Южно-Уральский государственный колледж, корпус Информационных Технологий и Экономики

**Отчет**

По дисциплине МДК.01.04. Системное программирование

Лабораторная работа №14 «Обработка массивов указателей»

Вариант №10

Выполнил студент Яковлев Е.К.

Группа ИСп320Д

2023 г.

**Ход работы:**

1. **Задание:** в двумерной целочисленной матрице найти количество: 1) четных чисел; 2) чисел, делящихся на 4. Все четные элементы заменить на 2.
   1. Код программы:

#include <stdio.h>

#include <locale>

// Функция для подсчета количества четных чисел в матрице

int countEven(int matrix[3][3]) {

int count = 0;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (matrix[i][j] % 2 == 0) {

count++;

}

}

}

return count;

}

// Функция для подсчета количества чисел, делящихся на 4, в матрице

int countDivisibleByFour(int matrix[3][3]) {

int count = 0;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (matrix[i][j] % 4 == 0) {

count++;

}

}

}

return count;

}

// Функция для замены всех четных элементов на число 2

void replaceEvenWithTwo(int matrix[3][3]) {

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

if (matrix[i][j] % 2 == 0) {

matrix[i][j] = 2;

}

}

}

}

int main() {

setlocale(LC\_ALL, "");

int matrix[3][3] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

printf("Матрица:\n");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

printf("\n");

}

printf("\nКоличество четных чисел: %d", countEven(matrix));

printf("\nКоличество чисел, делящихся на 4: %d", countDivisibleByFour(matrix));

replaceEvenWithTwo(matrix);

printf("\n\nМатрица после замены четных элементов на 2:\n");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

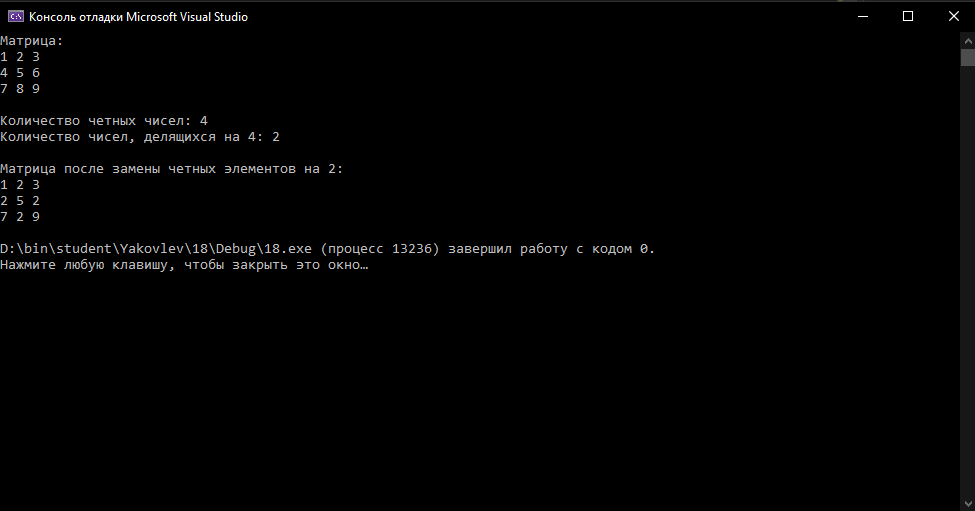
}

printf("\n");

}

return 0;

}

* 1. Снимок экрана: 

**Контрольные вопросы**

1. **Понятие указателя на функцию.**

Указатель на функцию - это переменная, которая содержит адрес памяти функции. Он позволяет нам обращаться к функции через этот адрес, вызывать ее и передавать ее как аргумент другой функции.

1. **Обращение к функции через указатель**

Обращение к функции через указатель осуществляется с помощью оператора разыменования (\*). Мы используем указатель на функцию как обычную переменную, но перед вызовом функции разыменовываем указатель, чтобы получить доступ к функции по адресу, который он хранит

**Вывод:** таким образом, научился работать с массивами указателей, передавать массивы в качестве аргументов функций.