**Отчет по лабораторной работе №14** по курсу \_\_Фундаментальная информатика\_\_\_\_\_\_\_\_\_\_\_\_

Студент группы М80-101Б-21 Постнов Александр Вячеславович, № по списку 17

Контакты www, e-mail:

Работа выполнена: «» 202 1г.

Преподаватель: каф. 806 \_\_\_\_\_Титов В.К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Отчет сдан « » \_\_\_\_\_\_\_\_\_2021\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

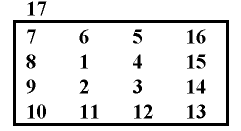
1. **Тема:** Вложенные циклы с параметрами. Обход и линеаризация матриц.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Цель работы:** составить и отладить программу на СИ на заданную тему. \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Задание** (*вариант: 17*)**:** Выполнить обход матрицы в заданном порядке:



1. **Оборудование(лабораторное)**:

ЭВМ -, процессор -, имя узла сети - с ОП - ГБ,

НМД - ГБ, терминал- адрес -, принтер -

Другие устройства -

*Оборудование ПЭВМ студента, если использовалось:*

Процессор AMD Ryzen 5 4500U, с ОП 8 ГБ

Другие устройства -

1. **Программное обеспечение:**

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 GNU/Linux, наименование Pop!\_OS версия 21.04

интерпретатор команд GNOME Terminal версия 3.38.2

Система программирования \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_версия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Редактор текстов emacs версия 3.27.20

Утилиты операционной системы cat

Прикладные системы и программы -

Местонахождение и имена файлов программ и данных -

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

Словесный алгоритм:

1) начальная точка обхода — это центральная точка матрицы, будем искать ее таким образом:

i = (n + 1) / 2 - 1 (округление вверх, минус 1, так как нумерация элементов идет с нуля)

j = (n + 1) / 2 - 1

2) обход матрицы будет производиться с использованием 4 циклов внутри главного цикла:

1. вертикальный нижний цикл c увеличением i

2. вертикальный верхний цикл c уменьшением i

3. горизонтальный правый цикл с увеличением j

4. горизонтальный левый цикл с уменьшением j

3) кол-во повторений цикла будет зависеть от зависеть от параметра counter, который будет увеличиваться в главном цикле. Использование этих циклов до тех пор, пока counter < n (размера матрицы)

4) за итерацию главного цикла должны использоваться или 1 и 3 цикл, или 2 и 4 цикл. Выбор будет зависеть от четности параметра k, который так же будет увеличиваться в главном цикле

5) в конце останется использовать какой-либо вертикальный цикл, так как в конце останется вертикальная линия, которая не была отображена.

6) выбор конечного вертикального цикла будет от j;

1. если j == 0, то нижний до n - 1

2. если j == n — 1, то верхний до 0

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

#include <stdio.h>

int main(void){

int a[10][10];

int n;

printf("Введите размер матрицы n \* n: ");

scanf("%d", &n);

for (int i = 0; i < n; i++){ //ввод матрицы

for (int j = 0; j < n; j++){

a[i][j] = (i + 1) \* 10 + j + 1;

}

}

printf("Заданная матрица: \n");

for (int i = 0; i < n; i++){ //вывод матрицы

for (int j = 0; j < n; j++){

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

}

printf("\n");

}

int i = (n + 1) / 2 - 1; //центральная точка матрицы(начальная точка обхода)

int j = (n + 1) / 2 - 1;

if (n > 0){ //на случай если в матрице 0 элементов

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

}

for (int counter = 1, k = 0; counter < n; counter++, k++){

if (k % 2 == 0){

for (int i0 = 0; i0 < counter; i0++){ //вертикальный нижний цикл

printf("%d ", a[++i][j]);

}

for (int j0 = 0; j0 < counter; j0++){ //горизонтальный правый цикл

printf("%d ", a[i][++j]);

}

}

if (k % 2 == 1){

for (int i0 = counter - 1; i0 >= 0; i0--){ //вертикальный верхний цикл

printf("%d ", a[--i][j]);

}

for (int j0 = counter - 1; j0 >= 0; j0--){ //горизонтальный левый цикл

printf("%d ", a[i][--j]);

}

}

}

if (n != 1 && i == 0){ //заполнение последней вертикальной линии

for (int h = 1; h < n; h++){ //вертикальный нижний цикл

printf("%d ", a[h][j]);

}

}

else if (n != 1 && i == n - 1){

for (int h = n - 2; h >= 0; h--){ //вертикальный верхний цикл

printf("%d ", a[h][j]);

}

}

printf("\n");

return 0;

}

Тесты:

1 → 11

2 → 11 21 22 12

3 → 22 32 33 23 13 12 11 21 31

5 → 33 43 44 34 24 23 22 32 42 52 53 54 55 45 35 25 15 14 13 12 11 21 31 41 51

8 → 44 54 55 45 35 34 33 43 53 63 64 65 66 56 46 36 26 25 24 23 22 32 42 52 62 72 73 74 75 76 77 67 57 47 37 27 17 16 15 14 13 12 11 21 31 41 51 61 71 81 82 83 84 85 86 87 88 78 68 58 48 38 28 18

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

[alex@alex 14c(?)]$ cat head.txt

-----------------------------------------------------

| Лабораторная работа №14 |

| Обход и линеаризация матриц. |

| Выполнил: студент группы М8О-101Б-21 |

| Постнов Александр Вячеславович |

-----------------------------------------------------

[alex@alex 14c(?)]$ cat main.c

#include <stdio.h>

int main(void){

int a[10][10];

int n;

printf("Введите размер матрицы n \* n: ");

scanf("%d", &n);

for (int i = 0; i < n; i++){ //ввод матрицы

for (int j = 0; j < n; j++){

a[i][j] = (i + 1) \* 10 + j + 1;

}

}

printf("Заданная матрица: \n");

for (int i = 0; i < n; i++){ //вывод матрицы

for (int j = 0; j < n; j++){

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

}

printf("\n");

}

int i = (n + 1) / 2 - 1; //центральная точка матрицы(начальная точка обхода)

int j = (n + 1) / 2 - 1;

if (n > 0){ //на случай если в матрице 0 элементов

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

}

for (int counter = 1, k = 0; counter < n; counter++, k++){

if (k % 2 == 0){

for (int i0 = 0; i0 < counter; i0++){ //вертикальный нижний цикл

printf("%d ", a[++i][j]);

}

for (int j0 = 0; j0 < counter; j0++){ //горизонтальный правый цикл

printf("%d ", a[i][++j]);

}

}

if (k % 2 == 1){

for (int i0 = counter - 1; i0 >= 0; i0--){ //вертикальный верхний цикл

printf("%d ", a[--i][j]);

}

for (int j0 = counter - 1; j0 >= 0; j0--){ //горизонтальный левый цикл

printf("%d ", a[i][--j]);

}

}

}

if (n != 1 && i == 0){ //заполнение последней вертикальной линии

for (int h = 1; h < n; h++){ //вертикальный нижний цикл

printf("%d ", a[h][j]);

}

}

else if (n != 1 && i == n - 1){

for (int h = n - 2; h >= 0; h--){ //вертикальный верхний цикл

printf("%d ", a[h][j]);

}

}

printf("\n");

return 0;

}

[alex@alex 14c(?)]$ g++ main.c

[alex@alex 14c(?)]$ ./a.out

Введите размер матрицы n \* n: 1

Заданная матрица:

11

11

[alex@alex 14c(?)]$ ./a.out

Введите размер матрицы n \* n: 2

Заданная матрица:

11 12

21 22

11 21 22 12

[alex@alex 14c(?)]$ ./a.out

Введите размер матрицы n \* n: 3

Заданная матрица:

11 12 13

21 22 23

31 32 33

22 32 33 23 13 12 11 21 31

[alex@alex 14c(?)]$ ./a.out

Введите размер матрицы n \* n: 5

Заданная матрица:

11 12 13 14 15

21 22 23 24 25

31 32 33 34 35

41 42 43 44 45

51 52 53 54 55

33 43 44 34 24 23 22 32 42 52 53 54 55 45 35 25 15 14 13 12 11 21 31 41 51

[alex@alex 14c(?)]$ ./a.out

Введите размер матрицы n \* n: 8

Заданная матрица:

11 12 13 14 15 16 17 18

21 22 23 24 25 26 27 28

31 32 33 34 35 36 37 38

41 42 43 44 45 46 47 48

51 52 53 54 55 56 57 58

61 62 63 64 65 66 67 68

71 72 73 74 75 76 77 78

81 82 83 84 85 86 87 88

44 54 55 45 35 34 33 43 53 63 64 65 66 56 46 36 26 25 24 23 22 32 42 52 62 72 73 74 75 76 77 67 57 47 37 27 17 16 15 14 13 12 11 21 31 41 51 61 71 81 82 83 84 85 86 87 88 78 68 58 48 38 28 18

**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 |
| 1 | дом | 21.11 | 2:25 | Выдала несуществующий элемент матрицы при n = 0 | При n = 0, не выводил центральный элемент, так как его нет. |  |
| 2 | дом | 21.11 | 2:30 | Программа не скомпилировалась из-за синтаксической ошибки | Добавил “;” в строках 17, 21. |  |

**10. Замечания автора :**

11.**Выводы**

\_\_В ходе лабораторной работе я составил и отладил программу на СИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_Постнов\_\_\_\_\_\_\_\_\_\_\_\_\_\_