Пензенский государственный университет

Кафедра "Вычислительная техника"

**ОТЧЕТ**

по лабораторной работе №8

по курсу «Программирование»

на тему "Файлы"

Выполнили:

Студенты группы 22ВВ2

Беляев Д. И.

Захаров А. С.

Ипполитов И. Д.

Приняли:

Юрова О. В.

Патунин Д. В.

Пенза 2022

**Название**Файлы

**Цель работы**Изучение способов описания файлов и основных принципов работы с файлами в различных режимах: создания, модификации, добавления и чтения.

**Лабораторное задание**

1.  а). Задан массив целых чисел А(20). Все элементы, кратные числу К, заменить на нуль.

б). Дана матрица N(6,5). Найти строку с максимальной суммой элементов.

2. В задание №6 лабораторной работы добавить работу с файлами:

а) в функции ввода исходные данные записывать в файл;

б) в функциях обработки и вывода данные читать из файла, результат сохранять в файл;

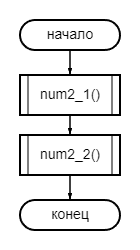
в) для второго задания лабораторной работы №6 данные дописывать в файл с исходными данными для первого задания;

г) результаты выполнения и первого, и второго задания тоже должны быть сохранены в одном файле.

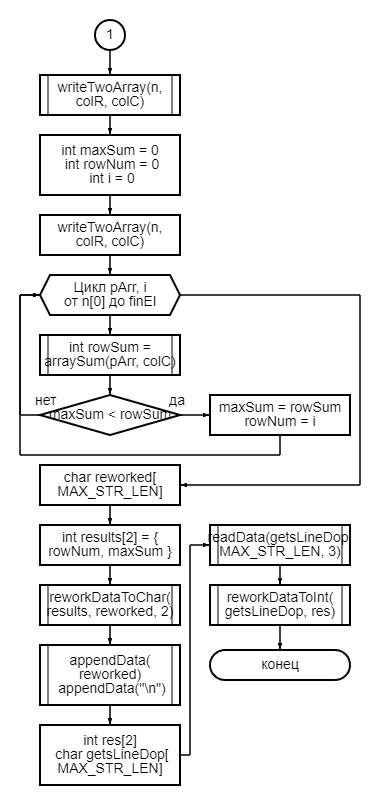
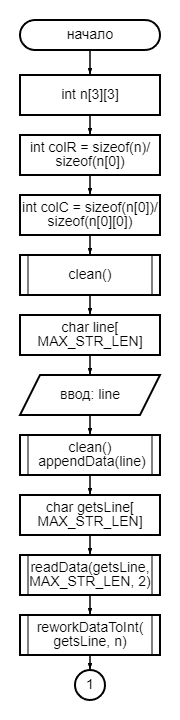
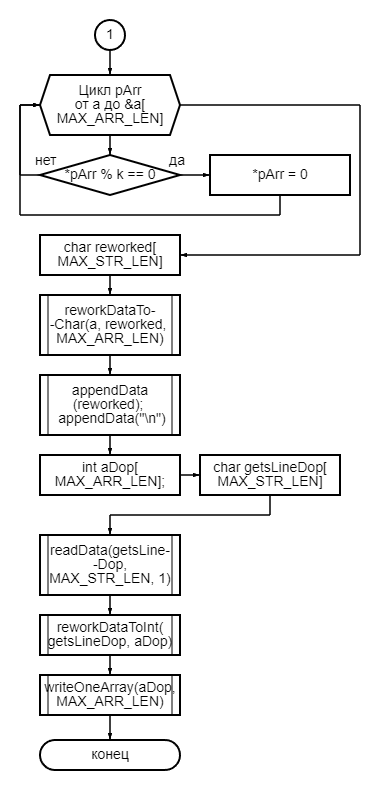
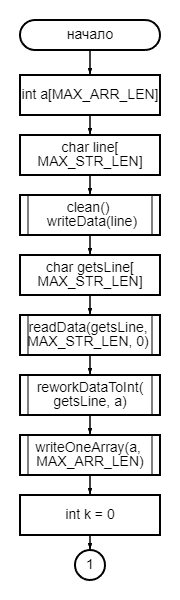
**Ход работы:**

1. **Схема программы**

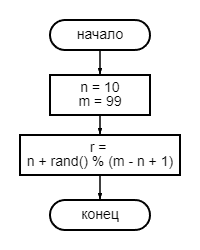
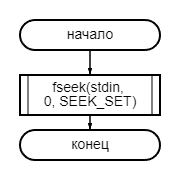
* **main.c**

****

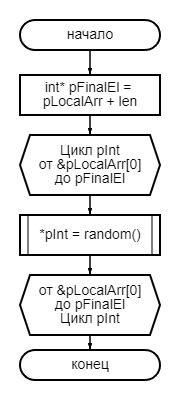
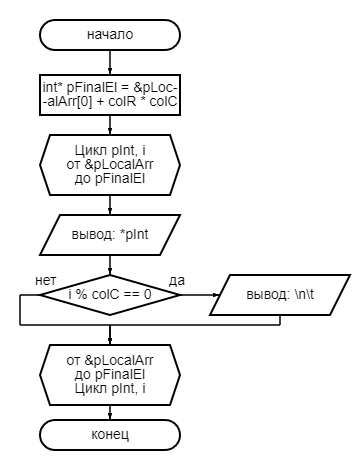
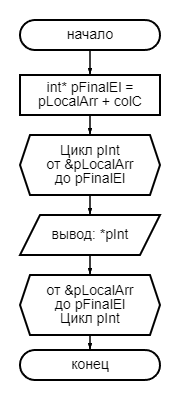
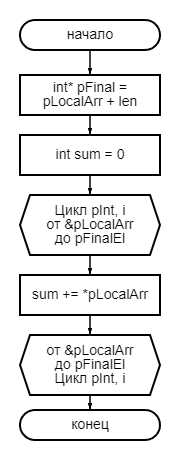
* **var12.c**



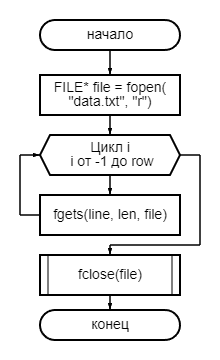
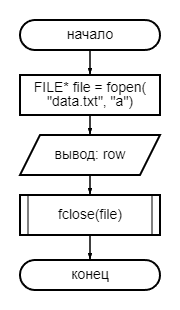
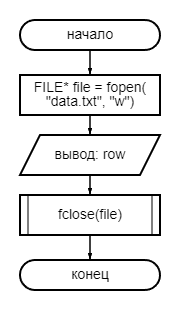
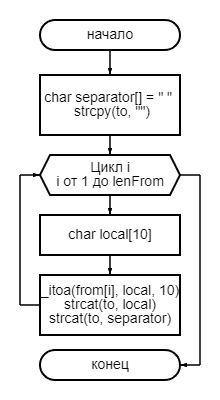
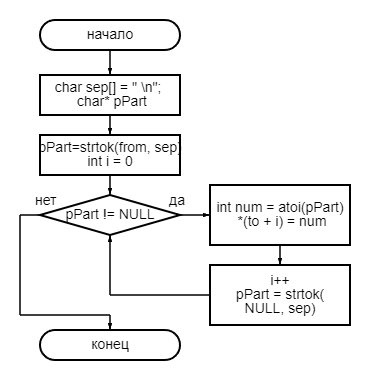
* **func.h**



* **dataWork.h**



* **arrayWork.h**



1. **Код программы**

* **main.c**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include "arrayWork.h"

#include "dataWork.h"

int main()

{

printf("==Var2==\n\n");

num2\_1();

printf("\n-------------\n");

num2\_2();

return 0;

}

* **var12.c**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#define MAX\_ARR\_LEN 20

#define MAX\_STR\_LEN 100

int num2\_1()

{

//заготовленный массив 20 34 35 65 75 87 64 32 15 89 65 78 34 12 11 99 37 69 42 55

int a[MAX\_ARR\_LEN];

printf("Insert array (one line): ");

// Заполняем data

char line[MAX\_STR\_LEN];

fgets(line, MAX\_STR\_LEN, stdin);

clean();

writeData(line);

// Преобразуем из data

char getsLine[MAX\_STR\_LEN];

readData(getsLine, MAX\_STR\_LEN, 0);

reworkDataToInt(getsLine, a);

// Выводим массив

writeOneArray(a, MAX\_ARR\_LEN);

// Основной код

printf("\nInsert dop number: ");

int k = 0;

scanf("%d", &k);

printf("\n");

for (int\* pArr = a; pArr < &a[MAX\_ARR\_LEN]; pArr++)

{

if (\*pArr % k == 0)

\*pArr = 0;

}

// Преобразуем для data

char reworked[MAX\_STR\_LEN];

reworkDataToChar(a, reworked, MAX\_ARR\_LEN);

// Сохраняем в data

appendData(reworked);

appendData("\n");

// Преобразуем из data

int aDop[MAX\_ARR\_LEN];

char getsLineDop[MAX\_STR\_LEN];

readData(getsLineDop, MAX\_STR\_LEN, 1);

reworkDataToInt(getsLineDop, aDop);

// Выводим массив

writeOneArray(aDop, MAX\_ARR\_LEN);

return 0;

}

int num2\_2()

{

// Заготовленный массив 12 14 56 43 98 76 45 23 10

int n[3][3];

int colR = sizeof(n) / sizeof(n[0]);

int colC = sizeof(n[0]) / sizeof(n[0][0]);

printf("Insert array (one line): ");

// Дополняем в data

clean();

char line[MAX\_STR\_LEN];

fgets(line, MAX\_STR\_LEN, stdin);

clean();

appendData(line);

// Преобразуем из data

char getsLine[MAX\_STR\_LEN];

readData(getsLine, MAX\_STR\_LEN, 2);

reworkDataToInt(getsLine, n);

// Выводим массив

writeTwoArray(n, colR, colC);

// Основной код

int maxSum = 0;

int rowNum = 0;

int i = 0;

int\* finEl = n[0] + colC \* colR - 1;

for (int\* pArr = n[0]; pArr <= finEl; pArr += colC, i++)

{

int rowSum = arraySum(pArr, colC);

if (maxSum < rowSum)

{

maxSum = rowSum;

rowNum = i;

}

}

// Преобразуем для data

char reworked[MAX\_STR\_LEN];

int results[2] = { rowNum, maxSum };

reworkDataToChar(results, reworked, 2);

// Добавляем в data

appendData(reworked);

appendData("\n");

// Преобразуем из data

int res[2];

char getsLineDop[MAX\_STR\_LEN];

readData(getsLineDop, MAX\_STR\_LEN, 3);

reworkDataToInt(getsLineDop, res);

printf("Max sum: %d - %d\n", res[0] + 1, res[1]);

}

* **func.h**

#pragma once

#include <stdlib.h>

#include <time.h>

#include <stdio.h>

// generate random bumber

int random()

{

// числа генерируются от 10 до 99

int n = 10;

int m = 99;

// генерируется рандомное число

int r = n + rand() % (m - n + 1);

return r;

}

int clean()

{

fseek(stdin, 0, SEEK\_SET);

return 0;

}

* **arrayWork.h**

#pragma once

#include "func.h"

#include <stdlib.h>

#include <stdio.h>

int fillArray(int\* pLocalArr, int len)

{

int\* pFinalEl = pLocalArr + len;

for (int\* pInt = &pLocalArr[0]; pInt < pFinalEl; pInt++)

{

\*pInt = random();

}

return 0;

}

int writeOneArray(int\* pLocalArr, int colC)

{

int\* pFinalEl = pLocalArr + colC;

printf("Array:\n\t");

for (int\* pInt = pLocalArr; pInt < pFinalEl; pInt++)

{

printf("%d ", \*pInt);

}

printf("\n");

return 0;

}

int writeTwoArray(int\* pLocalArr, int colR, int colC)

{

int\* pFinalEl = &pLocalArr[0] + colR \* colC;

printf("Array:\n\t");

for (int\* pInt = &pLocalArr[0], i = 1; pInt < pFinalEl; pInt++, i++)

{

printf("%d ", \*pInt);

if (i % colC == 0)

printf("\n\t");

}

printf("\n");

return 0;

}

// find sum of array

int arraySum(int\* pLocalArr, int len)

{

int\* pFinal = pLocalArr + len - 1;

int sum = 0;

for (pLocalArr; pLocalArr <= pFinal; pLocalArr++)

{

sum += \*pLocalArr;

}

return sum;

}

* dataWork.h

#pragma once

#include <stdlib.h>

#include <time.h>

#include <stdio.h>

#include <string.h>

int writeData(char row[])

{

FILE\* file = fopen("data.txt", "w");

fprintf(file, row);

fclose(file);

return 0;

}

int appendData(char row[])

{

FILE\* file = fopen("data.txt", "a");

fprintf(file, row);

fclose(file);

return 0;

}

int readData(char line[], int len, int row)

{

FILE\* file = fopen("data.txt", "r");

for (int i = -1; i < row; i++)

{

fgets(line, len, file);

}

fclose(file);

return 0;

}

int reworkDataToInt(char from[], int\* to)

{

char sep[] = " \n";

char\* pPart;

pPart = strtok(from, sep);

int i = 0;

while (pPart != NULL)

{

int num = atoi(pPart);

\*(to + i) = num;

i++;

pPart = strtok(NULL, sep);

}

return 0;

}

int reworkDataToChar(int\* from, char to[], int lenFrom)

{

char separator[] = " ";

strcpy(to, "");

for (int i = 0; i < lenFrom; i++)

{

char local[10];

\_itoa(from[i], local, 10);

strcat(to, local);

strcat(to, separator);

}

return 0;

}

1. **Ручной счет**
   1. 14 68 96 55 60 79 80 60 76 61 13 48 19 33 64 74 46 60 26 96  
      0 0 0 55 0 79 0 0 0 61 13 0 19 33 0 0 0 0 0 0
   2. 81 38 34 49 66 - 268

83 57 38 18 22 - 218

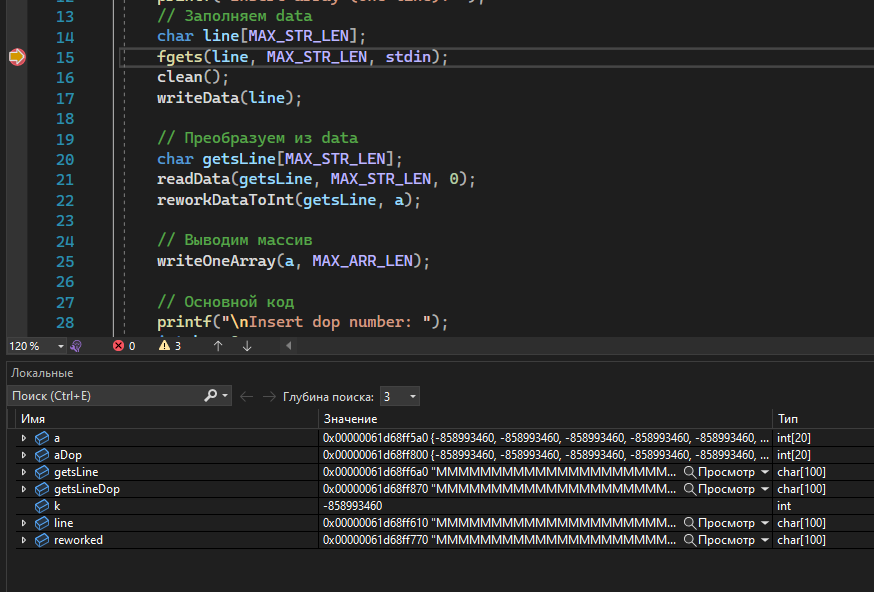
**79 81 73 75 59 - 367**

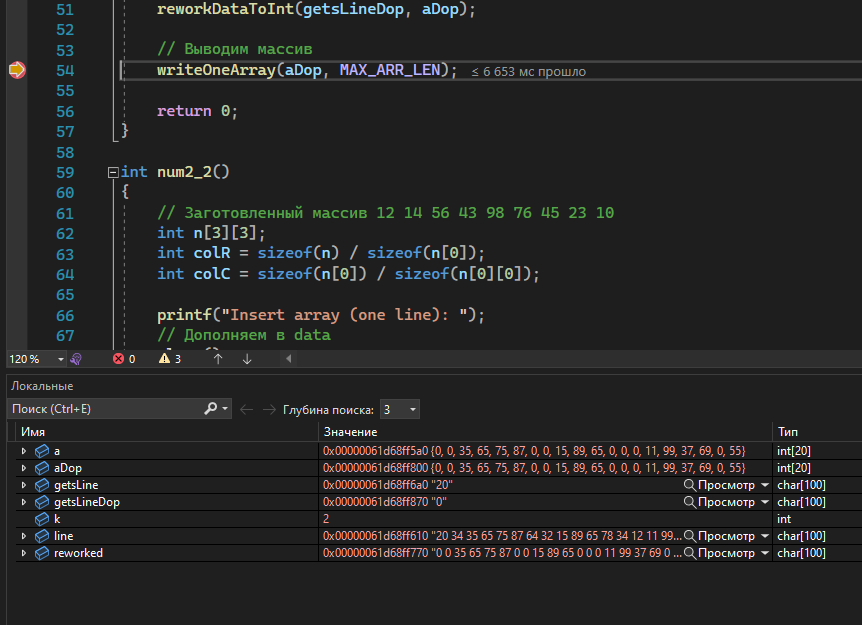
58 34 40 37 66 - 235

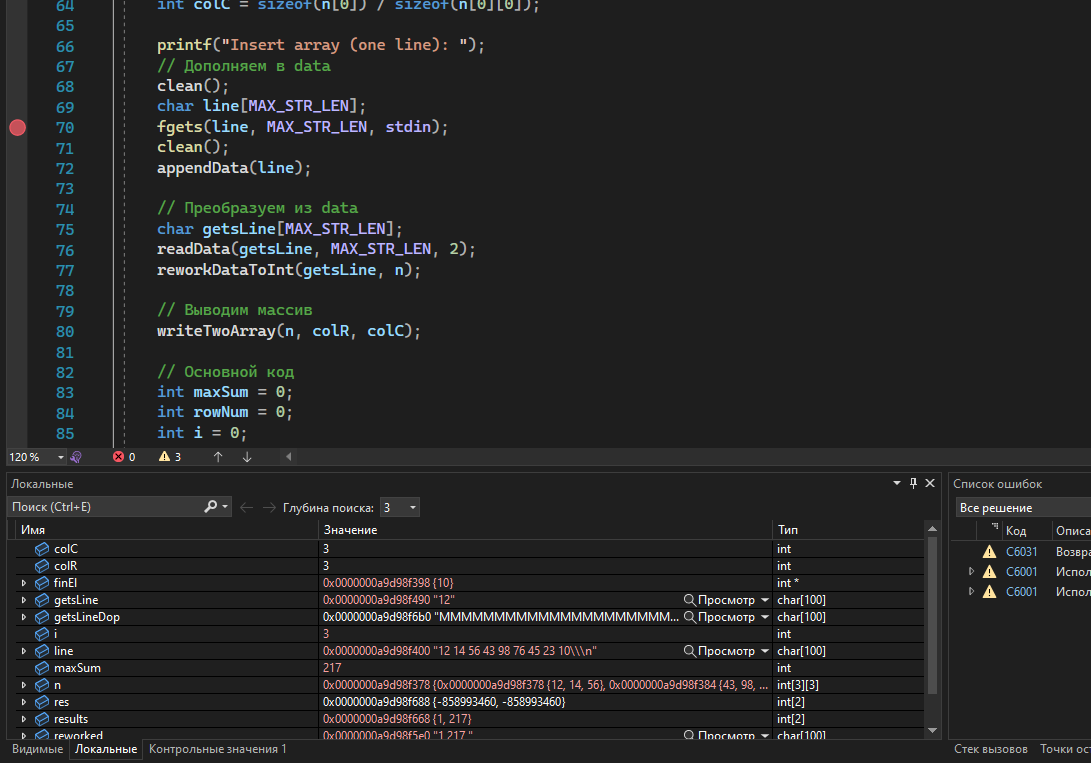
93 76 81 45 24 - 319

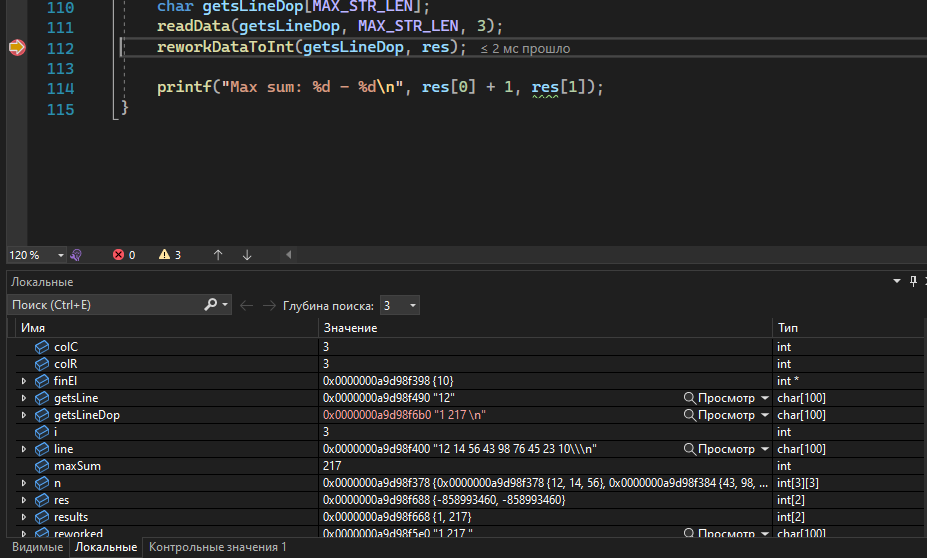
92 70 79 57 13 - 311

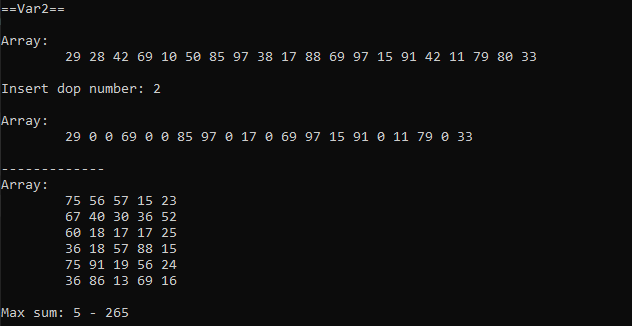
1. **Пояснительный текст к программе**
   1. Массив A длиной 20 заполняется с клавиатуры. Данные сохраняются, после чего считываются. Задается число k. После перебирается каждый символ массива и проверяется: если элемент массива кратен числу k, то он заменяется на ноль 0, если нет, то остается без изменений. Массив Сохраняется в файл
   2. Матрица (двумерный массив) с размерами 6x5 вводится с клавиатуры. После чего сохраняется в файл. После этого программа считывает данные и построчно проверяет сумму всех элементов каждой строчки, находит наибольшую и выводит ее. Ответ сохраняется в файл.
2. **Трассировка**









1. **Результат работы программы**

**Вывод:**Изучили способы описания файлов и основные принципы работы с файлами в различных режимах: создания, модификации, добавления и чтения. Благодаря этому мы сделали программу в который используются полученные нами навыки по созданию файлов.