**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №7

«Обработка и печать числовой матрицы»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-15Б |  | преподаватель каф. ИУ5 |
| Коновалов Илья |  | Папшев И.С. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

**Задача:**

Разработать функцию для вывода на экран матрицы double matr[N][M] для распечатки двухмерных массивов. Размерность матрицы может быть произвольной.

Функция должна обеспечивать удобное для пользователя отображение матрицы на экране в (фиксированном (fixed) или )научном (scientific) формате с требуемой точностью ***n*** знаков после запятой (устанавливается setprecision(n)). Целые числа также выводятся в научном формате.

Использовать разработанную функцию для печати квадратной матрицы А размером N\*N (где N вводится с клавиатуры).

- все элементы главной диагонали матрицы А равны% 1;

- элементы, лежащие выше главной диагонали, вычисляются по формуле

**A i,j = xi / (j!)i ,**

а элементы, лежащие ниже главной диагонали, по формуле

**A i,j = (-x)i / (j!)i**, где **i,j =1,2,…,N**; **x=1**.

Для вычисления значений элементов матрицы использовать ***рекуррентные соотношения*** (без факториалов и степеней).Создание и заполнение матрицы реализовать в виде ***отдельных функций***.

Распечатать с помощью разработанной функции целочисленную статическую (заданную с помощью оператора описания) матрицу размером **B[10][10].** Для передачи матрицы в функцию печати создайте вспомогательный массив указателей на строки матрицы **B** . Значения элементов матрицы **В** определяются соотношением: **B[i][j]=i\*10+j**.

Вставить в программу и объяснить результаты выполнения следующих операторов

для матрицы В[10][10]:

cout<<B<<" "<<B[0]<<" "<<B[2]<<endl;

cout<<B[0][0]<<" "<<\*\*B <<" "<<\*B[0]<<endl;

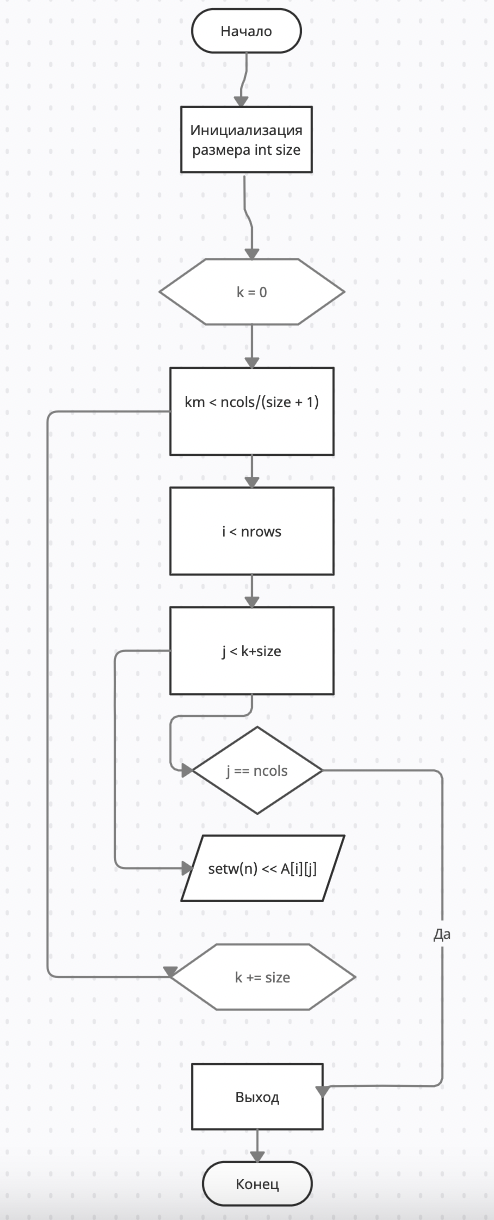
cout<<\*(\*(B+1))<<" "<<\*B[1]<<endl;

cout<<\*(B[0]+1)<<" " <<\*(\*B+1)<<endl;

cout<<B[0][20]<<" "<<\*(B[0]+20)<<" "<<\*B[2]<<endl;

**Алгоритм:**

Блок-схема функции для печати матрицы



**Код программы:**

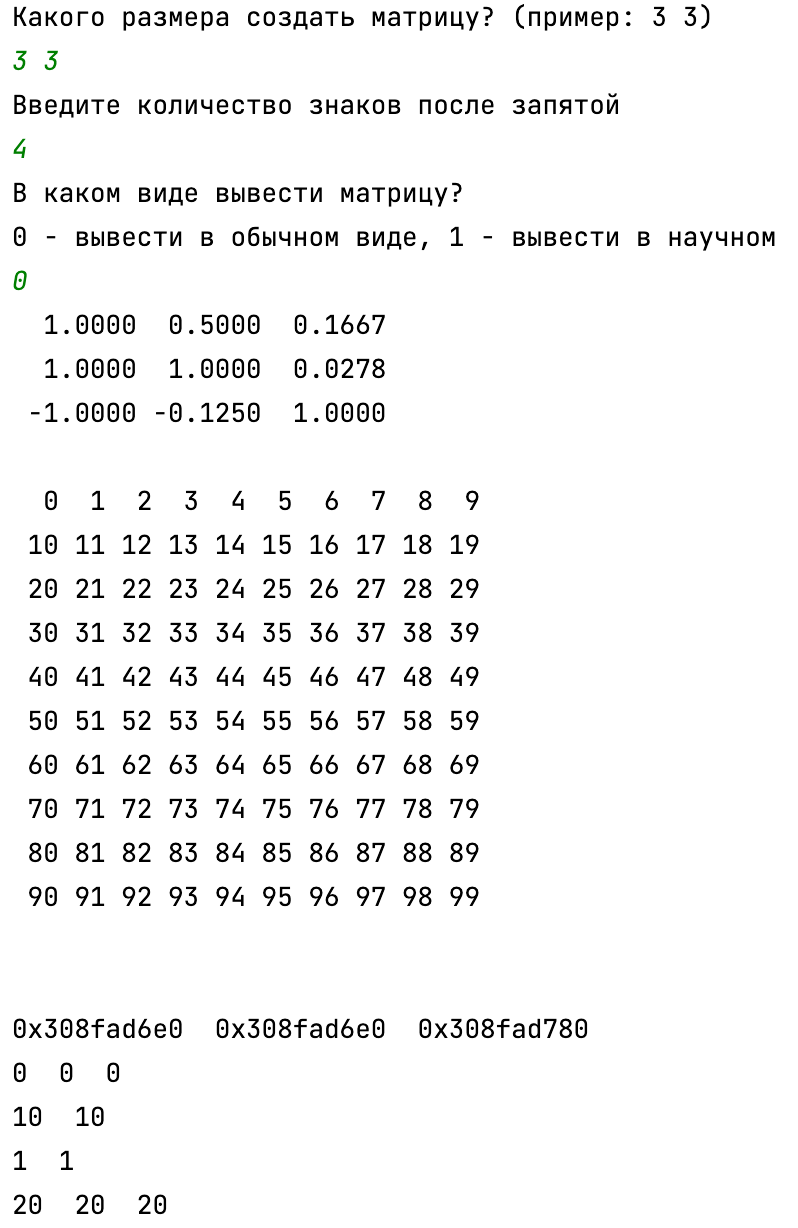
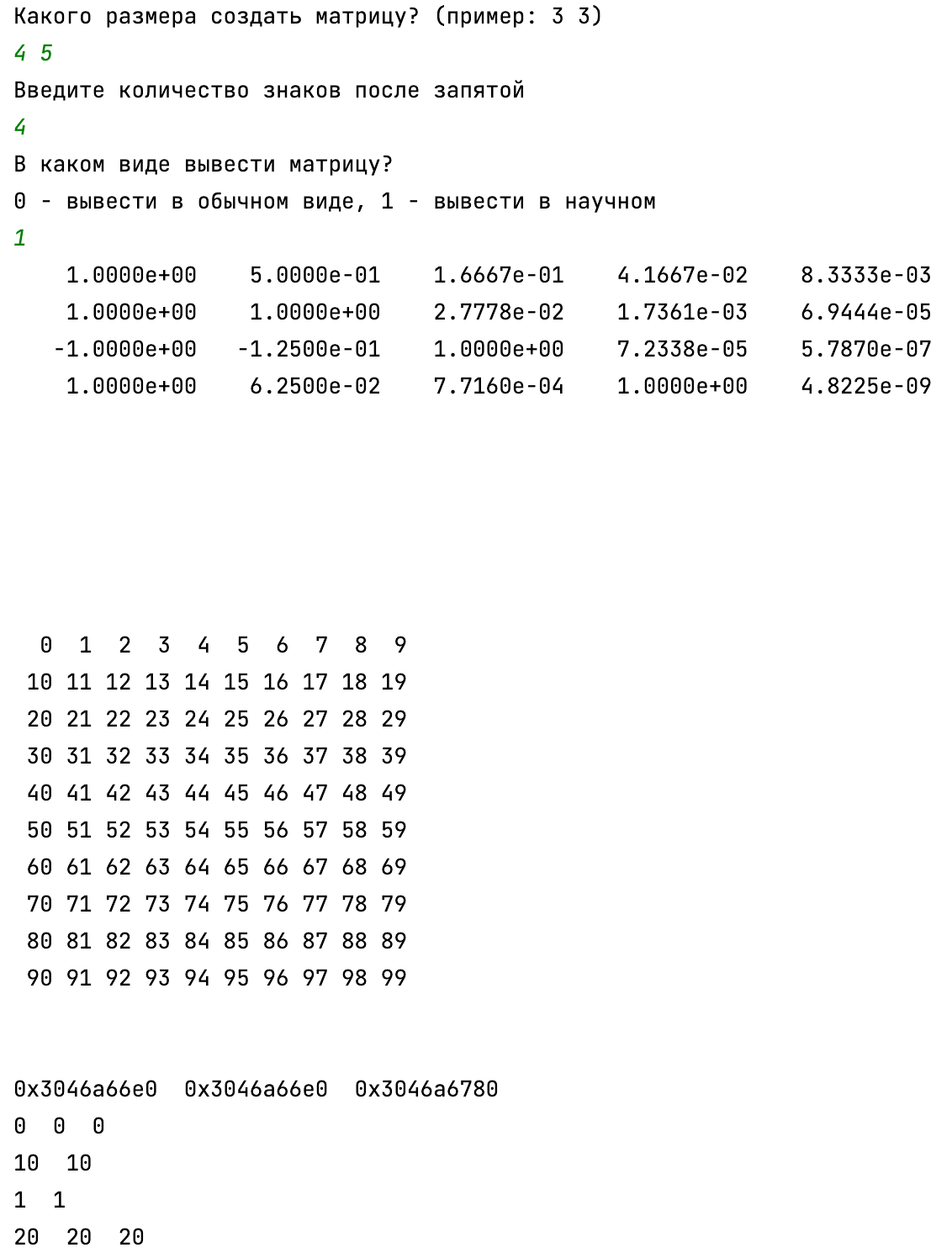
Основная программа:

#include **<iostream>**#include **<cmath>**#include **<iomanip>**#include **"PrintMatrix.h"  
using namespace** std;  
**typedef double** (\*TPF)(**double**);  
**void** FillMatrix(**double**\*\* A, **int** nrows, **int** ncols);  
**double** fact(**double** x);  
**int** main() {  
 **int** point;  
 **int** type;  
 **int** n = 0;  
 cout << **"Какого размера создать матрицу? (пример: 3 3)"** << **'\n'**;  
 **int** nrows, ncols;  
 cin >> nrows >> ncols;  
 cout << **"Введите количество знаков после запятой"** << **'\n'**;  
 cin >> point; *//nrows ncols* cout << **"В каком виде вывести матрицу?"** << **'\n'**;  
 cout << **"0 - вывести в обычном виде, 1 - вывести в научном"** << **'\n'**;  
 **do** cin >> type;  
 **while** ((type != 0) && (type != 1));  
 **if** (type == 1) {  
 cout << scientific << setprecision(point);  
 n = 10 + point;  
 } **else** {  
 cout << fixed << setprecision(point);  
 n = 4 + point;  
 }  
 **double** \*\*A = **new double** \*[nrows];  
 **for** (**int** i = 0; i < nrows; i++)  
 A[i] = **new double**[ncols];  
 FillMatrix(A, nrows, ncols);  
 PrintMatrix(A, nrows, ncols, n, type, point);  
 **double** B[10][10];  
 **for** (**int** i = 0; i < 10; i++) {  
 **for** (**int** j = 0; j < 10; j++) {  
 B[i][j] = i \* 10 + j;  
 }  
 }  
 **int** rowB = 10, colB = 10, nB = 3, typeB = 0, pointB = 0;  
 **double** \*\*C = **new double** \*[10];  
 **for** (**int** i = 0; i < 10; i++) {  
 C[i] = B[i];  
 }  
 cout << setprecision(0) << fixed;  
 PrintMatrix(C, rowB, colB, nB, typeB, pointB);  
 cout << endl;  
  
 cout << B << **" "** << B[0] << **" "** << B[2] << endl;  
 cout << B[0][0] << **" "** << \*\*B << **" "** << \*B[0] << endl;  
 cout << \*(\*(B + 1)) << **" "** << \*B[1] << endl;  
 cout << \*(B[0] + 1) << **" "** << \*(\*B + 1) << endl;  
 cout << B[0][20] << **" "** << \*(B[0] + 20) << **" "** << \*B[2] << endl;  
 system(**"pause"**);  
 **return** 0;  
}  
**void** FillMatrix(**double**\*\* A, **int** nrows, **int** ncols) {  
 **int** k = 0;  
 **for** (**int** i = 0; i < nrows; i++) {  
 **for** (**int** j = 0; j < ncols; j++) {  
 k++;  
 **if** (i == j) A[i][j] = 1.;  
 **else if** (j > i) {  
 **if** (i == 0)  
 A[i][j] = A[i][j - 1] / (j+1);  
 **else** A[i][j] = A[i - 1][j] \* A[0][j];  
 }  
 **else if** (i > j) {  
 **if** (i - 1 == j **and** j != 0) {  
 A[i][j] = A[i - 2][j] \* A[0][j] \* A[0][j];  
 }  
 **else** A[i][j] = abs(A[i - 1][j] \* A[0][j]);  
 **if** (i % 2== 0)  
 A[i][j] \*= -1;  
 }  
 }  
 }  
}  
**double** fact(**double** x) {  
 **if** (x == 0)  
 **return** 0;  
 **if** (x == 1)  
 **return** 1;  
  
 **return** x \* fact(x - 1);  
}

Функция для печати матрицы:

#pragma once  
**template** <**typename** T> **void**\* PrintMatrix(T\*\* A, **int**& nrows, **int**& ncols, **int** n, **int**& type, **int**& point) {  
 **int** size = 80 / n;  
 **int** k = 0;  
 **for** (**int** km = 0; km < (ncols / size + 1); km++) {  
 **for** (**int** i = 0; i < nrows; i++) {  
 **for** (**int** j = k; j < (k + size); j++) {  
 **if** (j == ncols) **break**;  
 std::cout << std::setw(n) << A[i][j];  
 }  
 std::cout << **'\n'**;  
 }  
 k += size;  
 std::cout << **'\n'**;  
 }  
}

**Результаты:**

**Вывод:**

В ходе выполнения лабораторной работы я приобрел следующие навыки программирования:

- создание двухмерных динамических массивов;

-использование файлов для хранения матриц;

- форматированный вывод матриц на экран;

- обработка матриц;

- передача двухмерных массивов в функцию;

- разработка шаблона функции печати числовых матриц;

- доступ к элементам матрицы через указатели и с помощью индексов;

- освоение технологии структурного программирования.