Пензенский государственный университет

Кафедра "Вычислительная техника"

**ОТЧЕТ**

по лабораторной работе №8

по курсу «Программирование»

на тему "Многомодульные программы"

Выполнили:

Студенты группы 22ВВ2

Беляев Д. И.

Захаров А. С.

Ипполитов И. Д.

Приняли:

Юрова О. В.

Патунин Д. В.

Пенза 2022

**Название**Многомодульные программы

**Цель работы**Изучение принципов написания многомодульных программ и приобретение навыков в написании таких программ и работе с ними.

**Лабораторное задание**

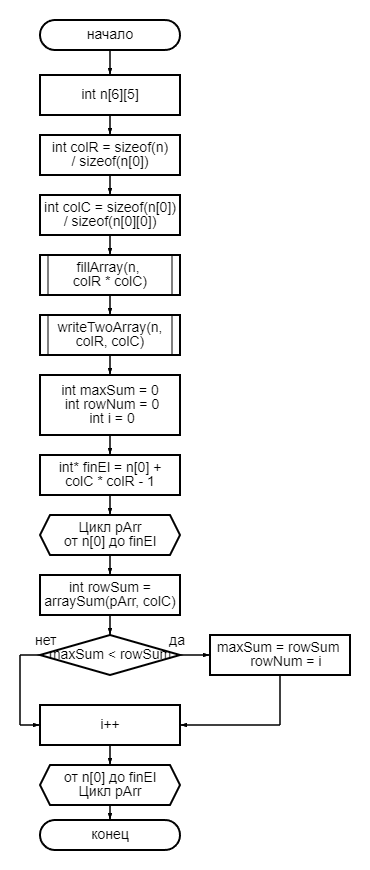
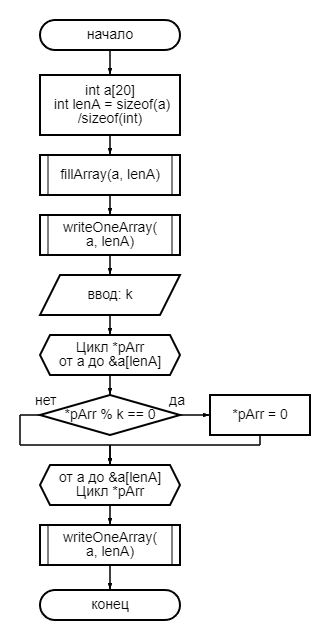
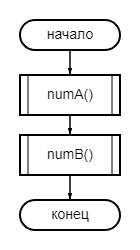
1. Выбрать вариант задания в соответствии с номером бригады.

2. Каждая функция 8-ой лабораторной работы должна быть помещена в отдельный файл. Далее в среде необходимо создать проект, в который включаются все файлы с функциями.

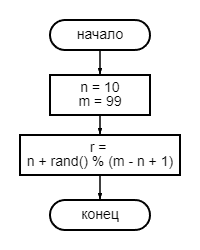
**Ход работы:**

1. **Схема программы**

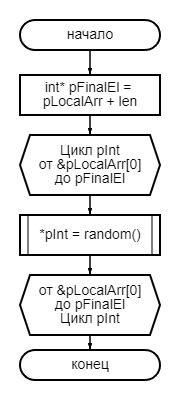
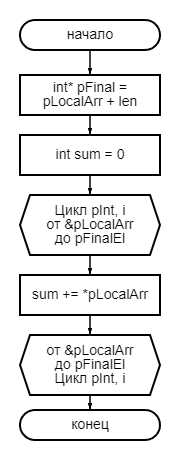
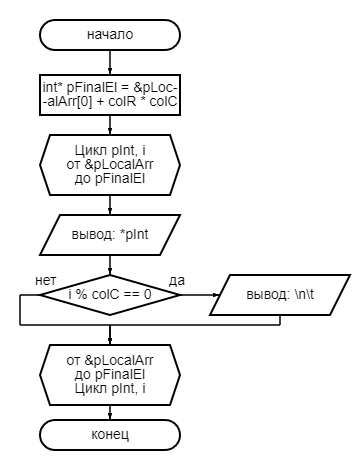
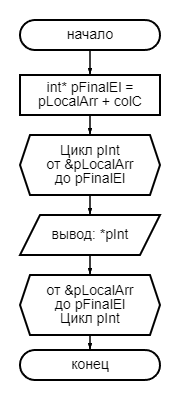
* **main.c**



* **func.h**



* **arrayWork.h**



1. **Код программы**

* **main.c**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "arrayWork.h"

int numA();

int numB();

int main()

{

srand(time(NULL));

numA();

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

numB();

return 0;

}

int numA()

{

int a[20];

int lenA = sizeof(a) / sizeof(int);

fillArray(a, lenA);

writeOneArray(a, lenA);

printf("\nInsert dop number: ");

int k = 0;

scanf("%d", &k);

printf("\n");

for (int\* pArr = a; pArr < &a[lenA]; pArr++)

{

if (\*pArr % k == 0)

\*pArr = 0;

}

writeOneArray(a, lenA);

return 0;

}

int numB()

{

int n[6][5];

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

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

fillArray(n, colR \* colC);

writeTwoArray(n, colR, colC);

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)

{

int rowSum = arraySum(pArr, colC);

if (maxSum < rowSum)

{

maxSum = rowSum;

rowNum = i;

}

i++;

}

printf("Max sum: %d - %d\n", rowNum + 1, maxSum);

return 0;

}

* **func.h**

#ifndef FUNC\_H\_INCLUDED

#define FUNC\_H\_INCLUDED

#include <stdlib.h>

#include <time.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;

}

#endif // FUNC\_H\_INCLUDED

* **arrayWork.h**

#ifndef ARRAY\_WORK\_H\_INCLUDED

#define ARRAY\_WORK\_H\_INCLUDED

#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;

int sum = 0;

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

{

sum += \*pLocalArr;

}

return sum;

}

#endif // !ARRAY\_WORK\_H\_INCLUDED

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

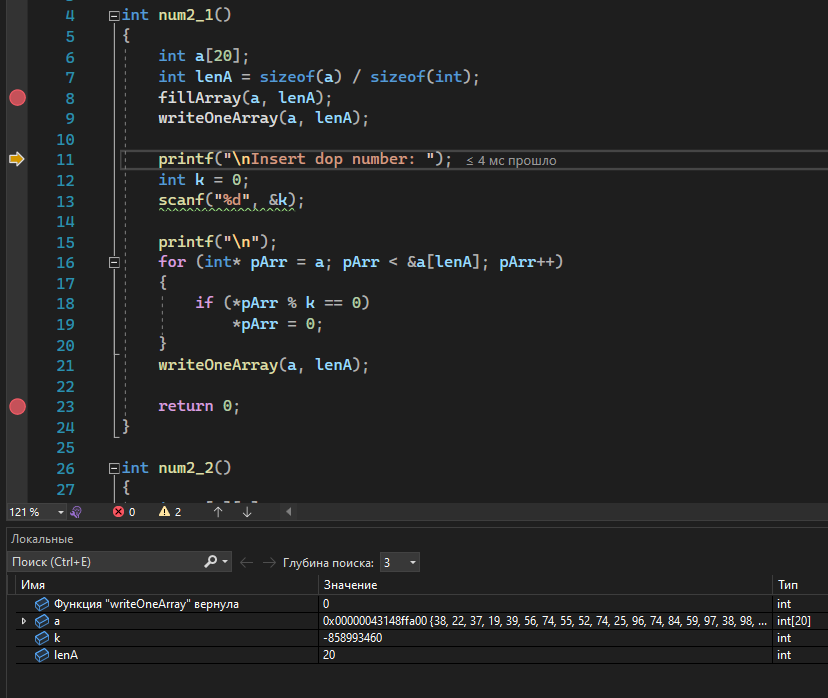
**79 81 73 75 59 - 367**

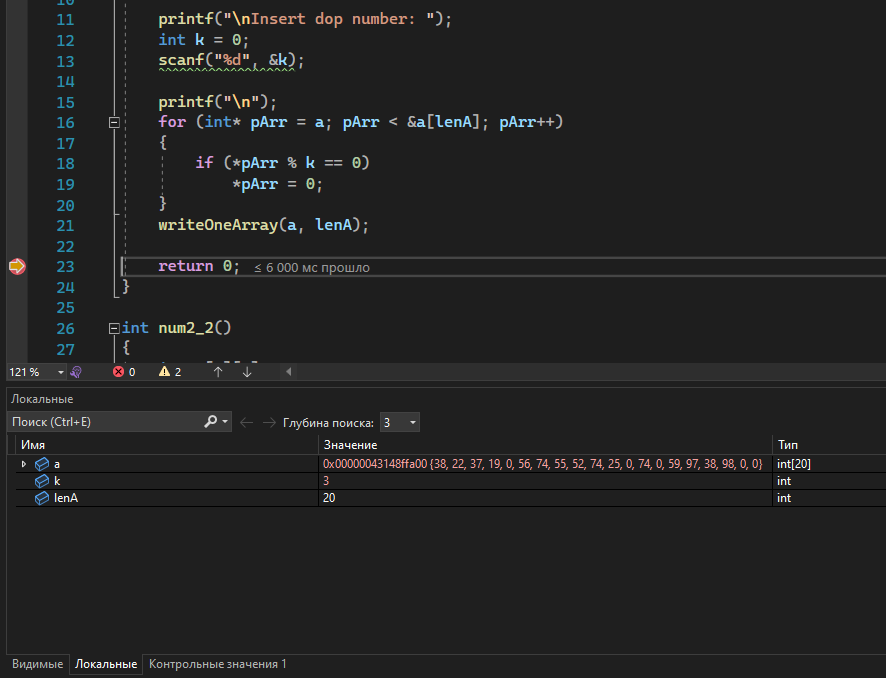
58 34 40 37 66 - 235

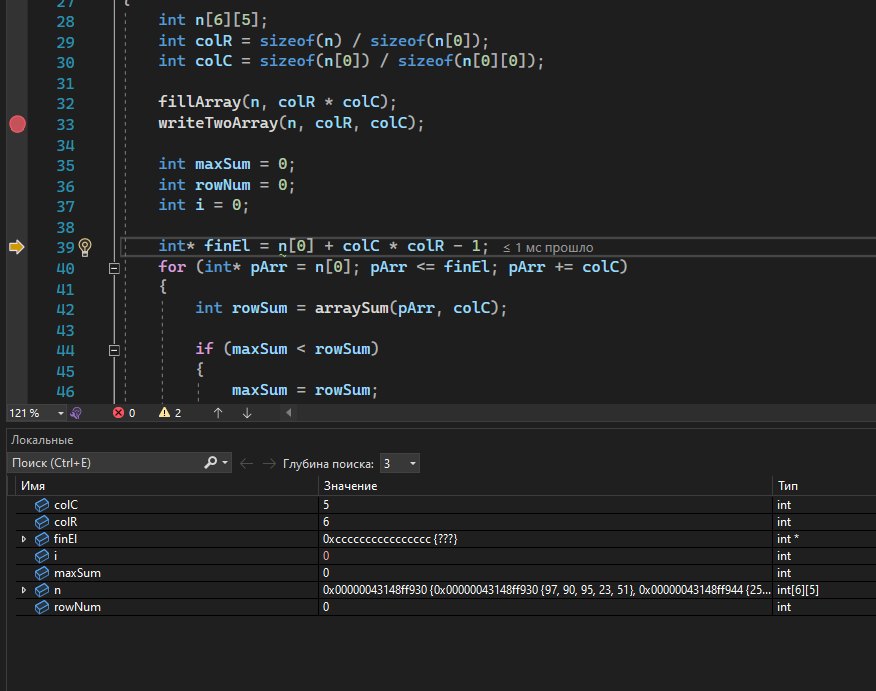
93 76 81 45 24 - 319

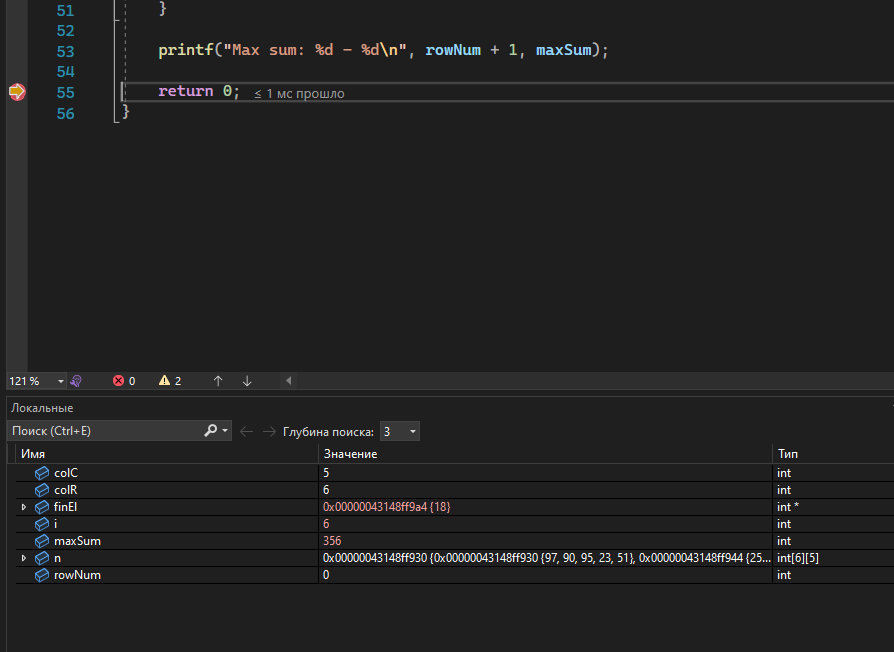
92 70 79 57 13 - 311

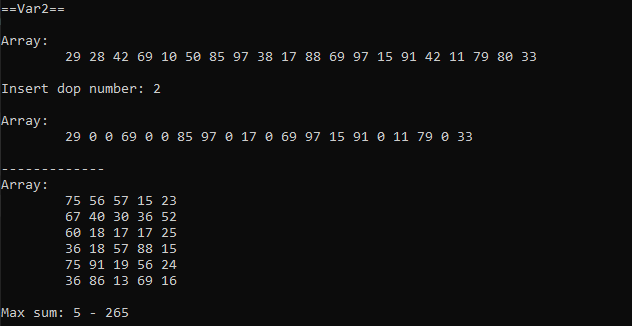
1. **Пояснительный текст к программе**
   1. Массив A длиной 20 заполняется псевдослучайными числами. Задается число k. После перебирается каждый символ массива и проверяется: если элемент массива кратен числу k, то он заменяется на ноль 0, если нет, то остается без изменений.
   2. Матрица (двумерный массив) с размерами 6x5 задается случайным образом. После этого программа построчно проверяет сумму всех элементов каждой строчки, находит наибольшую и выводит ее.
2. **Трассировка**









1. **Результат работы программы**

**Вывод:**Мы изучили принципы написания многомодульных программ и приобрели навыки в написании таких программ и работе с ними. Благодаря этому мы сделали программу в который используются полученные нами навыки по созданию многомодульныx программ.