

Отчёт по лабораторной работе №23 по курсу «Фундаментальная информатика».

Выполнил студент группы М80-109Б-22: **Филиппов Александр Михайлович** № по списку **23**.

Контакты: a.filippov04@yandex.ru

Работа выполнена: «17» декабря 2024 г.

Преподаватель: каф. 806 Аносова Наталья Павловна

Отчет сдан «___» _____ 20__ г., итоговая оценка ___

1. Тема: **Обработка матриц.**

2. Цель работы: Составить программу на Си, производящую обработку квадратной матрицы порядка $N \times N$ из целых чисел, вводимой из стандартного ввода.

3. Задание: (вариант №23).

4. Оборудование: Оборудование ПЭВМ студента, если использовалось:

Процессор AMD Ryzen 7 4800HS with Radeon Graphic

ОП 8 ГБ

SSD 512 ГБ

Монитор 1920x1080

5. Программное обеспечение: Программное обеспечение ПЭВМ студента, если использовалось:

Операционная система семейства linux

Наименование версия ubuntu 22.04 jammy

Интерпретатор команд bash версия 5.1.16(1)-release

Система программирования VSCode

6. Идея, метод, алгоритм решения задачи [в формах: словесной, псевдокода, графической (блок-схема, диаграмма, рисунок, таблица) или формальные спецификации с пред- и постусловиями]:

Заполняем матрицу размера $n * n$ по спирали к центру начиная с верхнего правого угла числами от 1 до $n*n$ и выводим.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию).

Входные данные	Выходные данные	Описание тестируемого случая
3	<div>Матрица: 3 2 1 4 9 8 5 6 7</div>	Матрица 3 на 3
4	<div>Матрица: 4 3 2 1 5 14 13 12 6 15 16 11 7 8 9 10</div>	Матрица 4 на 4

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int n; // Объявляем переменную n для хранения размера матрицы

    printf("Введите размерность матрицы n: "); // Выводим приглашение для ввода размера матрицы
```

```

scanf("%d", &n); // Считываем размер матрицы с клавиатуры и со-
храняем в переменную n

int **matrix = (int **)malloc(n * sizeof(int *)); // Выделяем память для массива указате-
лей на строки матрицы

for (int i = 0; i < n; i++) // Цикл для выделения памяти под каждую строку матрицы
{
    matrix[i] = (int *)malloc(n * sizeof(int)); // Выделяем память для строки матрицы
}

int num = 1; // Инициализируем переменную num для
хранения текущего числа, которым будем заполнять матрицу
int top = 0, bottom = n - 1, left = 0, right = n - 1; // Инициализируем границы матрицы
для спирального заполнения
int dir = 0; // Инициализируем переменную dir для
определения направления движения (0 - вправо, 1 - вниз, 2 - влево, 3 - вверх)

while (top <= bottom && left <= right) // Пока границы не пересеклись
{
    if (dir == 0) // Заполнение строки справа налево
    {
        for (int i = right; i >= left; i--)
        {
            // Проходим по строке от правого края до левого
            matrix[top][i] = num++; // Записываем текущее число в ячейку и увеличиваем num
        }
        top++; // Сдвигаем верхнюю границу
        dir = 1; // Меняем направление на вниз
    }
    else if (dir == 1) // Заполнение столбца сверху вниз
    {
        for (int i = top; i <= bottom; i++)
        {
            // Проходим по столбцу от верхней границы до нижней
            matrix[i][left] = num++; // Записываем текущее число в ячейку и увеличиваем
num
        }
        left++; // Сдвигаем левую границу
        dir = 2; // Меняем направление на влево
    }
    else if (dir == 2) // Заполнение строки слева направо
    {
        for (int i = left; i <= right; i++)
        {
            // Проходим по строке от левого края до правого
            matrix[bottom][i] = num++; // Записываем текущее число в ячейку и увеличиваем
num
        }
        bottom--; // Сдвигаем нижнюю границу
        dir = 3; // Меняем направление на вверх
    }
    else if (dir == 3) // Заполнение столбца снизу вверх
    {
        for (int i = bottom; i >= top; i--)
        {
            // Проходим по столбцу от нижней границы до верхней

```

```

        matrix[i][right] = num++; // Записываем текущее число в ячейку и увеличиваем
num
    }
    right--; // Сдвигаем правую границу
    dir = 0; // Меняем направление на вправо
}
}

printf("Матрица:\n"); // Выводим сообщение о начале вывода матрицы
for (int i = 0; i < n; i++) // Цикл для вывода строк матрицы
{
    for (int j = 0; j < n; j++) // Цикл для вывода элементов каждой строки
    {
        printf("%3d ", matrix[i][j]); // Выводим элемент матрицы с форматированием для вы-
равнивания
    }
    printf("\n"); // Переходим на новую строку после вывода каждой строки матрицы
}

// Освобождаем выделенную память
for (int i = 0; i < n; i++) // Цикл для освобождения памяти, выделенной под каждую строку
матрицы
{
    free(matrix[i]); // Освобождаем память, выделенную под строку
}
free(matrix); // Освобождаем память, выделенную под массив указателей на строки

return 0; // Завершаем программу без ошибок
}

```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы: замечания отсутствуют.

11. Вывод:

Весьма простая задача, трудностей, как и особой заинтересованности не возникло

Подпись студента _____