

# Отчет по лабораторной работе № 14 по курсу Архитектура компьютера и информационных сетей

Студент группы М8О-103Б-22 Киселев Артём Олегович, № по списку 10

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

Работа выполнена: 2022 г.

Преподаватель: доцент каф. 806 Никулин С.П.

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

Отчет сдан « 10 » декабря 2022 г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

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

2. **Цель работы:** Научиться работать с вложенными циклами, делать обход и линеаризацию матриц.

3. **Задание (вариант № 12):** Составить программу ввода квадратной матрицы и печати в строку всех её элементов в заданном порядке следования

4. **Оборудование (лабораторное):**  
ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ Мб,  
НМД \_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_. Принтер \_\_\_\_\_  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Ryzen 3 3200u 2.6GHz с ОП 8 ГБ НМД SSD 256 ГБ, HDD 1000 ГБ . Монитор Встроенный 1920x1080

Другие устройства Touchpad Synaptics

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

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
Утилиты операционной системы \_\_\_\_\_

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

Местонахождение и имена файлов программ и данных \_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX, наименование Ubuntu версия 22.04.1  
интерпретатор команд bash версия 5.1.16

Система программирования C версия \_\_\_\_\_

Редактор текстов Visual Studio Code версия 28.2

Утилиты операционной системы \_\_\_\_\_

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

C - компилируемый статически типизированный язык программирования общего назначения.

gcc - компилятор.

Visual Studio Code - текстовый редактор.

Алгоритм:

- 1) Считываем количество строк (столбцов) n
- 2) Заполняем массив размером n \* n,
- 3) Начинаем обход в заданном порядке, проходим по каждому элементу заполненного в п.2 массива
  - 3.1) Если этот элемент находится в 1 столбце и в 1 строке, то выводим его, после этого если мы имеем матрицу размером 1x1, то завершаем проход.
  - 3.2) Есть два состояния: 1) когда идём по диагонали влево вниз, 2) когда идём по диагонали вправо, вверх. Изначально мы находимся в 1)
  - 3.3) Если мы в состоянии один, то, пока можем идти по диагонали влево вниз, идём в этом направлении. После этого если можем пойти вниз, то вниз, иначе вправо. Меняем состояние на 2)
  - 3.4) Если мы в состоянии два, то, пока можем идти по диагонали вправо вверх, идём в этом направлении. После этого если можем пойти вправо, то вправо, иначе вниз. Меняем состояние на 1)

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

- 1) Составить и отладить программу на Си.
- 2) Сделать протокол.
- 3) Сделать отчет.

Тесты:

```

1 1
2 Ввод:
3 1
4 0
5 Вывод:
6 0
7 2
8 Ввод:
9 2
10 0 1
11 2 3
12 Вывод:
13 0 2 1 3
14 3
15 Ввод:
16 3
17 0 1 2
18 3 4 5
19 6 7 8
20 Вывод:
21 0 3 1 2 4 6 7 5 8
22 4
23 Ввод:
24 4
25 1 3 4 10
26 2 5 9 11
27 6 8 12 15
28 7 13 14 16
29 Вывод:
30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
31 5
32 Ввод:
33 5
34 0 1 2 3 4
35 5 6 7 8 9
36 10 11 12 13 14
37 15 16 17 18 19
38 20 21 22 23 24
39 Вывод:
40 0 5 1 2 6 10 15 11 7 3 4 8 12 16 20 21 17 13 9 14 18 22 23 19 24

```

```

6)
Ввод:
6
0 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 26 27 28 29
30 31 32 33 34 35
Вывод:
0 6 1 2 7 12 18 13 8 3 4 9 14 19 24 30 25 20 15 10 5 11 16 21 26 31 32 27 22 17 23 28 33 34 29 35
7)
Ввод:
7
0 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 26 27
28 29 30 31 32 33 34
35 36 37 38 39 40 41
42 43 44 45 46 47 48
Вывод:
0 7 1 2 8 14 21 15 9 3 4 10 16 22 28 35 29 23 17 11 5 6 12 18 24 30 36 42 43 37 31 25 19 13 20 26 32 38 44 45 39 33 27 34 40 46 47 41 48

```

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)

```
Labs > lab14 > C 14.c > main()
1  // Вариант 12
2  // 1 3 4 10
3  // 2 5 9 11
4  // 6 8 12 15
5  // 7 13 14 16
6  #include <stdio.h>
7  #include <math.h>
8  #include <ctype.h>
9
10 int main() {
11     // Считываем количество строк (столбцов)
12     int n;
13     scanf("%d",&n);
14     // Заполняем массив размером n * n
15     int i = 1, j = 1, k = 0, c = 0, state = 0, t = 0;
16     int m[n * n];
17     while (t < n * n) {
18         scanf("%d", &m[t]);
19         t++;
20     }
21
22     // Начинаем обход в заданном порядке
23     while (k < n * n) {
24         if (i == 1 && j == 1) {
25             c = (i - 1) * n + (j - 1);
26             k++;
27             printf("%d ", m[c]);
28             if (n == 1) {
29                 break;
30             }
31         }
32         if (state == 0) {
33             while (j != 1 && i != n) {
34                 j--;
35                 i++;
36                 k++;
37                 c = (i - 1) * n + (j - 1);
38                 printf("%d ", m[c]);
39             }
40             if (i < n) {
41                 i++;
42                 k++;
43                 c = (i - 1) * n + (j - 1);
44                 printf("%d ", m[c]);
45             } else {
46                 j++;
47                 k++;
48                 c = (i - 1) * n + (j - 1);
49                 printf("%d ", m[c]);
50             }
51             state = 1;
52     }
```

```

53         if (state == 1) {
54             while (j != n && i != 1) {
55                 j++;
56                 i--;
57                 k++;
58                 c = (i - 1) * n + (j - 1);
59                 printf("%d ", m[c]);
60             }
61             if (j < n) {
62                 j++;
63                 k++;
64                 c = (i - 1) * n + (j - 1);
65                 printf("%d ", m[c]);
66             } else {
67                 i++;
68                 k++;
69                 c = (i - 1) * n + (j - 1);
70                 printf("%d ", m[c]);
71             }
72             state = 0;
73         }
74     }
75     printf("\n");
76
77
78
79
80
81     return 0;
82 }

```

Labs > lab14 > ≡ file1

```

1  1
2  0

```

Labs > lab14 > ≡ file2

```

1  2
2  0 1
3  2 3

```

Labs > lab14 > ≡ file3

```

1  3
2  0 1 2
3  3 4 5
4  6 7 8

```

Labs > lab14 > ≡ file4

```

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

```

Labs > lab14 > ≡ file5

```

1  5
2  0 1 2 3 4
3  5 6 7 8 9
4  10 11 12 13 14
5  15 16 17 18 19
6  20 21 22 23 24

```

Labs > lab14 > ≡ file6

```

1  6
2  0 1 2 3 4 5
3  6 7 8 9 10 11
4  12 13 14 15 16 17
5  18 19 20 21 22 23
6  24 25 26 27 28 29
7  30 31 32 33 34 35

```

Labs > lab14 > ≡ file7

```

1  7
2  0 1 2 3 4 5 6
3  7 8 9 10 11 12 13
4  14 15 16 17 18 19 20
5  21 22 23 24 25 26 27
6  28 29 30 31 32 33 34
7  35 36 37 38 39 40 41
8  42 43 44 45 46 47 48

```

```

• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file1
0
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file2
0 2 1 3
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file3
0 3 1 2 4 6 7 5 8
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file5
0 5 1 2 6 10 15 11 7 3 4 8 12 16 20 21 17 13 9 14 18 22 23 19 24
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file6
0 6 1 2 7 12 18 13 8 3 4 9 14 19 24 30 25 20 15 10 5 11 16 21 26 31 32 27 22 17 23 28 33 34 29 35
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab14$ ./14 < file7
0 7 1 2 8 14 21 15 9 3 4 10 16 22 28 35 29 23 17 11 5 6 12 18 24 30 36 42 43 37 31 25 19 13 20 26 32 38 44 45 39 33 27 34 40 46 47 41 48

```

**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 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:** в процессе работы я научился основам работы с вложенными циклами, производить обход матриц

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента **ЯО**