Московский государственный технический

университет им. Н.Э. Баумана

Курс «Основы информатики»

Отчёт по лабораторной работе №8

Обработка печати матрицы

Проверил:

Преподаватель каф. ИУ5

Аксёнова М.В.

Подпись и дата:

Факультет «Информатика и система управления»

Кафедра ИУ5 «Система обработки информации и управления»

Выполнил:

Студент группы ИУ5-13Б

Пермяков Дмитрий Кириллович

Подпись и дата:

**Постановка задачи.**

***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**.

Переменные:

int rows, cols; double x = 1 - rows- строки, cols- колонки, x- введённое число

auto\*\* matrix- двумерный массив размером rows на cols.

double B[10][10]- вторая статистическая матрица по условии задачи

Функции:

int fact(int num)- функция факториала

void FillArr(double\*\* arr, int rows, int cols, int x)- заполняет двумерный массив по формуле

void PrintArr(double\*\* arr, int rows, int cols, double x) – печатает матрицу в научном и фиксированном виде

**Код программы**

**Заголовки:**

#include <iostream>

#include <iomanip>

#include <math.h>

#include <ctime>

int fact(int num);

void PrintArr(double\*\* arr, int rows, int cols, double x);

void FillArr(double\*\* arr, int rows, int cols, int x);

Функции:

#include "Header.h"

int fact(int num)

{

if (num == 1 || num == 0)

return 1;

else

return (fact(num - 1) \* num);

}

void FillArr(double\*\* arr, int rows, int cols, int x)

{

for (int i = 1; i <= rows; ++i)

{

for (int j = 1; j <= cols; ++j)

{

if (i - 1 == j - 1)

arr[i - 1][j - 1] = 1;

else

{

if (j - 1 > i - 1)

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

else

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

}

}

}

}

void PrintArr(double\*\* arr, int rows, int cols, double x)

{

size\_t precision;

bool typeMatrix;

std::cout << "Choose the matrix print type: scientific or fixed (1/0): ";

std::cin >> typeMatrix;

std::cout << "Enter the precision of the calculations. Precision = ";

std::cin >> precision;

size\_t length\_row;

if (typeMatrix)

length\_row = precision + 8;

else if (!typeMatrix)

length\_row = precision + 3;

size\_t cols\_new = 81 / length\_row;

size\_t printed\_cols = 0;

while (printed\_cols < cols)

{

for (size\_t i = 0; i < rows; ++i)

{

for (size\_t j = 0; j < cols\_new; ++j)

{

if (typeMatrix)

{

if (arr[i][j + printed\_cols] < 0)

std::cout << std::scientific << std::setprecision(precision) << arr[i][j + printed\_cols] << " ";

else

std::cout << std::scientific << std::setprecision(precision) << ' ' << arr[i][j + printed\_cols] << " ";

if (printed\_cols + j == cols - 1)

break;

}

else if (!typeMatrix)

{

if (arr[i][j + printed\_cols] < 0)

std::cout << std::fixed << std::setprecision(precision) << arr[i][j + printed\_cols] << " ";

else

std::cout << std::fixed << std::setprecision(precision) << ' ' << arr[i][j + printed\_cols] << " ";

if (printed\_cols + j == cols - 1)

break;

}

}

std::cout << std::endl;

}

printed\_cols += cols\_new;

std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

<< "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n";

}

}

MAIN:

#include "Header.h"

int main()

{

srand(time(NULL));

int rows, cols; double x = 1;

std::cout << "Enter the number of \nrows= "; std::cin >> rows;

std::cout << "columns= "; std::cin >> cols;

auto\*\* matrix = new double\* [rows];

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

matrix[i] = new double[cols];

FillArr(matrix, rows, cols, x);

PrintArr(matrix, rows, cols, x);

double B[10][10]{0};

for (size\_t i = 0; i < 10; i++)

{

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

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

}

auto B2 = new double\* [10];

for (size\_t i = 0; i < 10; i++)

B2[i] = B[i];

std::cout << std::setw(36) << " AARAY B \n";

PrintArr(B2, 10, 10, x);

// удаляем содержимое динамич массива

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

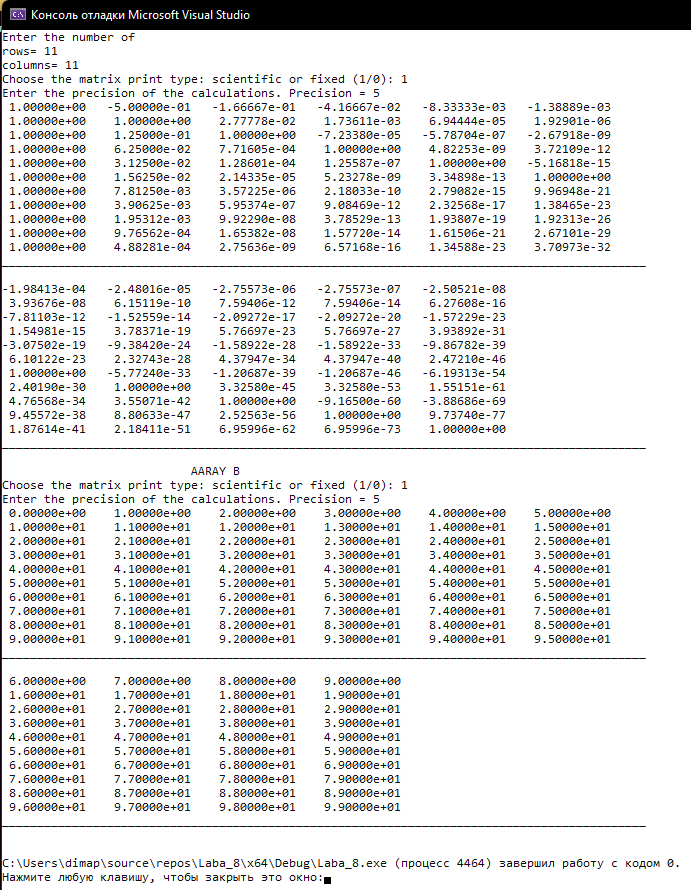
delete[] matrix[i];

delete[] matrix;

delete[] sum;

}

**Тестирование:**

****

