**Московский государственный технический**

**университет им. Н. Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы программирования»

Отчет по лабораторной работе № 8

«Обработка и печать числовых матриц»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б |  | преподаватель каф. ИУ5 |
| Поляков Леонид |  | Правдина А. Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Постановка задачи**

### Часть первая

Создать квадратную **динамическую матрицу** A размера N*N и заполнить её следующими значениями:

* все элементы главной диагонали равны 1;
* элементы, лежащие выше главной диагонали, вычисляются по формуле A_{i,j}=\cfrac{x^i}{(j!)^i} , а элементы, лежащие ниже главной диагонали, по формуле A_{i,j}=\cfrac{(-x)^i}{(j!)^i}, где i,j =1,2,...,N.

В зависимости от размера матрицы и ширины поля вывода элемента матрицы, обеспечить удобное для пользователя отображение матрицы на экране (с этим связано одно из требований функции вывода массива).

#### Требования к первой части

Обратите внимание, что для выполнения первой части необходимо создать указанные ниже функции и учесть для них определенные требования:

1. **Универсальная функция выделения памяти для двухмерного динамического массива** размера N*M (N - количество строк, N - количество столбцов):
2. **Функция заполнения матрицы** A:
   1. Для вычисления значений элементов матрицы использовать рекуррентные соотношения.
3. **Универсальная функция вывода двухмерного динамического массива** размера N*M.
   1. Функция должна быть реализована в самостоятельном модуле mprinter: объявление сигнатуры должно находиться в mprinter.h, а ее реализация - в mprinter.c или mprinter.cpp (в зависимости от языка программирования).
   2. Отображение строки массива не должно превышать рекомендуемую длину строки, описанную в [руководстве Google по стилю C++](https://habr.com/ru/post/499944/), а именно не более 80 символов. Рекомендуемое значение задать в виде константы в модуле mprinter. Если строка выходит за пределы, то необходимо обеспечить перенос непомещающихся столбцов так, как показано в примере ниже. Учесть, что длина столбцов не имеет ограничений.
   3. Матрица A и другие двухмерные массивы должны передаваться в функцию вывода через параметры.
   4. Функция вывода двумерного массива в качестве аргументов должна принимать признак того, в каком формате распечатывать значения элементов матрицы: в экспоненциальном формате (например, 1.23E+10) и в формате с фиксированной точкой с точностью precision. Заданную точность также передавать в качестве аргумента.
4. **Универсальная функция освобождения памяти**, выделенной для двухмерного динамического массива размера N*M.

### Часть вторая

Распечатать с помощью разработанной функции, используя **вспомогательный массив указателей на строки**, матрицу размером B[10][10], заданную с помощью оператора описания (**нединамическую**). Значение элементов матрицы B определяется соотношением: B[i][j]=i*10+j.

Вставьте в программу и объясните результаты выполнения операторов для матрицы B[10][10], представленных ниже.

#### C++

std::cout << B << std::endl;

system("Pause");

std::cout << B[4] << " " << \*B[2] << std::endl;

std::cout << &B[8] << " " << \*(\*B+5) << " " << \*(B[0]+6) << std::endl;

std::cout << (\*(\*(B + 8))) << " " << \*(B[1]+6) << std::endl;

std::cout << &B[2] + 1 << " " << B[7][3]++ << std::endl;

std::cout << B[1][20] << " " << \*(B[0] + 30) << std::endl;

#### Требования ко второй части

Обратите внимание, что:

1. для выполнения второй части необходимо создать указанные ниже функции:
   1. **Функция заполнения нединамической матрицы** B.
   2. **Функция получения вспомогательного массива указателей на строки матрицы** B.
2. матрицу B необходимо выводить в консоль с помощью вспомогательного массива указателей и разработанной функции вывода из модуля mprinter.

**Функции**:

* void create\_matrix(double\*\*& matrix, const unsigned N, const unsigned M) // выделение памяти под динамическую матрицу
  + matrix – ссылка на указатель на дин. массив элементов типа double\*[]
  + N, M – количество строк и столбцов матрицы
* void init\_dymanic\_matrix(double\*\* matrix, const unsigned N, const unsigned M) // инициализация динамической матрицы
  + matrix – указатель на динамический массив элементов типа double\*[]
  + N, M – количество строк и столбцов матрицы
* void init\_static\_matrix(double matrix[][constants::static\_Matrix\_Size], const unsigned N) // инициализация статической матрицы
  + matrix – указатель на статический массив типа double[constants::static\_Matrix\_Size][constants::static\_Matrix\_Size]
  + N – количество строк матрицы
* void print\_matrix(const double\* const matrix[], const unsigned N, const unsigned M, bool fixed, const unsigned precision) // функция форматированного вывода матрицы на экран
  + matrix – константный указатель на константный массив типа double\*[]
  + N, M – количество строк и столбцов передаваемой матрицы
  + Fixed – параметр, отвечающий за формат представления чисел. Значение true отвечает за формат с фиксированной точкой, значение false – за экспоненциальный формат
  + Precision – требуемая точность (знаков после запятой) вывода чисел
* void delete\_matrix(double\*\*& matrix, const unsigned N, const unsigned M) // функция освобождения памяти, выделенной динамически под матрицу
  + matrix – ссылка на указатель на массив типа double\*[]
  + N, M – количество строк и столбцов матрицы
* void get\_row\_pointers(double matrix[][constants::static\_Matrix\_Size], double\* row\_pointers[constants::static\_Matrix\_Size]) // функция инициализации массива указателей указателями на строки статической матрицы
  + matrix – указатель на статический массив типа double[constants::static\_Matrix\_Size][constants::static\_Matrix\_Size]
  + row\_pointers – указатель на массив типа double\*[]

**Разработка алгоритма**

# **первая часть**

В первой части лабораторной работы у пользователя запрашивается количество строк и столбцов матрицы, которую надо создать и заполнить по приведённым в ТЗ формулам, формат представления чисел (с фиксированной точной или экспоненциальный) и требуемая точность для отображения элементов матрицы. После этого матрица выводится на экран и удаляется

# **вторая часть**

Во второй части создаётся статическая матрица 10x10 и статический массив, хранящий указатели на строки этой матрицы. Матрица выводится на экран.

Далее выводится адрес первого элемента матрицы. После чего требуется понять, что выведет программа, основываясь на коде из ТЗ.

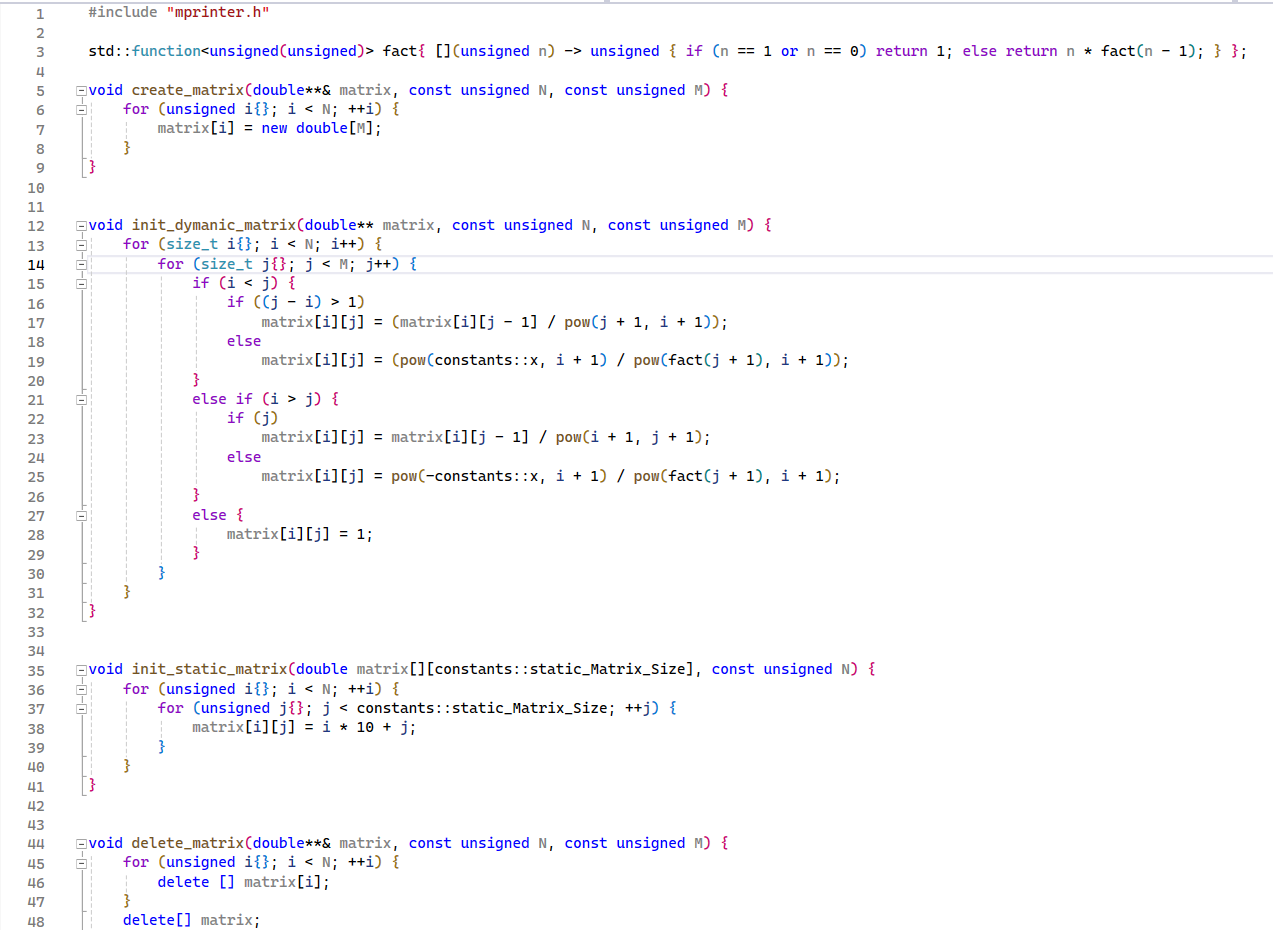
**Текст программы**

**constants.hИзображение выглядит как текст, Шрифт, число, линия

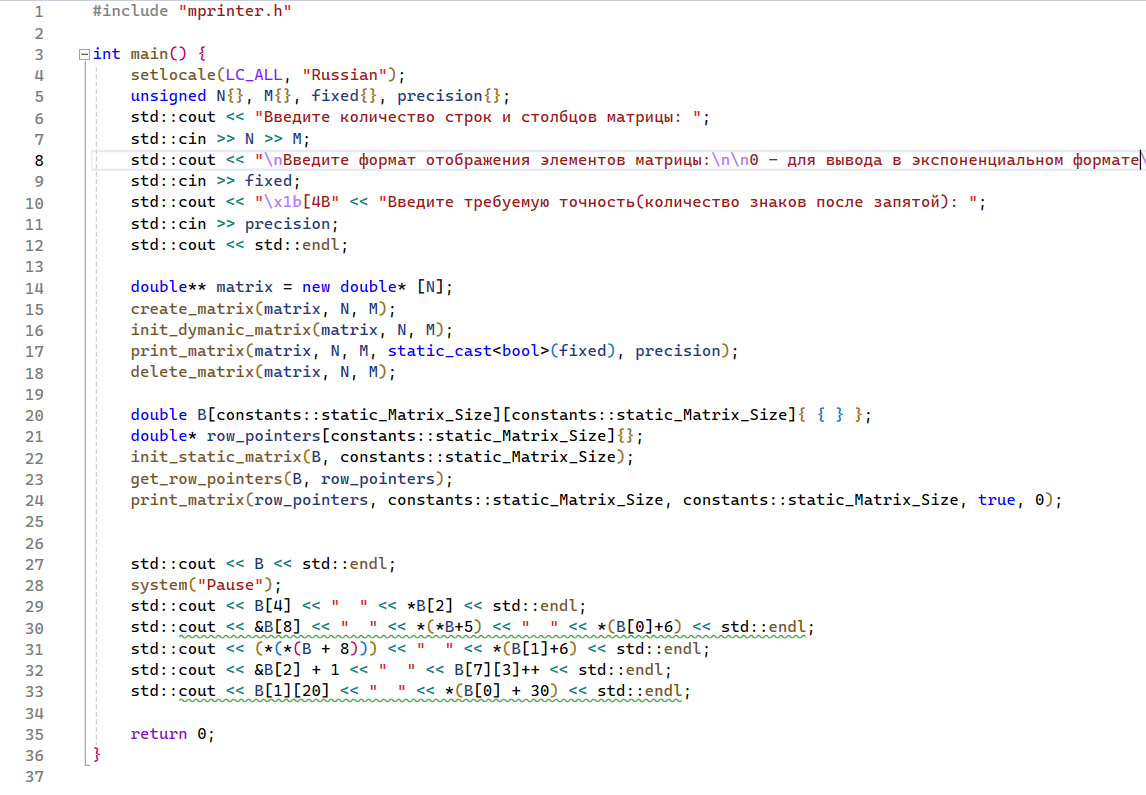
Автоматически созданное описание**

**mprinter.hИзображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание**

**mprinter.cppИзображение выглядит как текст, снимок экрана

Автоматически созданное описание**

**main.cpp**

**Анализ результатов**

**Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание**

**Результат работы программы соответствует ожиданиям и полностью удовлетворяет требованиям ТЗ**

**Вывод**

Элементы статических массивов хранятся в памяти последовательно, друг за другом, одним большим куском памяти. Динамические массивы могут хранить указатели на элементы, находящиеся в разных местах в памяти, но оператор new всегда выделяет память одним куском.

**Использованные источники**

* **Веб-сайт кафедры ИУ5, курс «основы программирования»**

[**https://cpp1.wiki.iu5edu.ru**](https://cpp1.wiki.iu5edu.ru)

* **Веб-сайт кафедры ИУ5 «автоматизированные системы обработки информации и управления»**

**https://e-learning.bmstu.ru/iu5/mod/folder/view.php?id=277**