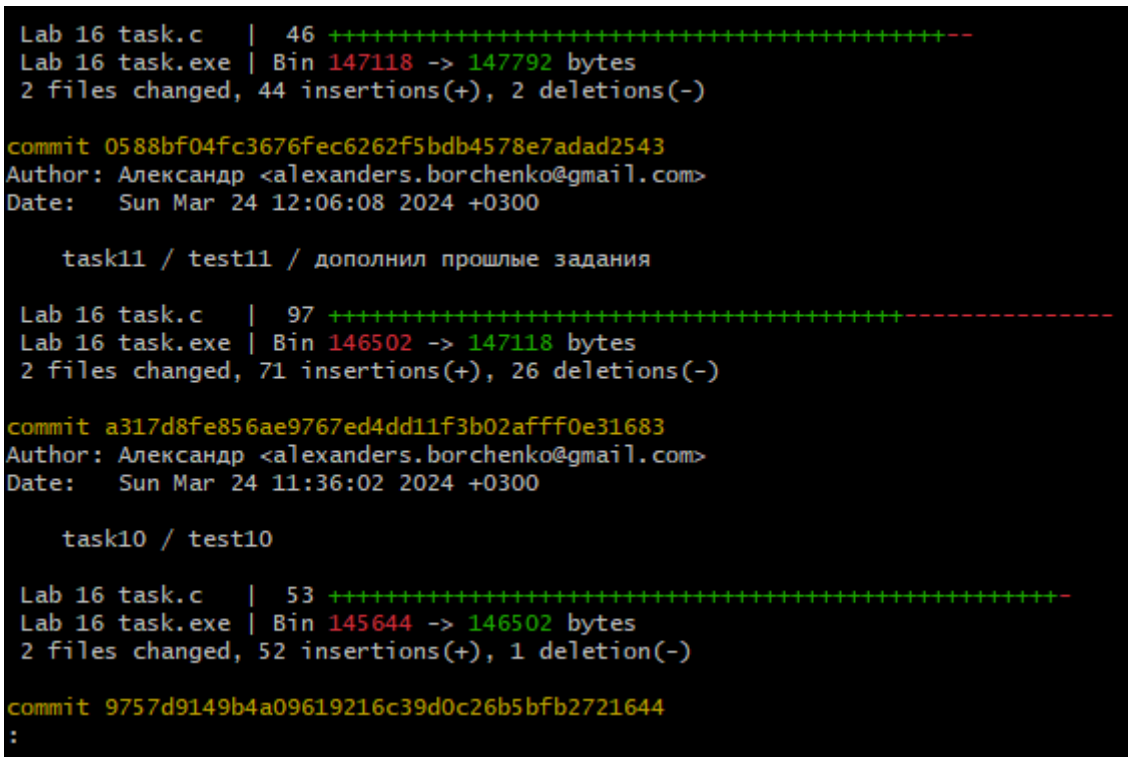
Результат тестирования (без вывода матрицы из 15 задания):

```
"C:\Users\Александр\CLionProjects\LAB15Matrix\Lab 16 task.exe"

Process finished with exit code 0
```

Результат выполнения:

```
commit 6947f1f51d96e3c0e37f54bd2d0721e55172a945e (HEAD -> master, origin/master)
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Mon Mar 25 08:49:13 2024 +0300

    finished lab

 Lab 16 task.c   |  36 ++++++++++++++++++------------------
 Lab 16 task.exe | Bin 150729 -> 150729 bytes
 2 files changed, 18 insertions(+), 18 deletions(-)

commit 473effc885bf63c7bc901062c44784e0f2d4fa1f
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 20:47:31 2024 +0300

    task15 / test15

 Lab 16 task.c   |  46 ++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 149638 -> 150729 bytes
 2 files changed, 45 insertions(+), 1 deletion(-)

commit 93d04434566b2bad35c9aa9ef9eb1d8b06b65bbc
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 15:17:57 2024 +0300

    task14 / test14

 Lab 16 task.c   |  79 ++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 148483 -> 149638 bytes
 2 files changed, 78 insertions(+), 1 deletion(-)

commit 74c4b22e002c604307ba539fb85805314b2d80a4
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 12:57:41 2024 +0300

    task13 / test13

 Lab 16 task.c   |  49 ++++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 147792 -> 148483 bytes
 2 files changed, 48 insertions(+), 1 deletion(-)

commit 12d0994b52a0181adefc121cf9017027e7ba1beb
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 12:29:10 2024 +0300

    task12 / test12 / create general test

 Lab 16 task.c   |  46 ++++++++++++++++++++++++++++++++++++++++++++--
 Lab 16 task.exe | Bin 147118 -> 147792 bytes
 2 files changed, 44 insertions(+), 2 deletions(-)

commit 0588bf04fc3676fec6262f5bdb4578e7adad2543
Author: Александр <alexanders.borchenko@gmail.com>
```

```
commit 0588bf04fc3676fec6262f5bdb4578e7adad2543
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 12:06:08 2024 +0300

    task11 / test11 / дополнил прошлые задания

 Lab 16 task.c   |  97 +++++++++++++++++++++++++++++++++++++++++-----------
 Lab 16 task.exe | Bin 146502 -> 147118 bytes
 2 files changed, 71 insertions(+), 26 deletions(-)

commit a317d8fe856ae9767ed4dd11f3b02afff0e31683
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 11:36:02 2024 +0300

    task10 / test10

 Lab 16 task.c   |  53 ++++++++++++++++++++++++++++++++++++++++++++++++++++
 Lab 16 task.exe | Bin 145644 -> 146502 bytes
 2 files changed, 52 insertions(+), 1 deletion(-)

:...skipping...
commit 694f1f51d96e3c0e37f54bd2d0721e55172a945e (HEAD -> master, origin/mas
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Mon Mar 25 08:49:13 2024 +0300

    finished lab

 Lab 16 task.c   |  36 ++++++++++++++++++------------------------
 Lab 16 task.exe | Bin 150729 -> 150729 bytes
 2 files changed, 18 insertions(+), 18 deletions(-)

commit 473effc885bf63c7bc901062c44784e0f2d4fa1f
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 20:47:31 2024 +0300

    task15 / test15

 Lab 16 task.c   |  46 +++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 149638 -> 150729 bytes
 2 files changed, 45 insertions(+), 1 deletion(-)

commit 93d04434566b2bad35c9aa9ef9eb1d8b06b65bbc
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 15:17:57 2024 +0300

    task14 / test14

 Lab 16 task.c   |  79 ++++++++++++++++++++++++++++++++++++++++++++++++++++
 Lab 16 task.exe | Bin 148483 -> 149638 bytes
 2 files changed, 78 insertions(+), 1 deletion(-)

commit 74c4b22e002c604307ba539fb85805314b2d80a4
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 12:57:41 2024 +0300

    task13 / test13

 Lab 16 task.c   |  49 ++++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 147792 -> 148483 bytes
 2 files changed, 48 insertions(+), 1 deletion(-)

commit 12d0994b52a0181adefc121cf9017027e7ba1beb
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 12:29:10 2024 +0300

    task12 / test12 / create general test

 Lab 16 task.c   |  46 ++++++++++++++++++++++++++++++++++++++++++++--
 Lab 16 task.exe | Bin 147118 -> 147792 bytes
 2 files changed, 44 insertions(+), 2 deletions(-)
```


Lab 16 task.c | 46 ++--
Lab 16 task.exe | Bin 147118 -> 147792 bytes
2 files changed, 44 insertions(+), 2 deletions(-)
commit 0588bf04fc3676fec6262f5bdb4578e7adad2543
Author: Александр <alexanders.borchenko@gmail.com>
Date: Sun Mar 24 12:06:08 2024 +0300
task11 / test11 / дополнил прошлые задания
Lab 16 task.c | 97 ++--------------
Lab 16 task.exe | Bin 146502 -> 147118 bytes
2 files changed, 71 insertions(+), 26 deletions(-)
commit a317d8fe856ae9767ed4dd11f3b02afff0e31683
Author: Александр <alexanders.borchenko@gmail.com>
Date: Sun Mar 24 11:36:02 2024 +0300
task10 / test10
Lab 16 task.c | 53 ++-
Lab 16 task.exe | Bin 145644 -> 146502 bytes
2 files changed, 52 insertions(+), 1 deletion(-)
commit 9757d9149b4a09619216c39d0c26b5bfb2721644
:

```
commit 9757d9149b4a09619216c39d0c26b5bfb2721644
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sun Mar 24 11:18:02 2024 +0300

    task9 / test9 / fix

 Lab 16 task.c   |  66 +++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 143283 -> 145644 bytes
 2 files changed, 65 insertions(+), 1 deletion(-)

commit 2cfaeb41c3384285d5e6c990b94330250557b3f3
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sat Mar 23 17:01:30 2024 +0300

    task8 / я все дальше от бога

 Lab 16 task.c   |  63 +++++++++++++++++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 142675 -> 143283 bytes
 2 files changed, 62 insertions(+), 1 deletion(-)

commit 700bf879f8e726c09229d35fc3c5fdfec0fe7ef0
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sat Mar 23 16:24:53 2024 +0300

    task7 / brain broken

 Lab 16 task.c   |  68 +++++++++++++++++++++++++++++++++++++++-----------------
 Lab 16 task.exe | Bin 142031 -> 142675 bytes
 2 files changed, 48 insertions(+), 20 deletions(-)

commit 19b1469f624ba707a811d5630219aa4ed72b5a2b
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sat Mar 23 10:46:47 2024 +0300

    task6 / fix test6

 Lab 16 task.c   |  60 ++++++++++++++++++++++++++++++++++++++++++++------------
 Lab 16 task.exe | Bin 141410 -> 142031 bytes
 2 files changed, 47 insertions(+), 13 deletions(-)

commit 83236a3de4cb441ce22449d1a5ffafd8ecfc781d
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sat Mar 23 10:24:19 2024 +0300

    task5

 Lab 16 task.c   |  68 ++++++++++++++++++++++++++++++++++++++++++++++++--------
 Lab 16 task.exe | Bin 139223 -> 141410 bytes
 matrix.c        |   2 +-
 3 files changed, 60 insertions(+), 10 deletions(-)

commit 588ff05ebef3024c179391ea8782de2a316a22e6
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Sat Mar 23 10:05:04 2024 +0300

    fix task4

 Lab 16 task.c   |  59 ++++++++++++++++++++++++++++++++++++++++++++++++++------
 Lab 16 task.exe | Bin 138569 -> 139223 bytes
 2 files changed, 53 insertions(+), 6 deletions(-)

commit fc66c582e722ec286463e2d64e9739535c1f6a51
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Fri Mar 22 12:35:49 2024 +0300

    task3 / ref bug getmin

 Lab 16 task.c   |  59 ++++++++++++++++++++++++++++++++++++++++++++++++++++++--
 Lab 16 task.exe | Bin 137423 -> 138569 bytes
 matrix.c        |  55 ++++++++++++++++++++++++++++------------------------
:
```

```
commit 45a0b870666d6965decc6377e00ef952ab67efb2
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Thu Mar 21 17:39:27 2024 +0300

    add task2/test2/ref insertionsortrawsmatrixcriteria

 Lab 16 task.c   |  41 ++++++++++++++++++++++++++++++++++++++++-
 Lab 16 task.exe | Bin 136789 -> 137423 bytes
 matrix.c        |  39 ++++++++++++++++++++++-----------------
 matrix.exe      | Bin 0 -> 135700 bytes
 4 files changed, 62 insertions(+), 18 deletions(-)

commit cfa03f2a2ea3958dc88ba44d2a33f3c1dd16e3e6
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Thu Mar 21 17:09:20 2024 +0300

    task 1 lab16 / test / bug fix

 Lab 16 task.c   |  24 ++++++++++++++----------
 Lab 16 task.exe | Bin 136771 -> 136789 bytes
 matrix.exe      | Bin 135700 -> 0 bytes
 3 files changed, 14 insertions(+), 10 deletions(-)

commit 2cf2797c16c1668ef1c09acaca6e0f205950d3d2
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Thu Mar 21 17:08:03 2024 +0300

    task 1 lab16 / bug fix

 CMakeLists.txt                                     |   3 +-
 Lab 16 task.c                                      |  29 ++++++++++++++++
 Lab 16 task.exe                                    | Bin 0 -> 136771 bytes
 ...json => codemodel-v2-fd8fdc9c5efa26cbe0b4.json} |   2 +-
 ...55.json => index-2024-03-21T13-36-29-0030.json} |   4 +--
 ...et-LAB15Matrix-Debug-f3ca871ad783d5e80010.json} |  12 +++++--
 cmake-build-debug/.ninja_log                       |   2 +-
 cmake-build-debug/CMakeFiles/clion-Debug-log.txt   |   2 +-
 cmake-build-debug/build.ninja                      |   8 ++++-
 matrix.c                                           |  37 ++++++++++-----------
 matrix.exe                                         | Bin 135700 -> 135700 bytes
 11 files changed, 71 insertions(+), 28 deletions(-)

commit b518f50464e0a49c7b1f1c9ab8e48f3ccef98e74
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Wed Mar 13 12:03:03 2024 +0300

    add ncols and nrows = 0

 matrix.c   |   2 ++
 matrix.exe | Bin 135700 -> 135700 bytes
 2 files changed, 2 insertions(+)

commit 34297774fe77cc6b6bb977bd57452c0f33aeb499
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Wed Mar 13 11:49:57 2024 +0300

    fix func freememmatrix

 matrix.c   |  10 +++++-----
 matrix.exe | Bin 135700 -> 135700 bytes
 2 files changed, 5 insertions(+), 5 deletions(-)

commit 0733eae0f937a34baa800f7b320eb563402f4ca8
Author: Александр <alexanders.borchenko@gmail.com>
Date:   Tue Mar 12 19:12:16 2024 +0300

    bug fix / complete test

 matrix.c   | 238 +++++++++++++++++++++++++++++++++++++++++++++++++++++--------
 matrix.exe | Bin 132767 -> 135700 bytes
:
```

Вывод: в ходе выполнения лабораторной работы я получил навыки работы с многомерными массивами.