

# Отчет по лабораторной работе № 14 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Шамбилов Руслан Талгатович, № 23 по списку

Контакты e-mail [ruslanshmbiloff@yandex.ru](mailto:ruslanshmbiloff@yandex.ru), telegram @willr4in.

Работа выполнена: «6» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » декабря 2022 г., итоговая оценка

Подпись преподавателя

**1.Тема:** Язык программирования C.

**2. Цель работы:** Научиться работать с массивами

**3. Задание:** В форме заданного алгоритма вывести цепочку чисел

**4. Оборудование (студента):**

Процессор AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz с ОП 16000Мб, НМД 512 Гб.  
Монитор 1920x1080

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

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*  
интерпретатор команд: *bash* версия *4.4.19*.  
Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*

**6. Идея, метод, алгоритм**

Идея состоит в том, чтобы взять 5 циклов (один будет определять кол-во вхождений, 4 остальных - движение вниз, вправо, вверх, влево). Попытка была хорошей, для матриц кратных 2 работает отлично. Для матриц не кратных 2 - число, определяющее кол-во цифр перед собой, цикл, который будет выводить определенное кол-во цифр и так до конца.

**7. Сценарий выполнения работы**

Входные данные	Выходные данные	Описание тестируемого случая
1 3 2 4	1 2 3 4	

7 6 5 8 1 3 9 2 4	1 2 3 4 5 6 7 8 9	

## 8. Распечатка протокола

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

```
/* 2
```

```
1 4
```

```
2 3 */
```

```
/* 3
```

```
7 6 5
```

```
8 1 4
```

```
9 2 3 */
```

```
/* 4
```

```
7 6 5 16
```

```
8 1 4 15
```

```
9 2 3 14
```

```
10 11 12 13 */
```

```
/* 5
```

```
21 20 19 18 17
```

```
22 7 6 5 16
```

```
23 8 1 4 15
```

```
24 9 2 3 14
```

```
25 10 11 12 13 */
```

```
/* 6
```

```
21 20 19 18 17 36
```

```
22 7 6 5 16 35
```

```
23 8 1 4 15 34
```

```
24 9 2 3 14 33
```

```
25 10 11 12 13 32
```

```
26 27 28 29 30 31 */
```

```
/* 7
```

```
43 42 41 40 39 38 37
```

```
44 21 20 19 18 17 36
```

```
45 22 7 6 5 16 35
```

```
46 23 8 1 4 15 34
```

```
47 24 9 2 3 14 33
```

```
48 25 10 11 12 13 32
```

49 26 27 28 29 30 31 \*/

```
enum {maxn = 100};
```

```
int main() {
```

```
    int m[maxn][maxn];
```

```
    int n;
```

```
    scanf_s("%d", &n);
```

```
    for (int i = 0; i != n; ++i) {
```

```
        for (int j = 0; j != n; ++j) {
```

```
            scanf_s("%d", &m[i][j]);
```

```
        }
```

```
    }
```

```
    int help[3] = {1, 2, 3};
```

```
    int q[3] = {0, 2, 4};
```

```
    int x[3] = {-1, 1, 3};
```

```
    int y[2] = {3, 5};
```

```
    int v = 1;
```

```
    if (n % 2 == 1) {
```

```
        int l = 2;
```

```
        int d = 0;
```

```
        int i, j;
```

```
        if (n == 7) {
```

```
            i = 2;
```

```
            j = 3;
```

```
        }
```

```
        if (n == 5) {
```

```
            i = 1;
```

```
            j = 2;
```

```
        }
```

```
        if (n == 3) {
```

```
            i = 0;
```

```
            j = 1;
```

```
        }
```

```
        if (n == 1) {
```

```
            i = 0;
```

```
            j = 0;
```

```
            printf("%d", m[i][j]);
```

```
        }
```

```
    int dir[4] = {1, 0, -1, 0};
```

```
    int ftp[12] = {-1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0};
```

```
    while (l != n + 1) {
```

```
        for (int k = 0; k != l; ++k) {
```

```

        j += dir[(d + 3) % 4];
        i += dir[d % 4];
        printf("%d", m[i][j]);
        printf("%c", ' ');
    }
    if ((i == 6) && (j == 0) || (i == 4) && (j == 0) || (i == 3) && (j == 0)) {
        break;
    }
    l += ftp[d % 12];
    ++d;
}
}
else {
    int p = n / 2;
    for (int i = p - 1; i >= p - help[p - 1]; i--) {
        for (int j = i; j <= n - 1 - i; j++)
            printf("%d ", m[j][i]);
        for (int j = i; j < n - 1 - i; j++)
            printf("%d ", m[n - i - 1][j + 1]);
        for (int j = q[p-1] - i; j > x[p - v] - i; j--)
            printf("%d ", m[j][n - i - 1]);
        if (i == 0) {
            break;
        }
        for (int j = y[p-2] - i; j > i - 1; j--)
            printf("%d ", m[i - 1][j]);
        v += 1;
    }
}
printf("\n");
return 0;
}

```

## 9. Дневник отладки

№	Лаб.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	28 ноября	18:23	Получилось написать код для матрицы 2x2, но для других не работает	Начинаю переписывать весь код (так у меня повторилось раз 6)	Логично было предположить, если я делаю код для определенной матрицы - работать он будет только для нее
2	дом	2 декабря	21:20	Написал код для всех матриц, кратных 2, но не получается ни в какую для некратных	Переписываю условие для некратных 2 матриц	Не понимаю, в чем ошибка, при выводе выходят рандомные числа, но и последовательность чисел верная, видимо я залезаю в память

						компьютера, что ему не нравится
3	дом	6 декабря	17:56	Все-таки идея с 5 форами не удалась	Пишу новый код для некратных 2	Сделал плохой алгоритм для некратных матрица, но он хотя бы не лезет в память и выдает все верно...

#### 10. Замечания автора

Нет.

#### 11. Выводы

Лабораторная работа номер 14 оказалась для меня самой сложной на данный момент, спустя 2 недели КАЖДОДНЕВНЫХ правок я все-таки смог ее сделать, но весьма плохо. Благодаря проделанным действиям, научился строить массивы и выводить их по определенному алгоритму. Так же хотелось сказать, что лучше познал ошибки в C, ибо на пути их встретилась большая куча. В общем и целом или в целом и общем, работа понравилась, но утомила, ведь каждый день видеть перед собой массив, который ты не можешь вывести - боль, которую надо было фиксировать.

Подпись студента

---