

# Отчет по лабораторной работе №14 по курсу фундаментальная информатика

Студент группы М8О-101Б-20 Ядров Артем Леонидович, № по списку 28

Контакты www, e-mail, icq, skype temayadrow@gmail.com

Работа выполнена: «    » \_\_\_\_\_ 202\_\_ г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Вложенные циклы с параметрами. Обход и линеаризация матриц.

2. **Цель работы:** Составить программу ввода квадратной матрицы и печати в строку всех ее элементов в заданном порядке следования (обхода).

3. **Задание (вариант № 28):**

10	9	8	7
11	16	15	6
12	13	14	5
1	2	3	4

4. **Оборудование (лабораторное):**

ЭВМ Intel Pentium G2140, процессор 3.30 GHz, имя узла сети Cameron с ОП 8096 Мб, НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel core i5 2.60 GHz с ОП 8096 Мб, НМД 131072 Мб. Монитор dell  
Другие устройства \_\_\_\_\_

5. **Программное обеспечение (лабораторное):**

Операционная система семейства Unix, наименование Ubuntu версия 18.15.0  
интерпретатор команд bash версия 4.4.20

Система программирования GNU версия 5.8.13

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы gcc, cat

Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных stud/208104

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix, наименование Fedora версия 32  
интерпретатор команд bash версия 5.0.17

Система программирования Clion версия 2020.2.1

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы gcc, cat

Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере home/Temich

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

1. Будем считывать число n (размерность матрицы), пока не считаем конец файла
2. Заметим, что нам предстоит обойти квадрат, вершины которого меняются. При чем вершины стороны квадрата параллельны осям, поэтому будем хранить 2 переменные для каждой оси. Также будем хранить переменные, отражающие наше текущее положение, а также скорость нашего движения.
3. Каждый раз, когда мы доходим до граничной точки, будем сдвигать соответствующую границу и обновлять направление скорости. Будем идти до тех пор, пока границы по оси x (по столбцам) не станут равными.
4. Остается вывести последний элемент, так как условие при нахождении на нем условие не выполняется.

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

Тесты:

Input: 1

1

2

4 3

1 2

3

7 6 3

8 9 4

1 2 3

4

10 9 8 7

11 16 15 6

12 13 14 5

1 2 3 4

5

13 12 11 10 9

14 23 22 21 8

15 24 25 20 7

16 17 18 19 6

1 2 3 4 5

Output:

Введенная матрица:

1

Напечатанная строка: 1

Введенная матрица:

4 3

1 2

Напечатанная строка: 1 2 3 4

Введенная матрица:

7 6 3

8 9 4

1 2 3

Напечатанная строка: 1 2 3 4 5 6 7 8 9

Введенная матрица:

10 9 8 7

11 16 15 6

12 13 14 5

1 2 3 4

Напечатанная строка: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Введенная матрица:

13 12 11 10 9

14 23 22 21 8

15 24 25 20 7

16 17 18 19 6

1 2 3 4 5

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 2

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).

[Temi4@localhost labs]\$ cat lab14.c

/\*Лабораторная работа №14. Вариант 28

\* Выполнил студент группы М80-101Б-20 Ядров А. Л. \*/

```
#include <stdio.h>

int main() {
    int a[7][7], n = 1;
    while (n){
        printf("Введите размерность матрицы\n");
        if (scanf("%d", &n) == EOF){
            break;
        }
        printf("Введите матрицу размера %d^%d\n", n, n);
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                scanf("%d", &a[i][j]); //считывание входных данных
            }
        }
        printf("Введенная матрица:\n");
        for (int i = 0; i < n; i++){
            for (int j = 0; j < n; j++){
                printf("%-10d ", a[i][j]);
            }
            printf("\n"); //перевод строки
        }
        printf("Напечатанная строка: ");
        int x_left = 0, x_right = n - 1, y_lower = n - 1, y_upper = 0; //инициализация границ
        int curx = 0, cury = n - 1, vx = 1, vy = 0; //инициализация текущих координат и скорости движения
        while (x_left != x_right) {
            printf("%d ", a[cury][curx]);
            curx += vx; // очередное передвижение
            cury += vy;
            if (curx == x_right && cury == y_lower) { //перемена направления движения
                vx = 0;
                vy = -1;
                if (y_lower != y_upper) {
                    y_lower--;
                    printf("%d ", a[cury][curx]);
                    cury--;
                } else {
                    printf("%d ", a[cury][curx]);
                    vx = -1;
                    vy = 0;
                    continue;
                }
            }
        }
        if (curx == x_right && cury == y_upper) {
            vx = -1;
            vy = 0;
            if (x_right != x_left) {
                x_right--;
                printf("%d ", a[cury][curx]);
                curx--;
            }
        }
        if (curx == x_left && cury == y_upper) {
            vx = 0;
            vy = 1;
            if (y_upper != y_lower) {
                y_upper++;
                printf("%d ", a[cury][curx]);
                cury++;
            }
        }
        if (curx == x_left && cury == y_lower) {
```

```

        vx = 1;
        vy = 0;
        if (x_left != x_right) {
            x_left++;
            printf("%d ", a[cury][curx]);
            curx++;
        }
    }
}
printf("%d\n", a[y_upper][x_left]);
}
return 0;
}

```

[Temi4@localhost labs]\$ gcc lab14.c

[Temi4@localhost labs]\$ ./a.out

Введите размерность матрицы

1

Введите матрицу размера 1^1

1

Введенная матрица:

1

Напечатанная строка: 1

Введите размерность матрицы

2

Введите матрицу размера 2^2

4 3

2 1

Введенная матрица:

4 3

2 1

Напечатанная строка: 2 1 3 4

Введите размерность матрицы

3

Введите матрицу размера 3^3

7 6 3

8 9 4

1 2 3

Введенная матрица:

7 6 3

8 9 4

1 2 3

Напечатанная строка: 1 2 3 4 3 6 7 8 9

Введите размерность матрицы

4

Введите матрицу размера 4^4

10 9 8 7

11 16 15 6

12 13 14 5

1 2 3 4

Введенная матрица:

10 9 8 7

11 16 15 6

12 13 14 5

1 2 3 4

Напечатанная строка: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Введите размерность матрицы

5

Введите матрицу размера 5^5

13 12 11 10 9

14 23 22 21 8

15 24 25 20 7

16 17 18 19 6

1 2 3 4 5

Введенная матрица:

13 12 11 10 9

14 23 22 21 8

15 24 25 20 7

16 17 18 19 6

1       2       3       4       5  
Напечатанная строка: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
Введите размерность матрицы

**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 \_\_\_\_\_

---



---



---

#### 11. Выводы

Я научился составлять программу, написанную на языке Си, ввода квадратной матрицы и печати ее элементов в порядке следования

---



---



---



---

Недочёты при выполнении задания могут быть устранены следующим образом:

---



---



---

Подпись студента \_\_\_\_\_