**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

Отчет по лабораторной работе №8

«**Обработка и печать числовой матрицы**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б  Казицин Алексей |  | преподаватель каф. ИУ5  Козлов А.Д. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

Постановка задачи

Создать квадратную матрицу A размером N\*N (где N вводится с клавиатуры), и заполнить её следующими значениями:

- все элементы главной диагонали 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**.

Для вычисления значений элементов матрицы использовать рекуррентные соотношения**.**

Реализовать алгоритм заполнения матрицы в виде функции.

В зависимости от размера матрицы и ширины поля вывода элемента матрицы, обеспечить удобное для пользователя отображение матрицы на экране. Оформить вывод матрицы размером N\*M на экран в виде функции с целью использования ее в последующих лабораторных работах для распечатки двумерных массивов.

Матрица должна передаваться в разрабатываемые функции через параметры.

Не изменяя кода функции вывода матрицы, распечатать матрицу в «научном» формате и в формате с фиксированной точкой с точностью 8 знаков после запятой.

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 **В** определяется соотношением: **B[i][j]=i\*10+j**.

Объясните, как передаются матрицы A и В в функцию вывода матриц на экран.

Вставьте в программу и объясните 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;

Разработка алгоритма

**Описание используемых переменных и функций с указанием наименования, типа (int, float, и т.п.) и назначения в программе:**

int point – количество знаков после запятой; int n – количество символов, умещающихся в одной строке одного столбца матрицы; int str – количество строк в матрице; int stolb – количество столбцов в матрице; double\*\* A – основная матрица, с которой работаем; double B[10][10] – матрица 10\*10 от 0 до 99; double\*\* C- динамический массив копирующий матрицу B[10][10]; i, j, km – счетчики в циклах; int size – количество столбцов, которые можно разместить в одном ряду; int k – вспомогательная переменная;

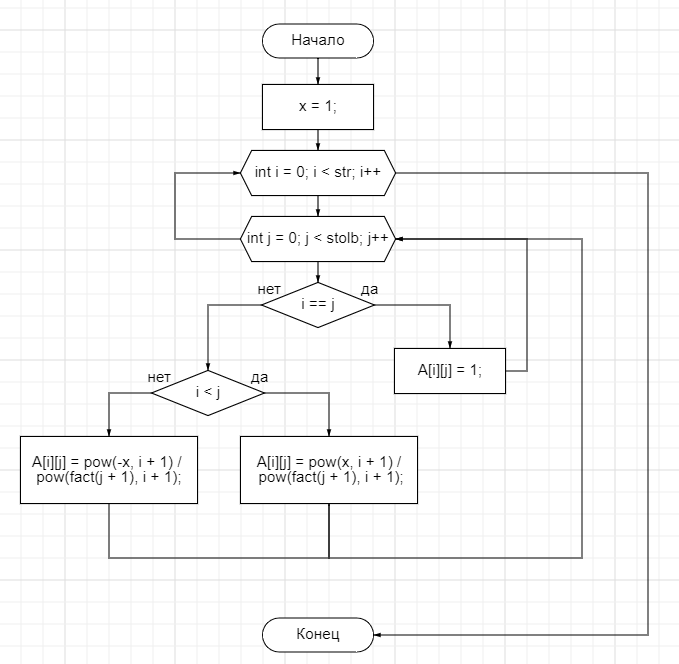
zapoln(double\*\* A, int str, int stolb, TPF fact) – заполняет матрицу по формуле, данной в условии.

fact(double x) – функция, которая высчитывает факториал числа.

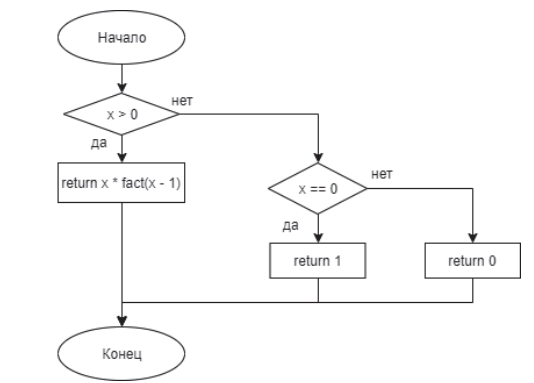
out(double\*\* A, int str, int stolb, int n) – функция вывода матрицы на экран

Блок-схема алгоритма

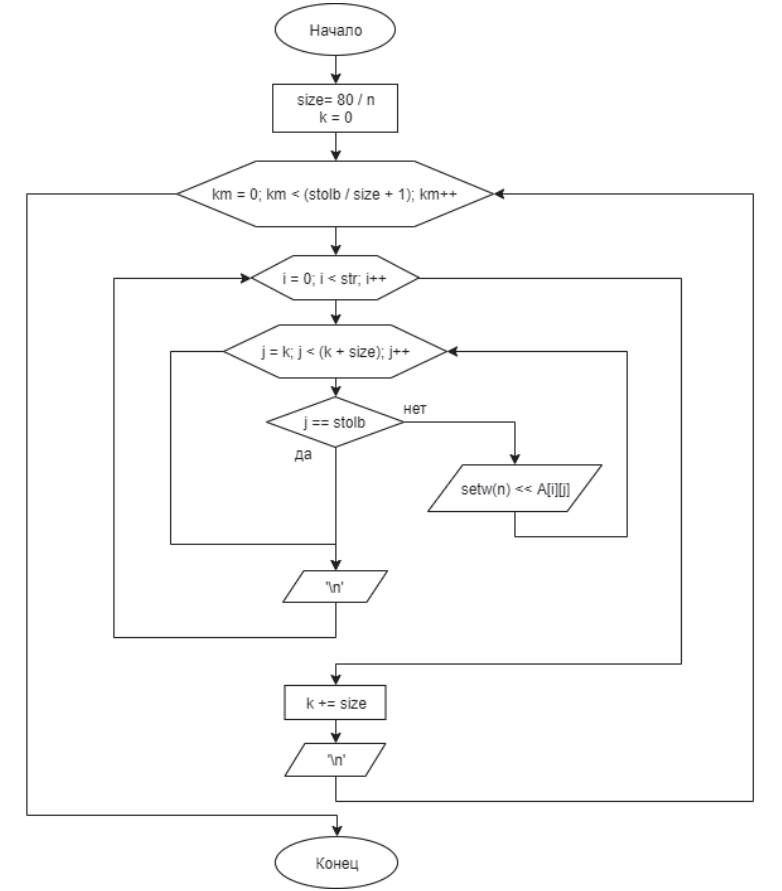
Функция zapoln(double\*\* A, int str, int stolb, TPF fact)



Функция fact(double x)



Функция out(double\*\* A, int str, int stolb, int n)



Текст программы

#include <iostream>

#include <cmath>

#include <iomanip>

#include <windows.h>

using namespace std;

typedef double (\*TPF)(double);

void zapoln(double\*\* A, int str, int stolb, TPF fact);

void out(double\*\* A, int nrow, int ncol, int n);

double fact(double x);

void zapoln(double\*\* A, int str, int stolb, TPF fact){

int x = 1;

for (int i = 0; i < str; i++)

for (int j = 0; j < stolb; j++){

if (i == j)

A[i][j] = 1;

else

if (i < j)

A[i][j] = pow(x, i + 1) / pow(fact(j + 1), i + 1);

else

A[i][j] = pow(-x, i + 1) / pow(fact(j + 1), i + 1);

}

}

double fact(double x){

if (x > 0)

return x \* fact(x - 1);

else

if (x == 0)

return 1;

else return 0;

}

void out(double\*\* A, int str, int stolb, int n){

int size = 80 / n;

int k = 0;

for (int km = 0; km < (stolb / size + 1); km++){

for (int i = 0; i < str; i++){

for (int j = k; j < (k + size); j++){

if (j == stolb) break;

cout << setw(n) << A[i][j];

}

cout << '\n';

}

k += size;

cout << '\n';

}

}

int main(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

int point = 8; //количество знаков после запятой

int n = 0; //количество символов, умещающихся в одной строке одного столбца

cout << "Введите размер квадратной матрицы:" << '\n';

int str, stolb;

cin >> str;

stolb = str;

cout << "Научный формат:" << "\n";

cout << scientific << setprecision(point);

n = 10 + point;

double\*\* A = new double\* [str];

for (int i = 0; i < str; i++)

A[i] = new double[stolb];

zapoln(A, str, stolb, fact);

out(A, str, stolb, n);

cout << "Фиксированный формат:" << "\n";

cout << fixed << setprecision(point);

n = 4 + point;

out(A, str, stolb, n);

cout << endl << endl;

//Вторая часть

double B[10][10];

for (int i = 0; i < 10; i++){

for (int j = 0; j < 10; j++){

B[i][j] = i \* 10 + j;

}

}

double\*\* C = new double\* [10];

for (int i = 0; i < 10; i++){

C[i] = B[i];

}

cout << setprecision(0) << fixed;

out(C, 10, 10, 3);

cout << '\n';

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;

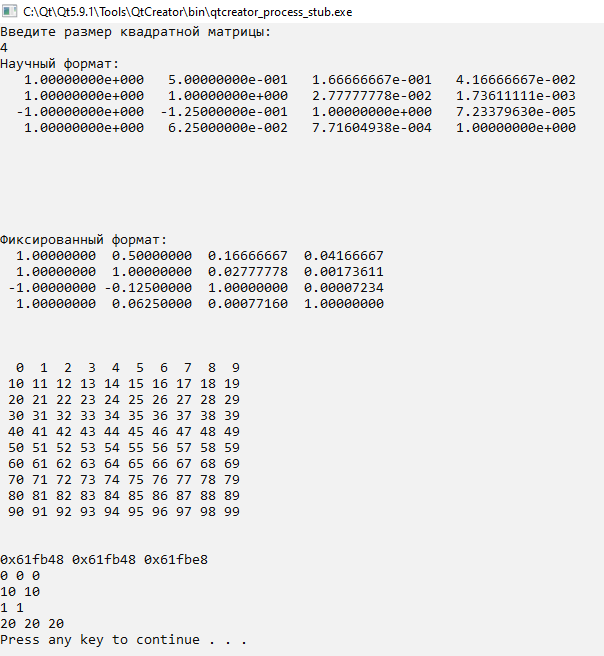
system("pause");

return 0;

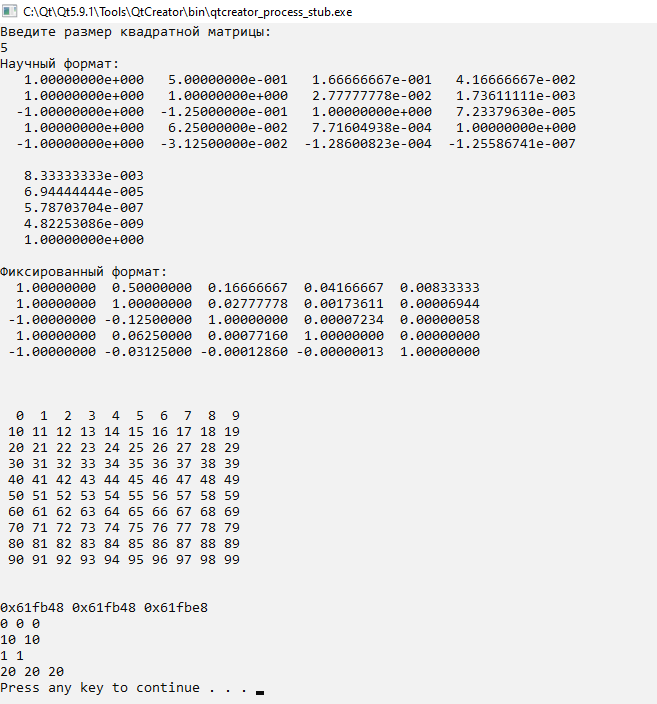
}

Анализ результатов

Размер матрицы – 4



Размер матрицы – 5



Размер матрицы – 6

