**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: Чернев Николай Андреевич |  | Проверил: |
| --- | --- | --- |
| студент группы ИУ5-14Б |  | преподаватель каф. ИУ5 |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Задание**

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 переменных

Функция filling\_matrix:

* int x - переменная для заполнения матрицы по формуле
* int i, j - счетчики циклов

Функция print\_matrix:

* int cnt\_in\_line - кол-во столбцов, которые помещаются в одном ряду
* int i, j - счетчики циклов

Функция main:

* int N - размер динамического двумерного массива
* double \*\*a - динамический двумерный массив, заполняемый по рекуррентной формуле
* double b[10][10] - статический двумерный массив
* int i, j - счетчики циклов
* int type - тип вывода(1 - научный, 2 - с фикс точкой)
* int acc - точность выводимых значений (кол-во знаков после запятой)

Описание функций

int factorial(int k) - принимает k, возвращает k!(факториал)

void filling\_matrix(double \*\*a, int N) - принимает двумерный массив a, его размер N, и заполняет эту матрицу по рекуррентной формуле, ничего не возвращает

template<typename T> - позволяет на этапе компиляции создавать разные экземпляры функции print\_matrix с таким типом вместо T, который подается в функцию при вызове

void print\_matrix(T a, int N, int type, int acc) - принимает двумерный массив а(типа T, чтобы можно было подавать как статические массивы, так и динамические), его размер N, тип вывода type и точность выводимых значений(кол-во знаков после запятой) acc. Выводит на экран этот двумерный массив таким образом, что те столбцы, которые не поместились в первый ряд, выводятся ниже, ничего не возвращает

**Код программы**

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int factorial(int k){

return (k == 0 || k == 1) ? 1 : factorial(k - 1) \* k;

}

void filling\_matrix(double \*\*a, int N){

int x = 1;

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

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

if (i < j){

if (i == 0) {

a[i][j] = a[i][j - 1] / pow(j + 1, i + 1);

}

else{

a[i][j] = a[i - 1][j] \* x / factorial(j + 1);

}

}

else if (i > j) {

if (j == 0) {

a[i][j] = (i == 1 ? -1 : 1) \* a[i - 1][j] \* (-x) / factorial(j + 1);

}

else{

a[i][j] = (i == 1 ? -1 : 1) \* a[i][j - 1] / pow(j + 1, i + 1);

}

}

else{

a[i][j] = 1;

}

}

}

}

template<typename T>

void print\_matrix(T a, int N, int type, int acc){

int cnt\_in\_line = min(160 / (acc + 8) + 1, N);

if (type == 1){

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

for (int j = 0; j < cnt\_in\_line; j++){

cout << setw(acc + 7) << fixed << setprecision(acc) << scientific << a[i][j] << ' ';

}

cout << '\n';

}

cout << '\n';

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

for (int j = cnt\_in\_line; j < N; j++){

cout << setw(acc + 7) << fixed << setprecision(acc) << scientific << a[i][j] << ' ';

}

cout << '\n';

}

}

if (type == 2) {

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

for (int j = 0; j < cnt\_in\_line; j++) {

cout << setw(acc + 4) << fixed << setprecision(acc) << a[i][j] << ' ';

}

cout << '\n';

}

cout << '\n';

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

if (cnt\_in\_line >= N) {

break;

}

for (int j = cnt\_in\_line; j < N; j++) {

cout << setw(acc + 4) << fixed << setprecision(acc) << a[i][j] << ' ';

}

cout << '\n';

}

}

}

int main() {

int N;

cout << "Введите N - размер матрицы\n";

cin >> N;

double \*\*a = new double\* [N];

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

a[i] = new double[N];

}

filling\_matrix(a, N);

int type, acc;

cout << "Введите 1(научный вывод) или 2(вывод с фикс точкой)\n";

cin >> type;

cout << "Введите точность значений\n";

cin >> acc;

print\_matrix(a, N, type, acc);

double b[10][10];

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

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

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

}

}

cout << "Введите 1(научный вывод) или 2(вывод с фикс точкой)\n";

cin >> type;

cout << "Введите точность значений\n";

cin >> acc;

print\_matrix(b, 10, type, acc);

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

delete[] a[i];

}

delete[] a;

return 0;

}

**Анализ результатов**

