

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

Студент группы М80-109Б-22 Степанов Алексей Николаевич, № по списку 18

Контакты aleksey.stepanov2004@mail.ru, telegram @Alex1stepa

Работа выполнена: «3» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

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

2. **Цель работы:** Составить программу ввода квадратичной матрицы и печати в строку её элементов в заданном порядке (обход) в соответствии с вариантом задачи.

3. **Задание (вариант 28):**

28

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

4. Оборудование (студента):

Процессор *Intel Core i5-8265U @ 8x 3.9GH* с ОП 7851 Мб, НМД 1024 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*

Утилиты операционной системы --

Прикладные системы и программы: *VTM(QT)*

Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

1. Затребуем размерность вводимой квадратной матрицы, после чего мы, используя 2 вложенных цикла считаем ее в двумерный массив.

2. Чтобы правильно по нему пройти заведем массивы смещений за 1 единицу дискретного времени по “х” и “у” нашей матрицы(если бы мы представляли ее нарисованной на бумаге), а также переменные, ставящие движению границы. Если мы достигаем 1 из них(текущие переменные совпадают с границами) – мы сменяем патерн движения на следующий, пока мы не сравняемся по переменным х, что будет означать попадание в центр и достижение последнего элемента, который надо будет еще вывести, ибо цикл не сможет этого сделать.

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

Тесты	Ожидание	Реальный результат работы
1 2 3 4	3 4 2 1	3 4 2 1
1	1	1
1 2 3 4 5 6 7 8 9	7 8 9 6 3 2 1 4 5	7 8 9 6 3 2 1 4 5
10 9 8 7 11 16 15 6 12 13 14 5 1 2 3 4	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
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 25	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

8.

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

Код программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <assert.h>
```

```
int64_t mx[4]={ 1,0,-1,0};
int64_t my[4]={ 0,-1,0,1};
```

```
int main()
```

```
{  int64_t matrix[100][100];
    int64_t n=0;
    printf("Enter the dimension of the matrix\n");
    scanf("%lld",&n);
    if(n==0){
        return 0;
    }
    printf("Enter the elements of the matrix\n");
    for(int64_t i=0;i<n;i++){
        for(int64_t j=0;j<n;j++){
            scanf("%lld",&matrix[j][i]);
        }
    }
    int64_t left_x_border=0;
    int64_t right_x_border=n-1;
    int64_t lowwer_y_border=n-1;
    int64_t upper_y_border=0;
    int64_t x=0;
    int64_t y=n-1;
    int64_t p=0; //pointer for mx and my
    int64_t flag=0;

    while(left_x_border!=right_x_border){
        printf("%lld ",matrix[x][y]);
        x+=mx[p];
        y+=my[p];
        if(x==right_x_border&&y==lowwer_y_border){
            p=1;
            if(lowwer_y_border!=upper_y_border){
                lowwer_y_border--;
                printf("%lld ",matrix[x][y]);
                y--;
            }
        }
        else{
            printf("%lld ",matrix[x][y]);
            p=2;
            continue;
        }
    }
    if(x==right_x_border&&y==upper_y_border){
        p=2;
        if(right_x_border!=left_x_border){
            right_x_border--;
            printf("%lld ",matrix[x][y]);
            x--;
        }
    }
    if(x==left_x_border&&y==upper_y_border){
        p=3;
        if(lowwer_y_border!=upper_y_border){
            upper_y_border++;
            printf("%lld ",matrix[x][y]);
        }
    }
}
```

```

        y++;
    }
}
if (x==left_x_border&& y==lower_y_border){
    p=0;
    if (left_x_border!=right_x_border){
        left_x_border++;

        printf("%lld ",matrix[x][y]);
        x++;
    }
}

}
printf("%lld ",matrix[x][y]);
return 0;
}

```

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

Замечания отсутствуют, работа конструктивная.

11. Выводы

От лабораторной работы получил исключительно положительные эмоции и впечатления. По моему мнению, знания, приобретенные мною на данной лабораторной работе, помогли мне лучше осознать принципы работы операционных систем, принципы представления матриц и их линеаризации в ЯП Си, принципы работы двумерных массивов в Си, отладчика Си, более пристально изучить язык программирования Си, научиться отлаживать свой код и находить ошибки разного типа, что несомненно поможет мне при решении практических задач.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента _____