

Кафедра прикладной математики

По дисциплине: «Программирование»

## Работа с текстовыми файлами».

ПМ-24-1

---

группа

ПОДПИСЬ, ДАТА

Сараев А.П.

фамилия, инициалы

К.т.н., доцент кафедры ПМ

ученая степень, ученое звание

ПОДПИСЬ, ДАТА

Мирошников А.И.

фамилия, инициалы

Липецк 2024

## **Цель работы**

Освоить способы динамического выделения памяти и разработки функций на примере программирования задач линейной алгебры. Освоить работу с текстовыми файлами данных.

## **Задание кафедры**

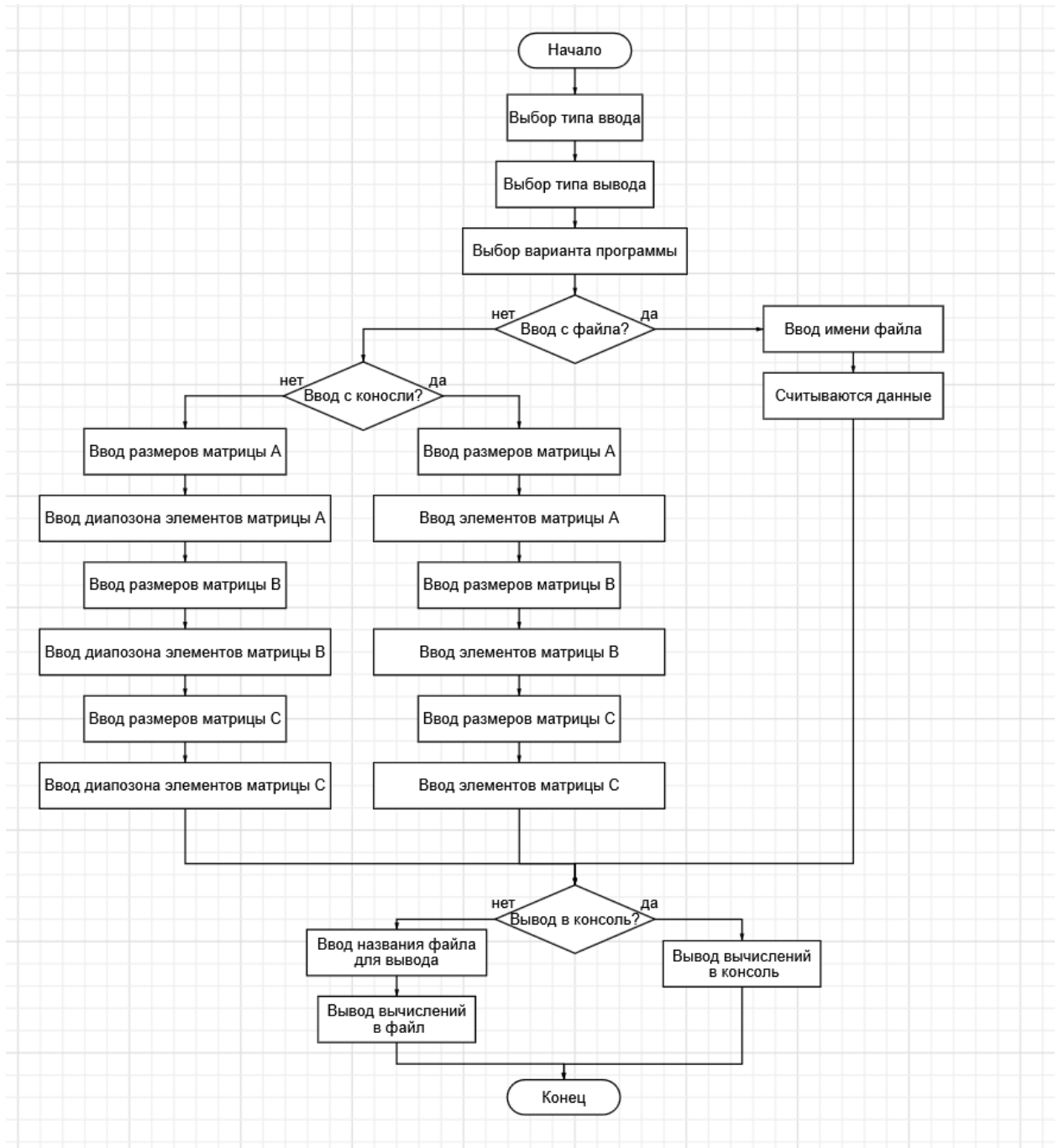
1. Реализовать вычисление заданного матричного выражения.
  2. Реализовать интерфейс для работы с пользователем. Программа должна осуществлять проверку некорректного ввода данных на всех этапах.
  3. Реализовать ввод матриц в трех вариантах по выбору пользователя: из файла (значения констант, размеры матриц и их содержимое записаны в файле), консольный ввод (значения констант, размеры матриц и их содержимое определяются пользователем), задание случайными числами (значения констант, размеры матриц определяются пользователем; разброс случайных значений задается пользователем). Имена файлов задаются пользователем. Имена файлов могут быть с пробелом.
  4. Реализовать вывод матриц, промежуточных и конечных результатов в двух вариантах по выбору пользователя: в файл, консольный вывод. Имена файлов задаются пользователем. Имена файлов могут быть с пробелом.
  5. Операции транспонирования, вычисления суммы, произведения на матрицу и на константу реализовать в виде отдельных функций.
- Необходимо контролировать размерность матриц при осуществлении матричных операций. Если размерность матриц не позволяет провести операцию, выдавать сообщение об ошибке.
6. Программа должна представлять результаты вычислений (в том числе промежуточных) в виде матрицы со значениями, выровненными по столбцам, например:
- 4 5 5  
345 54 2  
-102 4 85
7. Разработать 2 варианта работы программы:

1. Хранить матрицу в виде одномерного массива
2. Хранить матрицу в виде массива указателей

**Задания к лабораторной работе №1**

15	$(A+B/k)^T * C$
----	-----------------

## Блок-схема



## Текст программы

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <locale>
using namespace std;

////////////////////////////////////
////////////////////////////////////Общие функции////////////////////////////////////
////////////////////////////////////

void output_file_name(ofstream &fout) {
    if (fout.is_open()) {
        fout.close();
    }
    cout << "Введите название файла: ";
    bool f = 0;
    char* file_name = new char[256];
    while (cin.getline(file_name, 256)) {
        fout.open(file_name);
        if (!fout.is_open()) {
            if (f == 0) {
                f = 1;
                continue;
            }
            cout << "Не найден такой файл.\n";
            cout << "Введите название файла: ";
        }
        else
            break;
    }
}

void input_file_name(ifstream &fin) {
    if (fin.is_open()) {
        fin.close();
    }
    cout << "Введите название файла: ";
    char* file_name = new char[256];
    bool f = 0;
    while (cin.getline(file_name, 256)) {
        fin.open(file_name);
        if (!fin.is_open()) {
            if (f == 0) {
                f = 1;
                continue;
            }
            cout << "Не найден такой файл.\n";
            cout << "Введите название файла: ";
        }
        else
            break;
    }
}

int input_type() {
    int ch1;
    cout << "Выберете тип ввода данных: 1 - из файла, 2 - в консоли, 3 - случайные числа: ";
    while (cin >> ch1) {
        if (ch1 == 1) {
            break;
        }
        else if (ch1 == 2 || ch1 == 3)
            break;
    }
}
```

```

        else if (ch1 != 1 && ch1 != 2 && ch1 != 3) {
            cout << "Некоректный ввод.\n";
            cout << "Выберете тип ввода данных: 1 - из файла, 2 - в
консоли, 3 - случайные числа: ";
        }
    }
    return ch1;
}

int output_type() {
    int ch2;
    cout << "Выберете способ вывода результатов: 1 - в файл, 2 - в консоль:
";
    while (cin >> ch2) {
        if (ch2 == 1) {
            break;
        }
        else if (ch2 == 2)
            break;
        else if (ch2 != 1 && ch2 != 2) {
            cout << "Некоректный ввод.\n";
            cout << "Выберете способ вывода результатов: 1 - в файл, 2
- в консоль: ";
        }
    }
    return ch2;
}

int var_type() {
    int ch3;
    cout << "Выберете вариант программы: 1 - одномерный массив, 2 - массив
указателей: ";
    while (cin >> ch3) {
        if (ch3 == 1 || ch3 == 2) {
            break;
        }
        else{
            cout << "Некоректный ввод.\n";
            cout << "Выберете вариант программы: 1 - одномерный
массив, 2 - массив указателей: ";
        }
    }
    return ch3;
}

int k_const(int c, ifstream& fin) {
    int k;
    if (c == 1)
        fin >> k;
    else {
        cout << "Введите значение коснтанты (k != 0, |k| <= 20) k = ";
        while (cin >> k) {
            if (k != 0 && abs(k) <= 20)
                break;
            cout << "k != 0, |k| <= 20 !\n";
            cout << "Введите значение коснтанты (k != 0, |k| <= 20) k
= ";
        }
    }
    return k;
}

void input_matrix_size(int& size_x, int& size_y, int type, ifstream& fin,
char matr_name) {
    int a, b;

```

```

        if (type == 1) {
            fin >> a >> b;
            size_x = a, size_y = b;
            return;
        }
        else {
            cout << "Введите размер матрицы (от 1 до 10)" << matr_name << ": ";
            while (cin >> a >> b) {
                if (a >= 1 && a <= 10 && b >= 1 && b <= 10) {
                    size_x = a, size_y = b;
                    return;
                }
                else {
                    cout << "Неправильный размер матрицы!\n";
                    cout << "Введите размер матрицы (от 1 до 10)" <<
matr_name << ": ";
                }
            }
        }

    }

    bool check_add_sizes(int a_n, int a_m, int b_n, int b_m) {
        if (a_n != b_n || a_m != b_m) {
            cerr << "Несовпадающие размеры матриц.\n";
            return false;
        }
        return true;
    }

    bool check_mul_sizes(int a_n, int a_m, int b_n, int b_m) {
        if (a_m != b_n) {
            cerr << "Несовпадающие размеры матриц.\n";
            return false;
        }
        return true;
    }

    //////////////////////////////////////
    //////////////////////////////////////1 ВАРИАНТ (одномерный массив)////////////////////////////////////
    void generate_matrices(double* a, int a_n, int a_m, int l, int r) {
        for (int i = 0; i < a_n; i++) {
            for (int j = 0; j < a_m; j++) {
                a[i * a_m + j] = rand() % (r - l) + l;
            }
        }
    }

    bool input_matrix(double* matr, int size_x, int size_y, int type, ifstream
&fin, char matr_name) {
        if (type == 1) {
            for (int i = 0; i < size_x; i++) {
                for (int j = 0; j < size_y; j++) {
                    fin >> matr[i * size_y + j];
                    if (abs(matr[i * size_y + j]) > 20) {
                        cout << "Числа в матрице должны быть |a| <=
20 или пропущены некоторые значения!\n";
                        return false;
                    }
                }
            }
        }
        else if (type == 2) {
            cout << "Введите значения в матрице " << matr_name << ": \n";

```

```

        for (int i = 0; i < size_x; i++) {
            for (int j = 0; j < size_y; j++) {
                cin >> matr[i * size_y + j];
                if (abs(matr[i * size_y + j]) > 20) {
                    cout << "Числа в матрице должны быть |a| <=
20!\n";
                    return false;
                }
            }
        }
    }
    else{
        cout << "Введите диапазон чисел в матрице (от -20 до 20)" <<
matr_name << ": ";
        int l_a, r_a;
        while (cin >> l_a >> r_a) {
            if (abs(l_a) > 20 || abs(r_a) > 20) {
                cout << "Числа не могут быть вне диапазона.\n";
                cout << "Введите диапазон чисел в матрице (от -20 до
20)" << matr_name << ": ";
                continue;
            }
            if (l_a > r_a) {
                cout << "Левая граница не может быть больше
правой.\n";
                cout << "Введите диапазон чисел в матрице (от -20 до
20)" << matr_name << ": ";
                continue;
            }
            break;
        }
        generate_matrices(matr, size_x, size_y, l_a, r_a);
    }
    return true;
}

void add_matrices(double* a, int a_n, int a_m, double* b, int b_n, int b_m,
double* res) {
    if (a_n != b_n || a_m != b_m) {
        cerr << "Несовпадающие размеры матриц.";
        exit(0);
    }
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++)
            res[i * a_m + j] = a[i * a_m + j] + b[i * b_m + j];
    }
}

void mul_matr(double* a, int a_n, int a_m, double* b, int b_n, int b_m,
double* res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < b_m; j++) {
            for (int l = 0; l < b_n; l++) {
                res[i * b_m + j] += a[i * b_n + l] * b[l * b_m + j];
            }
        }
    }
}

void div_matr_by_num(double* a, int a_n, int a_m, int k, double* res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++)
            res[i * a_m + j] = a[i * a_m + j] / k;
    }
}

```



```

void transp_matr(double* a, int a_n, int a_m, double* res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++) {
            res[j * a_n + i] = a[i * a_m + j];
        }
    }
}

void print_mat(double* a, int a_n, int a_m, int c, ofstream &fout) {
    if (c == 1) {
        for (int i = 0; i < a_n; i++) {
            fout << setw(2);
            for (int j = 0; j < a_m; j++)
                fout << a[i * a_m + j] << setw(10);
            fout << '\n';
        }
    }
    else {
        for (int i = 0; i < a_n; i++) {
            cout << setw(2);
            for (int j = 0; j < a_m; j++)
                cout << a[i * a_m + j] << setw(10);
            cout << '\n';
        }
    }
}

//////////////////////////////////////
//////////////////////////////////////2 ВАРИАНТ (массив указателей)//////////////////////////////////////

void generate_matrices(double** a, int a_n, int a_m, int l, int r) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++) {
            a[i][j] = rand() % (r - l) + l;
        }
    }
}

bool input_matrix(double** matr, int size_x, int size_y, int type, ifstream&
fin, char matr_name) {
    if (type == 1) {
        for (int i = 0; i < size_x; i++) {
            for (int j = 0; j < size_y; j++) {
                fin >> matr[i][j];
                if (abs(matr[i][j]) > 20) {
                    cout << "Числа в матрице должрны быть |a| <=
20 или пропущены некоторые значения!\n";
                    return false;
                }
            }
        }
    }
    else if (type == 2) {
        cout << "Введите значения в матрице " << matr_name << ": \n";
        for (int i = 0; i < size_x; i++) {
            for (int j = 0; j < size_y; j++) {
                cin >> matr[i][j];
                if (abs(matr[i][j]) > 20) {
                    cout << "Числа в матрице должрны быть |a| <=
20!\n";
                    return false;
                }
            }
        }
    }
}

```

```

    }
    else {
        cout << "Введите диапазон чисел в матрице (от -20 до 20)" <<
matr_name << ": ";
        int l_a, r_a;
        while (cin >> l_a >> r_a) {
            if (abs(l_a) > 20 || abs(r_a) > 20) {
                cout << "Числа не могут быть вне диапазона.\n";
                cout << "Введите диапазон чисел в матрице (от -20 до
20)" << matr_name << ": ";
                continue;
            }
            if (l_a > r_a) {
                cout << "Левая граница не может быть больше
правой.\n";
                cout << "Введите диапазон чисел в матрице (от -20 до
20)" << matr_name << ": ";
                continue;
            }
            break;
        }
        generate_matrices(matr, size_x, size_y, l_a, r_a);
    }
    return true;
}

void add_matrices(double** a, int a_n, int a_m, double** b, int b_n, int b_m,
double** res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++)
            res[i][j] = a[i][j] + b[i][j];
    }
}

void mul_matr(double** a, int a_n, int a_m, double** b, int b_n, int b_m,
double** res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < b_m; j++) {
            for (int l = 0; l < b_n; l++) {
                res[i][j] += a[i][l] * b[l][j];
            }
        }
    }
}

void div_matr_by_num(double** a, int a_n, int a_m, double k, double** res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++)
            res[i][j] = a[i][j] / k;
    }
}

void transp_matr(double** a, int a_n, int a_m, double** res) {
    for (int i = 0; i < a_n; i++) {
        for (int j = 0; j < a_m; j++) {
            res[j][i] = a[i][j];
        }
    }
}

void print_mat(double** a, int a_n, int a_m, int c, ofstream& fout) {
    if (c == 1) {
        for (int i = 0; i < a_n; i++) {
            fout << setw(2);
            for (int j = 0; j < a_m; j++)

```

```

        fout << a[i][j] << setw(10);
        fout << '\n';
    }
}
else {
    for (int i = 0; i < a_n; i++) {
        cout << setw(2);
        for (int j = 0; j < a_m; j++)
            cout << a[i][j] << setw(10);
        cout << '\n';
    }
}
}

////////////////////////////////////
////////////////////////////////////

int main() {
    setlocale(LC_ALL, "Russian");
    ifstream fin;
    ofstream fout;
    while (1) {
        cout << '\n';
        int ch1 = input_type();
        if (ch1 == 1) input_file_name(fin);

        int ch2 = output_type();
        if (ch2 == 1) output_file_name(fout), fout << fixed <<
setprecision(4);
        else cout << fixed << setprecision(4);

        int ch3 = var_type();

        double k = k_const(ch1, fin);
        if (k == 0 || abs(k) > 20) {
            cout << "k не может быть равен 0 или |k| > 20 !\n";
            continue;
        }

        //////////////////////////////////

        int a_n, a_m, b_n, b_m, c_n, c_m;

        input_matrix_size(a_n, a_m, ch1, fin, 'A');
        if (a_n < 1 || a_n > 10 || a_m < 1 || a_m > 10) {
            cout << "Неправильный размер матрицы (от 1 до 10 столбцов
и колонок)!\n";
            continue;
        }
        double* a1 = new double[a_n * a_m];
        double** a2 = new double* [a_n];
        if (ch3 == 2) {
            delete[] a1;
            for (int i = 0; i < a_n; i++) {
                a2[i] = new double[a_m];
            }
        }
        else delete[] a2;
        if (ch3 == 1) {
            if (!input_matrix(a1, a_n, a_m, ch1, fin, 'A')) continue;
            else {
                if (ch2 == 1) fout << "Матрица A:\n";
                else cout << "Матрица A:\n";
                print_mat(a1, a_n, a_m, ch2, fout);
            }
        }
    }
}

```

```

    }
}
else { if (!input_matrix(a2, a_n, a_m, ch1, fin, 'A')) continue;
      else {
          if (ch2 == 1) fout << "Матрица A:\n";
          else cout << "Матрица A:\n";
          print_mat(a2, a_n, a_m, ch2, fout);
      }
}

input_matrix_size(b_n, b_m, ch1, fin, 'B');
if (b_n < 1 || b_n > 10 || b_m < 1 || b_m > 10) {
    cout << "Неправильный размер матрицы (от 1 до 10 столбцов
и колонок)!\n";
    continue;
}
double* b1 = new double[b_n * b_m];
double** b2 = new double* [b_n];
if (ch3 == 2) {
    delete[] b1;
    for (int i = 0; i < b_n; i++) {
        b2[i] = new double[b_m];
    }
}
else delete[] b2;
if (ch3 == 1) { if (!input_matrix(b1, b_n, b_m, ch1, fin, 'B'))
continue;
                else {
                    if (ch2 == 1) fout << "Матрица
B:\n";
                    else cout << "Матрица B:\n";
                    print_mat(b1, b_n, b_m, ch2,
fout);
                }
}
else { if (!input_matrix(b2, b_n, b_m, ch1, fin, 'B')) continue;
      else {
          if (ch2 == 1) fout << "Матрица B:\n";
          else cout << "Матрица B:\n";
          print_mat(b2, b_n, b_m, ch2, fout);
      }
}

input_matrix_size(c_n, c_m, ch1, fin, 'C');
if (c_n < 1 || c_n > 10 || c_m < 1 || c_m > 10) {
    cout << "Неправильный размер матрицы (от 1 до 10 столбцов
и колонок)!\n";
    continue;
}
double* c1 = new double[c_n * c_m];
double** c2 = new double* [c_n];
if (ch3 == 2) {
    delete[] c1;
    for (int i = 0; i < c_n; i++) {
        c2[i] = new double[c_m];
    }
}
else delete[] c2;
if (ch3 == 1) { if (!input_matrix(c1, c_n, c_m, ch1, fin, 'C'))
continue;
                else {
                    if (ch2 == 1) fout << "Матрица
C:\n";
                    else cout << "Матрица C:\n";

```

```

fout);
print_mat(c1, c_n, c_m, ch2,
}
}
else { if (!input_matrix(c2, c_n, c_m, ch1, fin, 'C')) continue;
else {
if (ch2 == 1) fout << "Матрица C:\n";
else cout << "Матрица C:\n";
print_mat(c2, c_n, c_m, ch2, fout);
}
}

////////////////////////////////////

if (ch2 == 1) fout << "B / k:\n";
else cout << "B / k:\n";
double* d1 = new double[b_n * b_m];
double** d2 = new double* [b_n];
if (ch3 == 2) {
delete[] d1;
for (int i = 0; i < b_n; i++) {
d2[i] = new double[b_m];
}
}
else delete[] d2;
if (ch3 == 1) {
div_matr_by_num(b1, b_n, b_m, k, d1);
print_mat(d1, b_n, b_m, ch2, fout);
}
else {
div_matr_by_num(b2, b_n, b_m, k, d2);
print_mat(d2, b_n, b_m, ch2, fout);
}

if (ch2 == 1) fout << "A + B / k:\n";
else cout << "A + B / k:\n";
if (!check_add_sizes(a_n, a_m, b_n, b_m))
continue;
double* e1 = new double[b_n * b_m];
double** e2 = new double* [b_n];
if (ch3 == 2) {
delete[] e1;
for (int i = 0; i < b_n; i++) {
e2[i] = new double[b_m];
}
}
else delete[] e2;
if (ch3 == 1) {
add_matrices(a1, a_n, a_m, d1, b_n, b_m, e1);
print_mat(e1, b_n, b_m, ch2, fout);
delete[] d1;
}
else {
add_matrices(a2, a_n, a_m, d2, b_n, b_m, e2);
print_mat(e2, b_n, b_m, ch2, fout);
for (int i = 0; i < b_n; i++)
delete[] d2[i];
delete[] d2;
}

if (ch2 == 1) fout << "(A + B / k)T\n";
else cout << "(A + B / k)T\n";
double* f1 = new double[b_m * b_n];
double** f2 = new double* [b_m];

```

```

if (ch3 == 2) {
    delete[] f1;
    for (int i = 0; i < b_m; i++) {
        f2[i] = new double[b_n];
    }
}
else delete[] f2;
if (ch3 == 1) {
    transp_matr(e1, a_n, a_m, f1);
    print_mat(f1, b_m, b_n, ch2, fout);
    delete[] e1;
}
else {
    transp_matr(e2, a_n, a_m, f2);
    print_mat(f2, b_m, b_n, ch2, fout);
    for (int i = 0; i < b_n; i++)
        delete[] e2[i];
    delete[] e2;
}

if (ch2 == 1) fout << "(A + B / k)T * C\n";
else cout << "(A + B / k)T * C\n";
if (!check_mul_sizes(b_m, b_n, c_n, c_m))
    continue;
double* g1 = new double[b_m * c_m];
double** g2 = new double* [b_m];
if (ch3 == 2) {
    delete[] g1;
    for (int i = 0; i < b_m; i++) {
        g2[i] = new double[c_m];
        for (int j = 0; j < c_m; j++)
            g2[i][j] = 0;
    }
}
else {
    delete[] g2;
    for (int i = 0; i < b_m; i++) {
        for (int j = 0; j < c_m; j++)
            g1[i * c_m + j] = 0;
    }
}
if (ch3 == 1) {
    mul_matr(f1, b_m, b_n, c1, c_n, c_m, g1);
    print_mat(g1, b_m, c_m, ch2, fout);
    delete[] f1;
    delete[] g1;
}
else {
    mul_matr(f2, b_m, b_n, c2, c_n, c_m, g2);
    print_mat(g2, b_m, c_m, ch2, fout);
    for (int i = 0; i < b_m; i++)
        delete[] f2[i];
    delete[] f2;
    for (int i = 0; i < b_m; i++)
        delete[] g2[i];
    delete[] g2;
}
if (ch3 == 1) { delete[] a1; delete[] b1; delete[] c1; }
else {
    for (int i = 0; i < a_n; i++)
        delete[] a2[i];
    delete[] a2;
    for (int i = 0; i < b_n; i++)
        delete[] b2[i];
    delete[] b2;
}

```

```
        for (int i = 0; i < c_n; i++)
            delete[] c2[i];
        delete[] c2;
    }
    break;
}
fin.close();
fout.close();
return 0;
}
```

## Контрольный пример

Консоль отладки Microsoft Visual Studio

Выберите тип ввода данных: 1 - из файла, 2 - в консоли, 3 - случайные числа: 1  
Введите название файла: input.txt  
Выберите способ вывода результатов: 1 - в файл, 2 - в консоль: 1  
Введите название файла: out.txt  
Выберите вариант программы: 1 - одномерный массив, 2 - массив указателей: 1

C:\Users\Андрей I\Desktop\ЛГТУ\Прога\Lab1\Debug\Lab1.exe (процесс 4676) завершил работу с кодом 0.  
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".  
Нажмите любую клавишу, чтобы закрыть это окно:

input.txt - Блокнот

Файл Правка Формат Вид Справка

3  
3 4  
1 2 3 4  
1 2 3 4  
1 2 3 4  
3 4  
4 5 6 7  
4 5 6 7  
4 5 6 7  
3 5  
-2 2 -8 4 -5  
-1 2 -3 4 -5  
-1 2 -3 4 -5

out.txt - Блокнот

Файл Правка Формат Вид Справка

Матрица A:  
1.0000 2.0000 3.0000 4.0000  
1.0000 2.0000 3.0000 4.0000  
1.0000 2.0000 3.0000 4.0000

Матрица B:  
4.0000 5.0000 6.0000 7.0000  
4.0000 5.0000 6.0000 7.0000  
4.0000 5.0000 6.0000 7.0000

Матрица C:  
-2.0000 2.0000 -8.0000 4.0000 -5.0000  
-1.0000 2.0000 -3.0000 4.0000 -5.0000  
-1.0000 2.0000 -3.0000 4.0000 -5.0000

B / k:  
1.3333 1.6667 2.0000 2.3333  
1.3333 1.6667 2.0000 2.3333  
1.3333 1.6667 2.0000 2.3333

A + B / k:  
2.3333 3.6667 5.0000 6.3333  
2.3333 3.6667 5.0000 6.3333  
2.3333 3.6667 5.0000 6.3333

(A + B / k)T  
2.3333 2.3333 2.3333  
3.6667 3.6667 3.6667  
5.0000 5.0000 5.0000  
6.3333 6.3333 6.3333

(A + B / k)T \* C  
-9.3333 14.0000 -32.6667 28.0000 -35.0000  
-14.6667 22.0000 -51.3333 44.0000 -55.0000  
-20.0000 30.0000 -70.0000 60.0000 -75.0000  
-25.3333 38.0000 -88.6667 76.0000 -95.0000

Консоль отладки Microsoft Visual Studio

Выберите тип ввода данных: 1 - из файла, 2 - в консоли, 3 - случайные числа: 3  
Выберите способ вывода результатов: 1 - в файл, 2 - в консоль: 2  
Выберите вариант программы: 1 - одномерный массив, 2 - массив указателей: 2  
Введите значение константы (k != 0, |k| <= 20) k = 9  
Введите размер матрицы (от 1 до 10) A: 3 4  
Введите диапазон чисел в матрице (от -20 до 20) A: -20 20  
Матрица A:  
-19.0000 7.0000 -6.0000 0.0000  
-11.0000 -16.0000 18.0000 18.0000  
-18.0000 4.0000 5.0000 5.0000

Введите размер матрицы (от 1 до 10) B: 3 4  
Введите диапазон чисел в матрице (от -20 до 20) B: -20 20  
Матрица B:  
-19.0000 7.0000 -19.0000 -9.0000  
15.0000 2.0000 7.0000 16.0000  
11.0000 -16.0000 2.0000 13.0000

Введите размер матрицы (от 1 до 10) C: 3 6  
Введите диапазон чисел в матрице (от -20 до 20) C: -20 20  
Матрица C:  
-8.0000 2.0000 1.0000 16.0000 18.0000 -5.0000  
-13.0000 -14.0000 -9.0000 -2.0000 9.0000 12.0000  
7.0000 -1.0000 15.0000 -6.0000 3.0000 -9.0000

B / k:  
-2.1111 0.7778 -2.1111 -1.0000  
1.6667 0.2222 0.7778 1.7778  
1.2222 -1.7778 0.2222 1.4444

A + B / k:  
-21.1111 7.7778 -8.1111 -1.0000  
-9.3333 -15.7778 18.7778 19.7778  
-16.7778 2.2222 5.2222 6.4444

(A + B / k)T  
-21.1111 -9.3333 -16.7778  
7.7778 -15.7778 2.2222  
-8.1111 18.7778 5.2222  
-1.0000 19.7778 6.4444

(A + B / k)T \* C  
172.7778 105.2222 -188.7778 -218.4444 -514.3333 144.5556  
158.4444 234.2222 183.1111 142.6667 4.6667 -248.2222  
-142.6667 -284.3333 -98.7778 -198.6667 38.6667 218.8889  
-204.0000 -285.3333 -82.3333 -94.2222 179.3333 184.3333

C:\Users\Андрей I\Desktop\ЛГТУ\Прога\Lab1\Debug\Lab1.exe (процесс 20300) завершил раб



## **Вывод**

1. Одномерный массив лучше подходит для задач, где важна скорость и минимальное использование памяти, так как в нем все ячейки идут непрерывно.
2. Массив указателей удобен при работе с динамическими или нерегулярными данными, где требуется гибкость, может состоят из массив разных длин, и изменение одного из них остальные не затрагиваются.