Министерство образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

ОТЧЕТ

По лабораторной №1  
На тему: Простые структуры данных  
По дисциплине: «Логика и основы алгоритмизации в инженерных задачах»

Выполнили Студенты группы 20ВВ2:  
Борунов Д.А

Сурков М.С.

Приняли:  
Митрохин М.А.  
Юрова О.В.

Пенза 2021г

**Цель работы:** вспомнить навыки работы со статическими и динамическими массивами структурами данных, принцип работы функции генерации случайных чисел на языке Си.

**Ход работы**

**Задание 1:** написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

**Метод решения программы:**

1. Объявить статический массив из n символов и две переменные c, d
2. Описать 2 функции поиска максимального и минимального элемента массива, на вход которых будет подаваться массив, в качестве указателя, из n элементов.

**Функция поиска максимального элемента массива:**

int find\_max(int\* b)

{

int i, max = 0;

for (i = 0; i < 8; i++)

{

if (b[i] > max)

{

max = b[i];

}

}

return max;

**Функция поиска минимального элемента массива:**

int find\_min(int\* b)

{

int i, min= 100;

for (i = 0; i < 8; i++)

{

if (b[i] < min)

{

min = b[i];

}

}

return min;

}

Цикл for позволяет пройтись по всем элементам массива, сравнивая их с локальными переменными max и min. Переменным max и min присваивается значение элемента массива на текущей итерации цикла, если оно больше или меньше max и min соответственно. После цикла обе функции возвращают значения max и min.

1. Присваиваем **c** и **d** значения, которые возвращают функции **find\_max** и  
   **find\_min** соответственно, находим разницу этих значений.

**Код программы для Задания 1**

#include <stdio.h>

#include <conio.h>

int find\_max(int\* b)

{

int i, max = 0;

for (i = 0; i < 8; i++)

{

if (b[i] > max)

{

max = b[i];

}

}

return max;

}

int find\_min(int\* b)

{

int i, min= 100;

for (i = 0; i < 8; i++)

{

if (b[i] < min)

{

min = b[i];

}

}

return min;

}

int main() {

int a[8] = { 9, 5, 20, 10, 14, 16, 18, 8 };

int c, d;

printf("intial array: ");

for (c = 0; c < 8; c++) {

printf("%d ", a[c]);

}

c = find\_max(a);

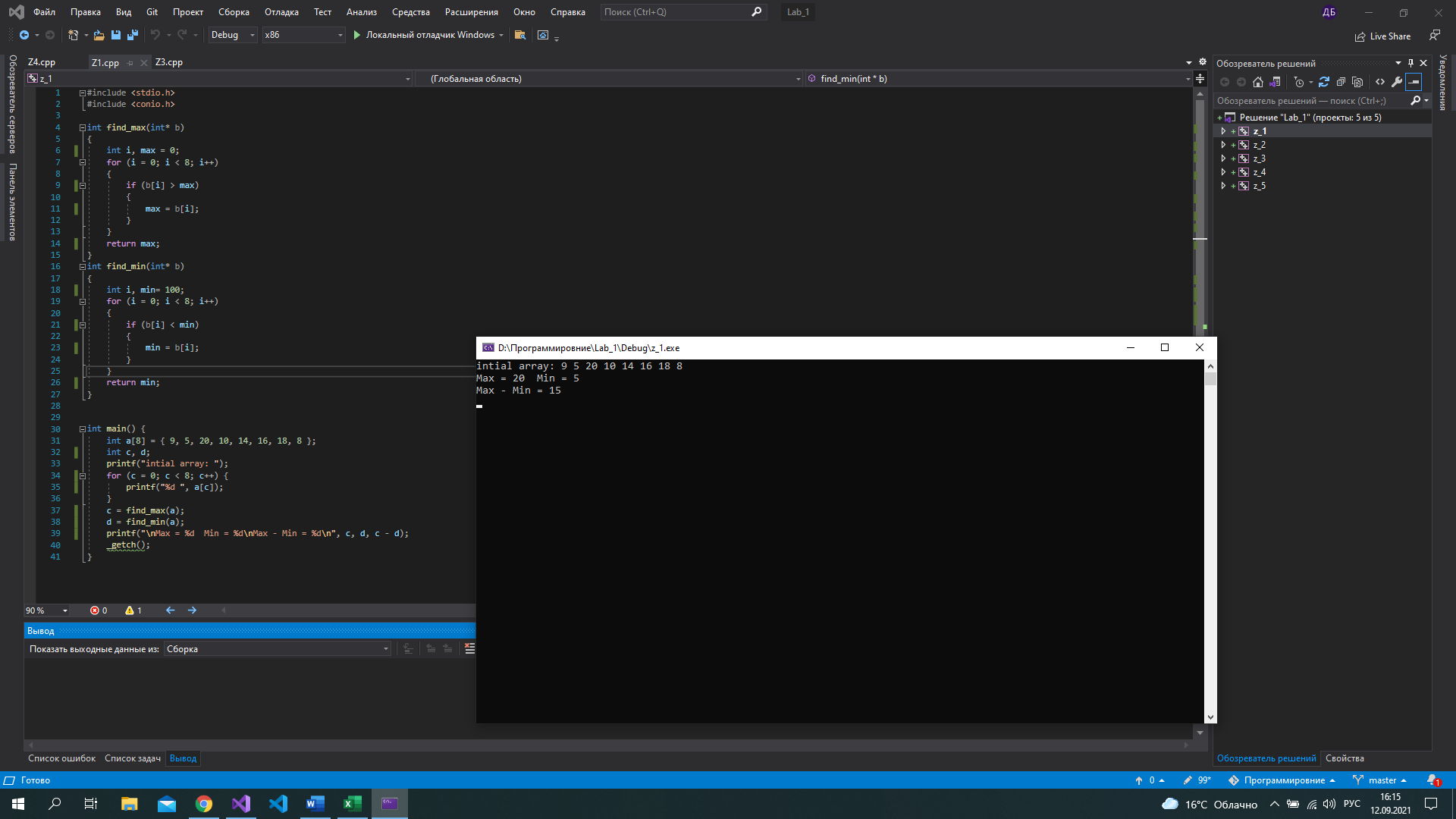
d = find\_min(a);

printf("\nMax = %d Min = %d\nMax - Min = %d\n", c, d, c - d);

\_getch();

}

**Результат работы программы**



**Задание 2:** написать программу, реализующую инициализацию массива

случайными числами.

**Метод решения**

1. Объявить массив **n** размера.
2. Используя цикл **for** и **rand()** проинициализировать каждый элемент массива.

**Код программы для Задания 2:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

#define SIZEOFMASS 10

int main() {

srand(time(NULL));

int i, a[SIZEOFMASS];

printf("initial array: ");

for (i = 0; i < SIZEOFMASS; i++)

{

a[i] = rand() % 100;

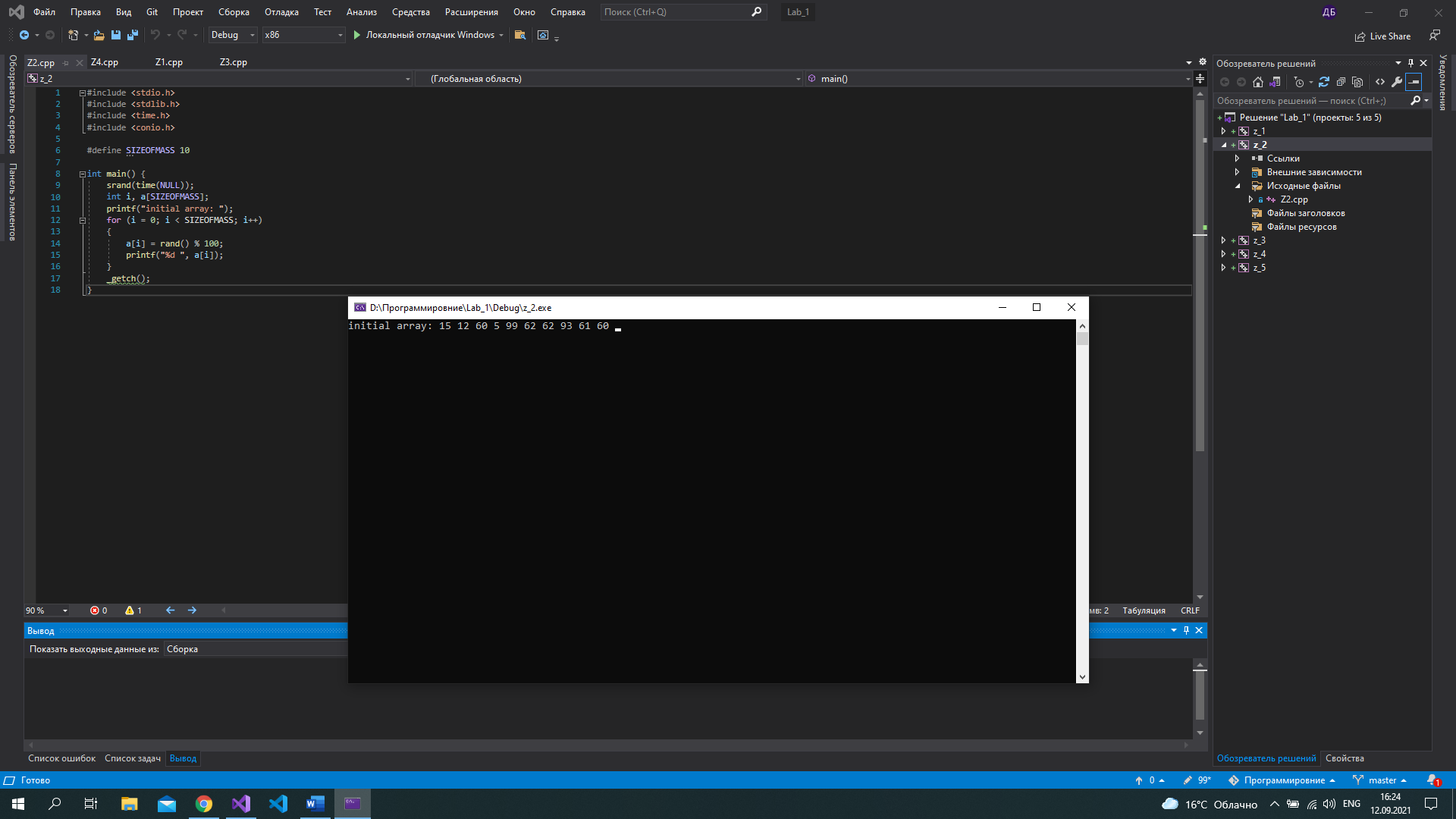
printf("%d ", a[i]);

}

\_getch();

}

**Результат работы программы**



**Задание 3:** написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

**Метод решения**

1. Считать с клавиатуры максимальное количество элементов **max**
2. Инициализировать указатель **а** на фрагмент памяти, объявленной при помощи функции **malloc,** размера **max\*sizeof(int)**
3. При помощи цикла **for** считать с клавиатуры необходимое количество элементов.

**Код программы для Задания 3**

#include <stdio.h>

#include <conio.h>

#include <malloc.h>

int main() {

int\* a;

int max\_size, i;

printf("max number of elements? ");

scanf\_s("%d", &max\_size);

a = (int\*)malloc(max\_size \* sizeof(int));

printf("\nenter %d numbers:\n", max\_size);

for (i = 0; i < max\_size; i++)

{

scanf\_s("%d", &a[i]);

}

printf("\nyour array: ");

for (i = 0; i < max\_size; i++)

{

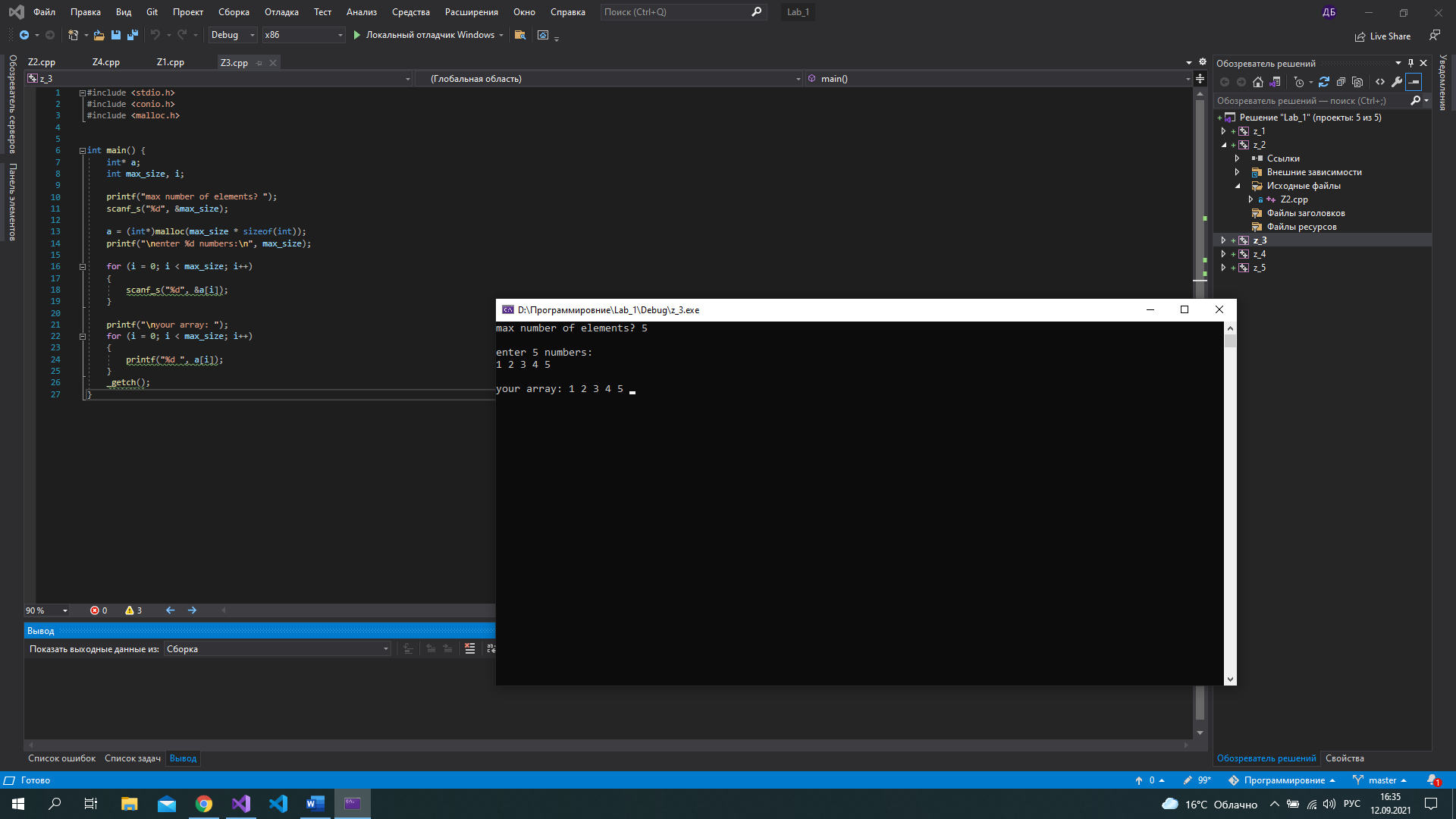
printf("%d ", a[i]);

}

\_getch();

}

**Результат работы программы**



**Задание 4:** написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

**Метод решения**

1. Объявить двумерный массив a[m][n]
2. Проинициализировать объявленный массив случайными числами функцией **rand(),** используя цикл **for**, вложенный в цикл **for**. Внешний цикл индексирует номер строки, внутренний индексирует номер столбца.

**Код программы для Задания 4**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

int main() {

srand(10);

int const m = 5, n = 5;

int a[m][n], i, j, sum\_str = 0, sum\_col = 0;

for (i = 0; i < m; i++)

{

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

{

a[i][j] = rand() % 10;

printf("%4d", a[i][j]);

}

printf("\n");

}

for (i = 0; i < m; i++)

{

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

{

sum\_str += a[i][j];

}

printf("summ of %d str = %d\n", i + 1, sum\_str);

sum\_str = 0;

}

printf("\n");

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

{

for (i = 0; i < m; i++)

{

sum\_col += a[i][j];

}

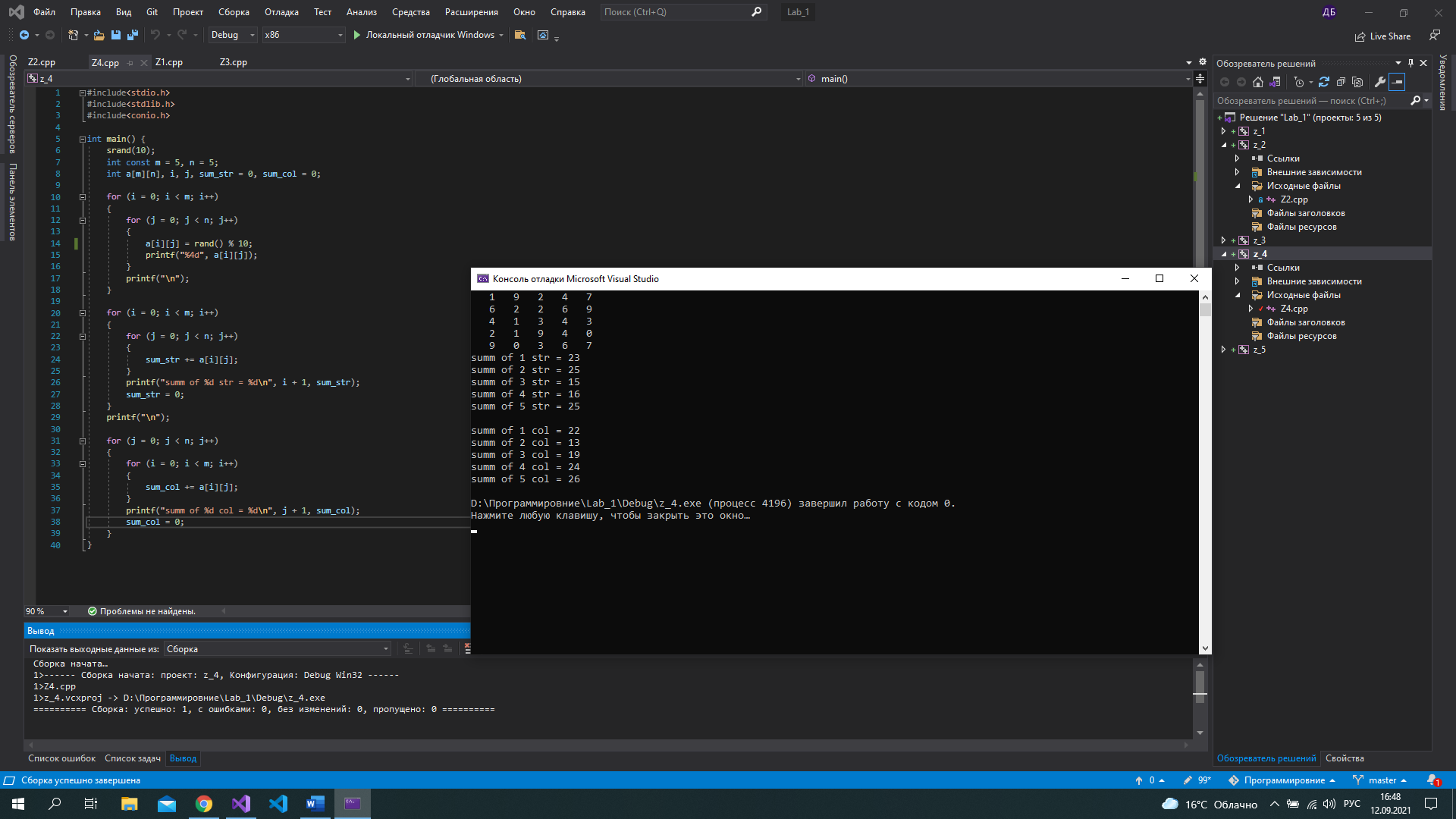
printf("summ of %d col = %d\n", j + 1, sum\_col);

sum\_col = 0;

}

}

**Результат работы программы**



**Задание 5**: написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

**Метод решения**

1. Объявить массив структур **student. (**поля: фамилия, имя, название факультета)
2. Заполнить структуры данными, введенными с клавиатуры.
3. Запросить у пользователя слово, по которому нужно найти структуру (фамилия или имя или название факультета)
4. Выполнить поиск введенного слова в инициализированных структурах. Сравнение строк проводить при помощи функции **strcmp().** При наличии совпадения выводить структуру на экран.

**Код программы Задания 5**

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#include <iostream>

#include <stdio.h>

#include <string.h>

#include <cstdlib>

#include <windows.h>

#include <conio.h>

#include <ctime>

#include <cmath>

#define MAX\_STUDENT 3

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i;

char insert[20];

struct student

{

char famil[20];

char name[20];

char facult[20];

} stud[MAX\_STUDENT];

for (i = 0; i < MAX\_STUDENT; i++)

{

printf("Введите фамилию студента: ");

scanf("%s", &stud[i].famil);

printf("Введите имя студента: ");

scanf("%s", &stud[i].name);

printf("Введите название факультета студента: ");

scanf("%s", &stud[i].facult);

}

while (TRUE) {

printf("\nКого вы хотите найти (фамилия или имя или факультет)? ");

scanf("%s", &insert);

printf("\nРезультат поиска:");

for (i = 0; i < MAX\_STUDENT; i++)

{

if (strcmp(insert, stud[i].famil) == 0 or strcmp(insert, stud[i].name) == 0 or strcmp(insert, stud[i].facult) == 0)

{

printf("\n%s %s %s", stud[i].famil, stud[i].name, stud[i].facult);

}

else

{

printf("Ничего не найдено");

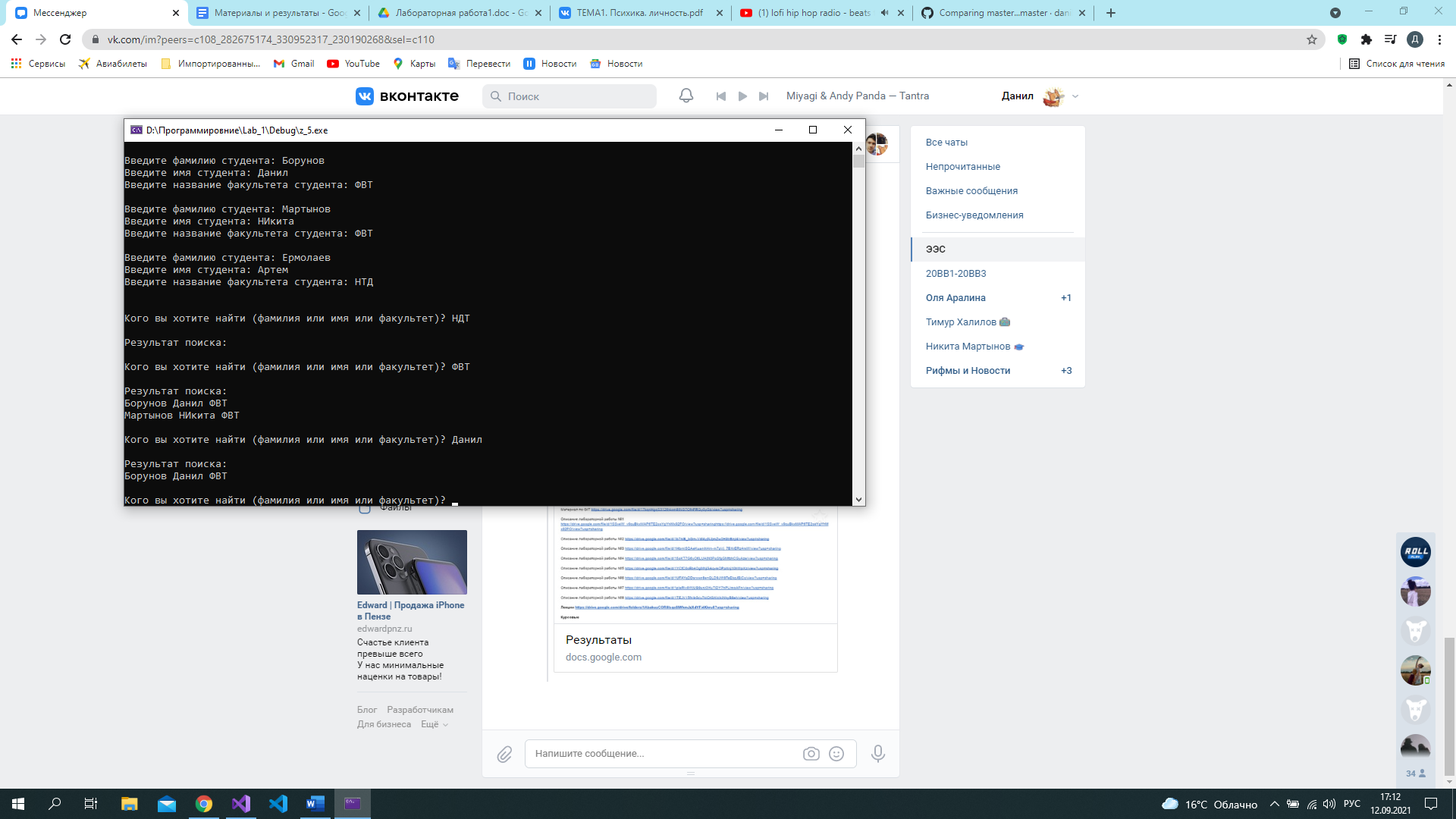
}

}

}

}

**Результат работы программы**



**Вывод:** вспомнили навыки работы со статическими и динамическими массивами структурами данных, принцип работы функции генерации случайных чисел на языке Си.