

Отчет по лабораторной работе № 15 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 « 11 » декабря 2022 г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Обработка матриц

2. **Цель работы:** Научиться обрабатывать квадратные матрицы

3. **Задание (вариант № 12):** Циклический сдвиг элементов матрицы по спирали по (против) часовой стрелки на 3 элемента

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)

2) Заполняем массив размером $n * n$

3) Начинаем обход в цикле.

3.1) Совершаем проход вправо и вниз, запоминая при этом предыдущие значения и меняя нынешнее значение на то, что было 3 итерации назад.

Меняем границы, в которых совершаем обход.

3.2) Совершаем проход влево и вверх, запоминая при этом предыдущие значения и меняя нынешнее значение на то, что было 3 итерации назад.

Меняем границы, в которых совершаем обход.

4) Выводим полученную матрицу

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 Ввод:
1
1
Вывод:
1
2 Ввод:
2
0 1
2 3
Вывод:
1 3
0 2
3 Ввод:
3
0 1 2
3 4 5
6 7 8
Вывод:
6 3 4
8 7 0
5 2 1
4 Ввод:
4
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
Вывод:
6 10 9 0
13 12 8 1
14 5 4 2
15 11 7 3

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

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

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 > lab15 > C 15.c > ...
1  // Вариант 12
2  // Циклический сдвиг элементов матрицы по спирали по (против) часовой стрелки на 3 элемента
3  #include <stdio.h>
4
5  int main() {
6      // Считываем количество строк (столбцов)
7      int n;
8      scanf("%d", &n);
9
10     // Заполняем массив размером n * n
11     int t = 0;
12     int mass[n * n];
13     while (t < n * n) {
14         scanf("%d", &mass[t]);
15         t++;
16     }
17
18     // Задаем начальную точку (i строку и j столбец)
19     int i = 1, j = 1;
20     // Задаем границы, в которых совершаем обход
21     int top_limit = 1, bottom_limit = n, right_limit = n, left_limit = 1;
22     // Задаем переменные, нужные для сохранения предыдущих трёх значений
23     int prev0 = 0, prev1 = 0, prev2 = 0, prev3 = 0;
24     // Задаем переменные, нужные для сохранения индексов предыдущих трёх значений
25     int c = 0, c2 = 0, c3 = 0;
26     int c1 = 0;
27     // Задаем начальное состояние (0 - идём вправо и вниз, 1 - идём влево и вверх)
28     int state = 0;
29
```

```

29
30 // Начинаем обход
31 t = 0;
32 for (int k = 0; k < n; k++) {
33     // Проход первого элемента
34     if (i == 1 && j == 1) {
35         c = (i - 1) * n + (j - 1);
36         prev3 = prev2;
37         prev2 = prev1;
38         prev1 = prev0;
39         prev0 = mass[c];
40         mass[c] = prev3;
41         t++;
42         if (t < 4) {
43             c3 = c2;
44             c2 = c1;
45             c1 = c;
46         }
47     }
48     // Совершаем проход вправо и вниз, запоминая при этом предыдущие значения и меняя нынешнее значение
49     if (state == 0) {
50         while (j < right_limit) {
51             j++;
52             c = (i - 1) * n + (j - 1);
53             prev3 = prev2;
54             prev2 = prev1;
55             prev1 = prev0;
56             prev0 = mass[c];
57             mass[c] = prev3;
58             t++;
59             if (t < 4) {
60                 c3 = c2;
61                 c2 = c1;
62                 c1 = c;
63             }
64         }
65         while (i < bottom_limit) {
66             i++;
67             c = (i - 1) * n + (j - 1);
68             prev3 = prev2;
69             prev2 = prev1;
70             prev1 = prev0;
71             prev0 = mass[c];
72             mass[c] = prev3;
73             t++;
74             if (t < 4) {
75                 c3 = c2;
76                 c2 = c1;
77                 c1 = c;
78             }
79         }
80         // Меняем границы, в которых совершаем обход
81         right_limit--;
82         top_limit++;
83         state = 1;
84     }

```

```

84     }
85     // Совершаем проход влево и вверх, запоминая при этом предыдущие значения и меняя нынешнее значение
86     if (state == 1) {
87         while (j > left_limit) {
88             j--;
89             c = (i - 1) * n + (j - 1);
90             prev3 = prev2;
91             prev2 = prev1;
92             prev1 = prev0;
93             prev0 = mass[c];
94             mass[c] = prev3;
95             t++;
96             if (t < 4) {
97                 c3 = c2;
98                 c2 = c1;
99                 c1 = c;
100             }
101         }
102         while (i > top_limit) {
103             i--;
104             c = (i - 1) * n + (j - 1);
105             prev3 = prev2;
106             prev2 = prev1;
107             prev1 = prev0;
108             prev0 = mass[c];
109             mass[c] = prev3;
110             t++;
111             if (t < 4) {
112                 c3 = c2;
113                 c2 = c1;
114                 c1 = c;
115             }
116         }
117         // Меняем границы, в которых совершаем обход
118         left_limit++;
119         bottom_limit--;
120         state = 0;
121     }
122 }
123 // Присваиваем первым трём элементам массива, значения трёх последних, по которым прошли.
124 mass[c3] = prev2;
125 mass[c2] = prev1;
126 mass[c1] = prev0;
127 // Выводим полученную матрицу
128 t = 0;
129 for (int i = 0; i < n; i++) {
130     for (int j = 0; j < n; j++) {
131         printf("%d ", mass[t]);
132         t++;
133     }
134     printf("\n");
135 }
136
137 return 0;
138 }

```

Labs > lab15 > test1

```

1  1
2  1

```

Labs > lab15 > test2

```

1  2
2  0 1
3  2 3

```

Labs > lab15 > test3

```

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

```

<pre>Labs > lab15 > ≡ test4 1 4 2 0 1 2 3 3 4 5 6 7 4 8 9 10 11 5 12 13 14 15</pre>	<pre>Labs > lab15 > ≡ test5 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</pre>	<pre>Labs > lab15 > ≡ test6 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</pre>
<pre>Labs > lab15 > ≡ test7 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</pre>		

```

• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ gcc 15.c -o 15
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test1
1
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test2
1 3
0 2
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test3
6 3 4
8 7 0
5 2 1
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test4
6 10 9 0
13 12 8 1
14 5 4 2
15 11 7 3
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test5
16 11 12 0 1
20 15 10 5 2
21 18 17 6 3
22 13 8 7 4
23 24 19 14 9
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test6
15 21 20 0 1 2
24 18 12 6 7 3
30 26 25 19 8 4
31 27 14 13 9 5
32 28 22 16 10 11
33 34 35 29 23 17
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$ ./15 < test7
30 23 24 0 1 2 3
28 21 14 7 8 9 4
35 36 29 22 15 10 5
42 37 32 31 16 11 6
43 38 25 18 17 12 13
44 39 40 33 26 19 20
45 46 47 48 41 34 27
• jonaj@JonAJ:~/Desktop/StudyDir/Labs/lab15$

```

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 лабораторной работы я научился обрабатывать квадратные матрицы, написал программу, которая производит циклический сдвиг элементов матрицы по спирали по часовой стрелке на 3 элемента

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента *AO*