Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу “Логика и основа алгоритмизации в ИЗ”

на тему “ Простые структуры данных”

Выполнил студент группы 21ВВ3:

Костюков Илья

Приняли:

д.т.н., профессор Митрохин М.А.,

к.т.н., доцент Юрова О.В.

Пенза 2022

**Название**

Простые структуры данных.

**Цель работы:** выполнить лабораторные указания 1-5 используя простые структуры данных.

**Лабораторное задание:**

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

Задание 2: написать программу, реализующую инициализацию массива

случайными числами.

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

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

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

**Задание 1:**

**Листинг**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <Windows.h>

int main(void)

{

SetConsoleOutputCP(1251); //русификация

int A[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

int max = A[0];

int min = A[0];

printf("Массив:");

for (int i = 0; i < 10; i++)

{

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

if (A[i] > max)

max = A[i];

if (A[i] < min)

min = A[i];

}

printf("\n");

printf("Максимальное число: %d\n", max);

printf("Минимальное число: %d\n", min);

int razn = max - min;

printf("Разница: %d", razn);

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1.

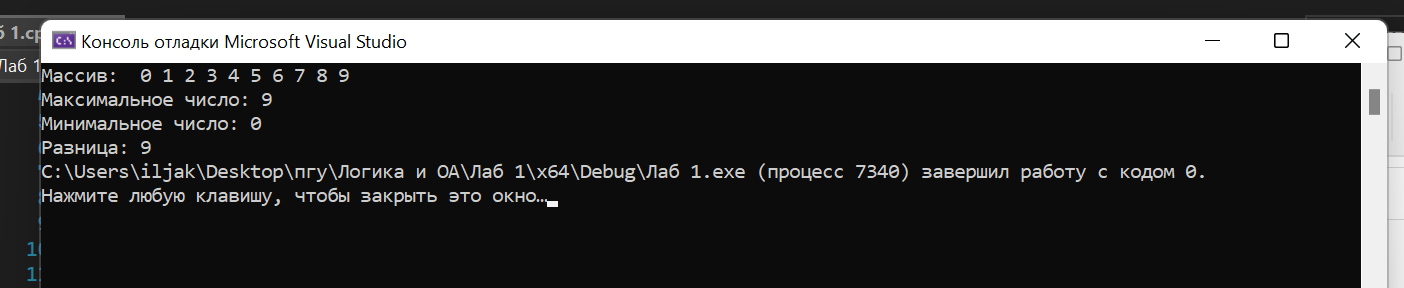


Рис. 1

Протокол трассировки программы показан на рисунке 2 и 3.

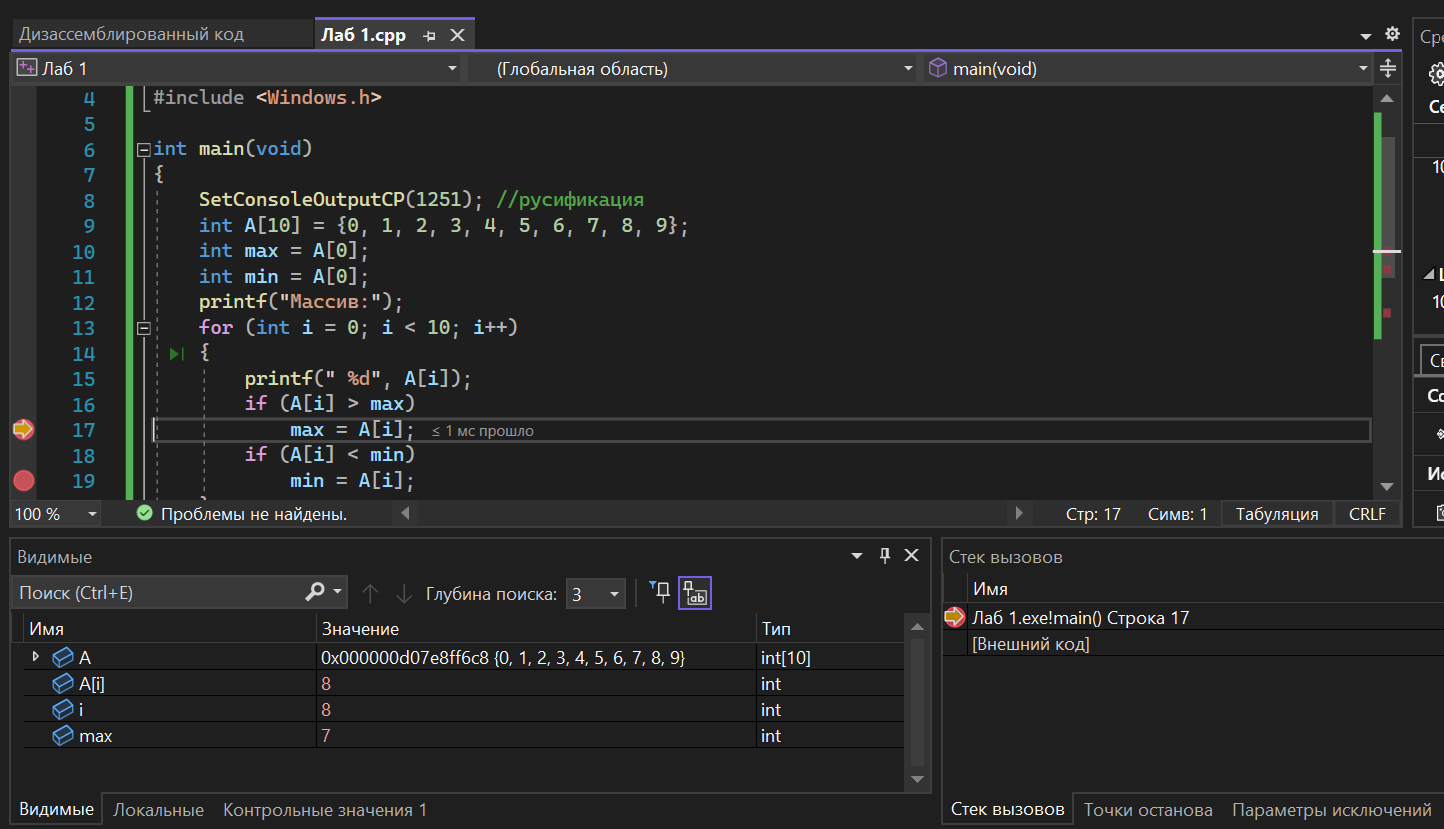


Рис. 2

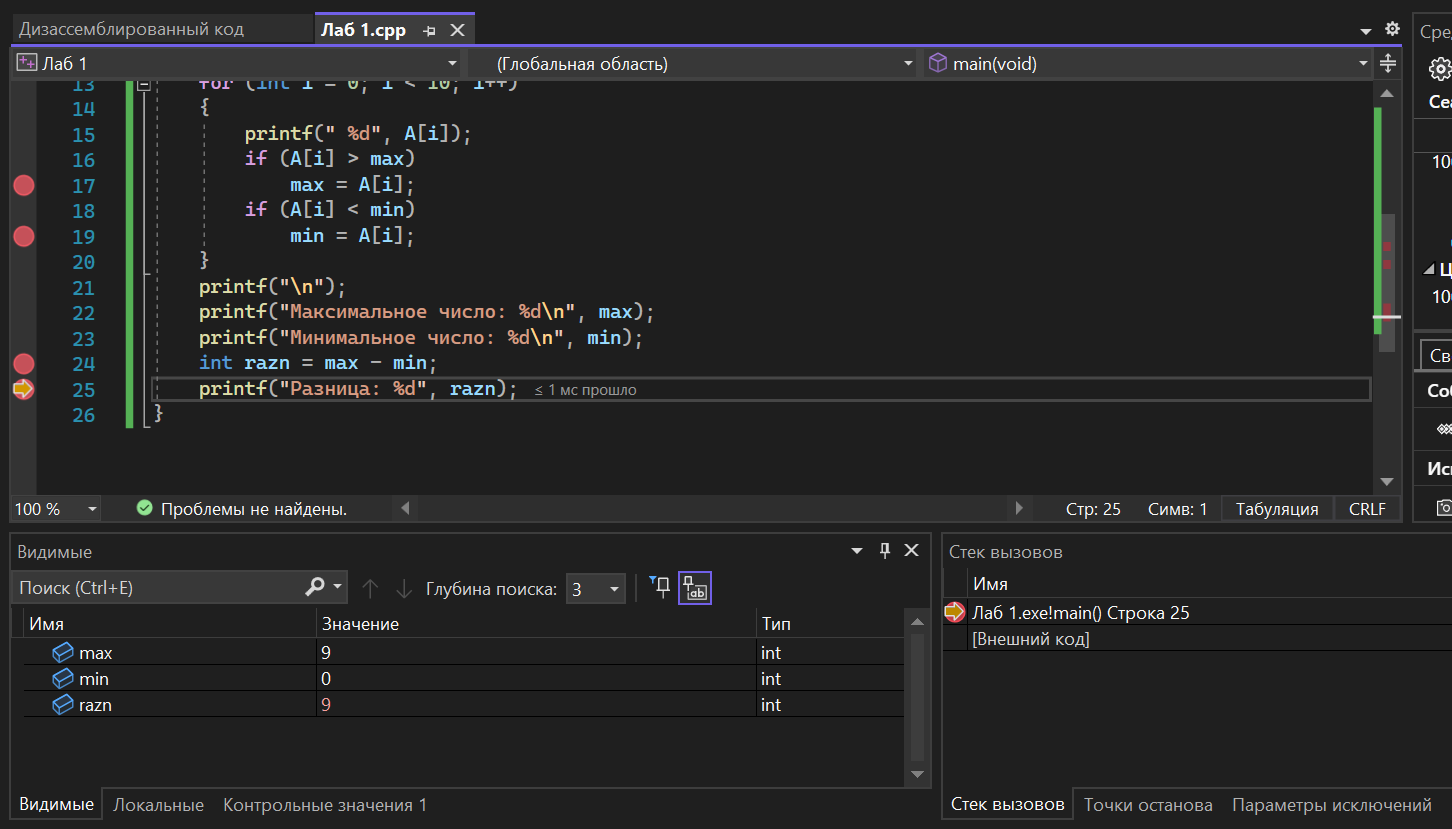


Рис. 3

Результат получился правильный, все действия совершены верно.

Результат работы программы, показанный на рисунке 1, совпал с результатами трассировки на рисунке 2 и 3.

**Задание 2:**

**Листинг**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <Windows.h>

int main(void)

{

SetConsoleOutputCP(1251); //русификация

srand(time(NULL));

int A[10];

printf("Массив, заполненный случайными числами:");

for (int i = 0; i < 10; i++)

{

A[i] = rand() % 100;

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

}

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1.

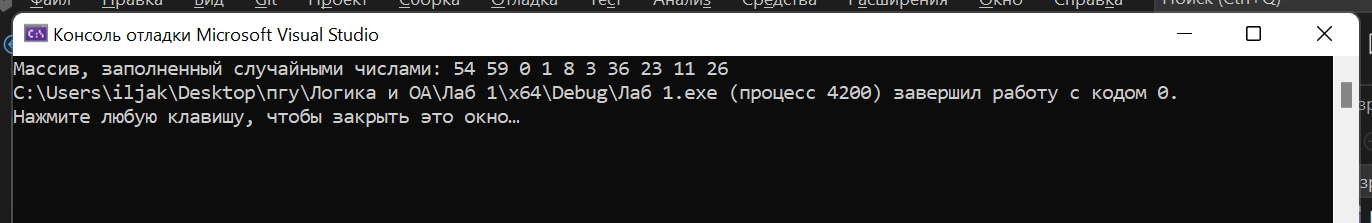


Рис. 1

Протокол трассировки программы показан на рисунке 2 и 3.

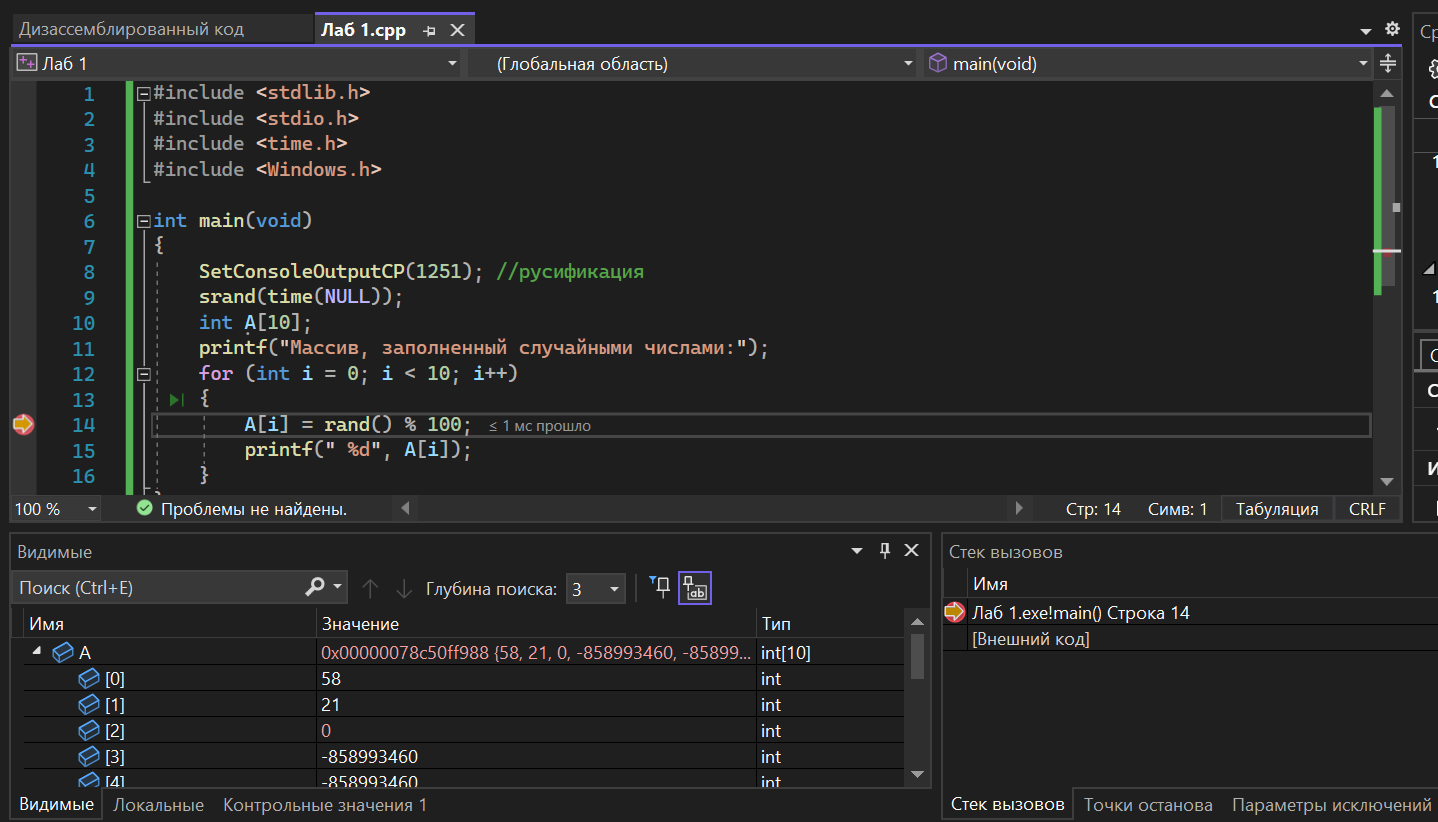


Рис. 2

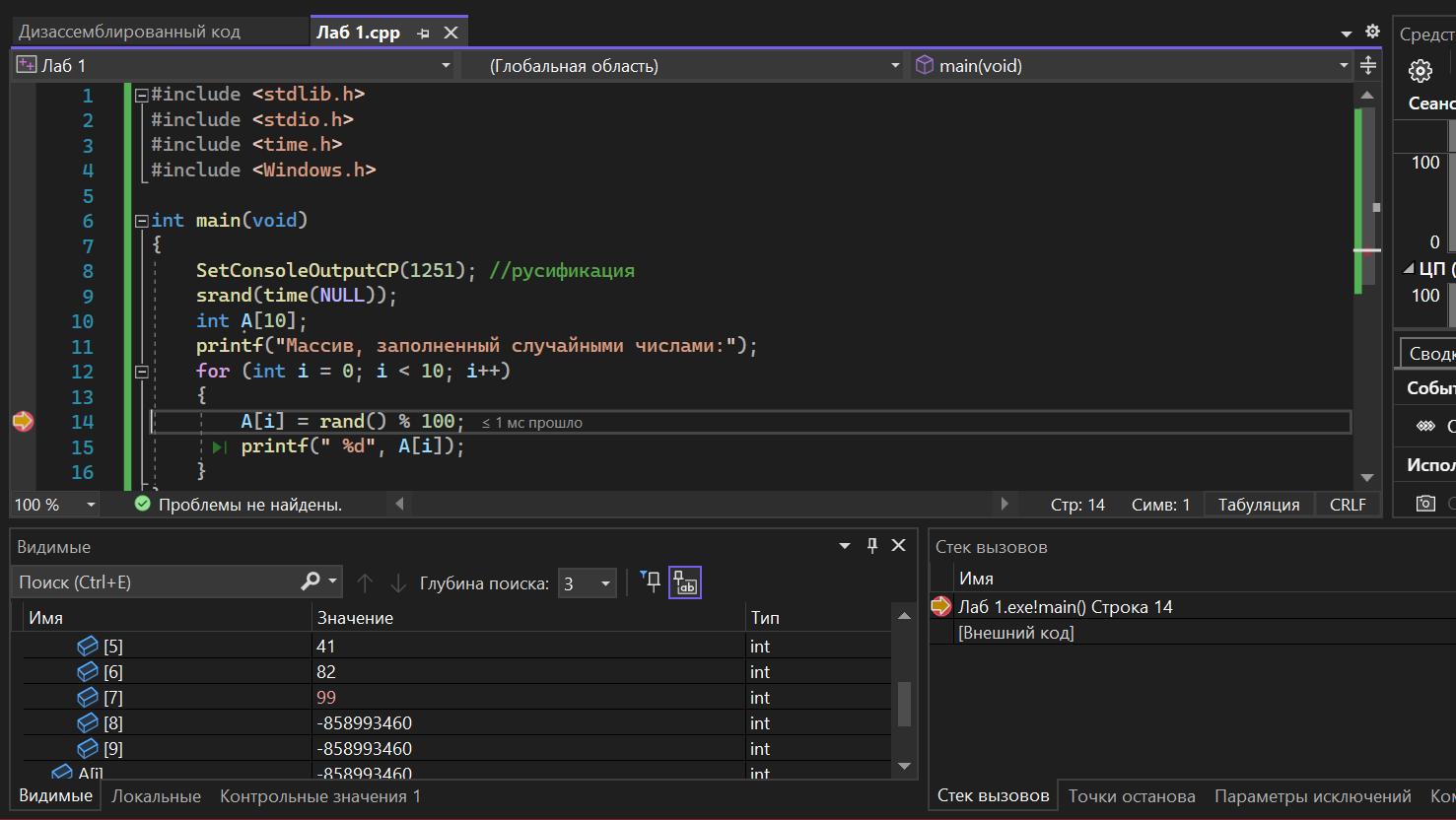


Рис. 3

Результат получился правильный, все действия совершены верно.

Результат работы программы, показанный на рисунке 1, совпал с результатами трассировки на рисунке 2 и 3.

**Задание 3:**

**Листинг**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <Windows.h>

int main(void)

{

SetConsoleOutputCP(1251); //русификация

srand(time(NULL));

int c;

printf("Введите размер массива: ");

scanf\_s("%d", &c);

int \*A = (int\*)malloc(c \* sizeof(int));

printf("Массив произвольного размера: ");

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

{

A[i] = rand() % 100;

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

}

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1.

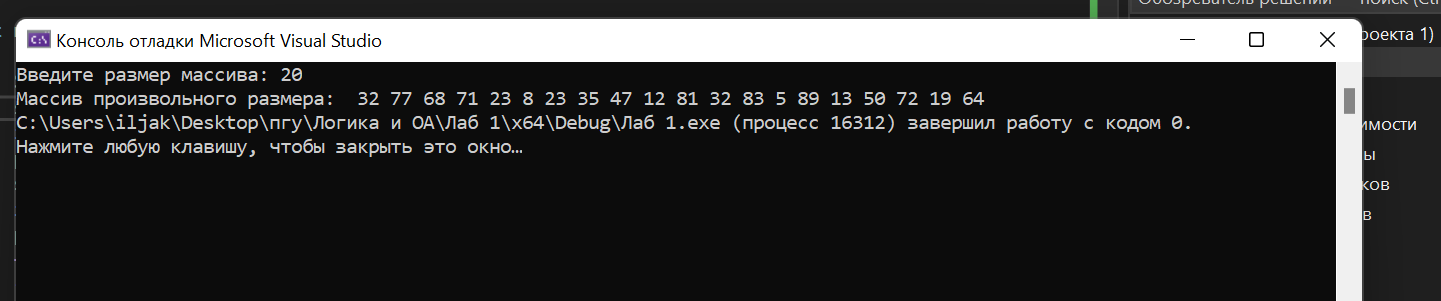


Рис. 1

Протокол трассировки программы показан на рисунке 2 и 3.

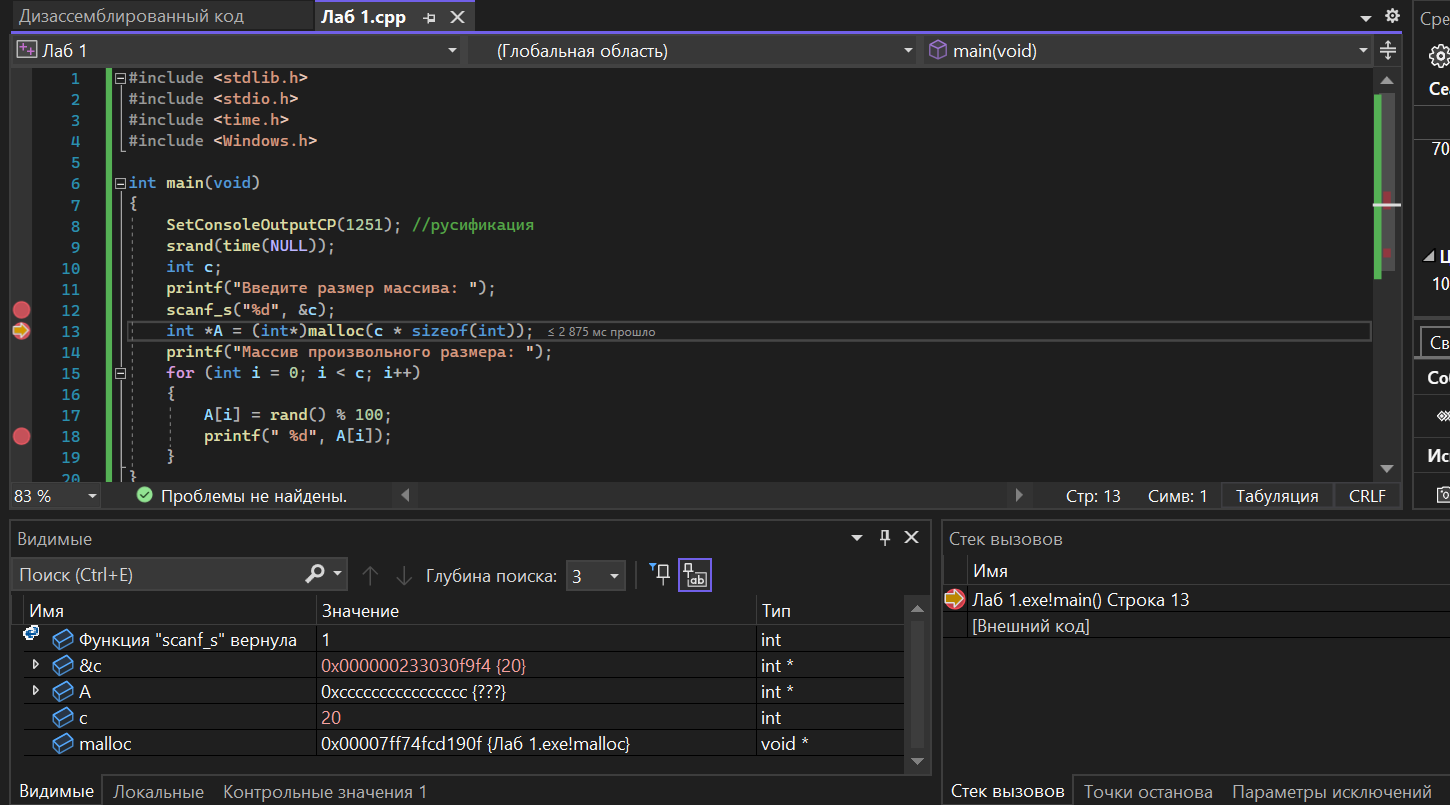


Рис. 2

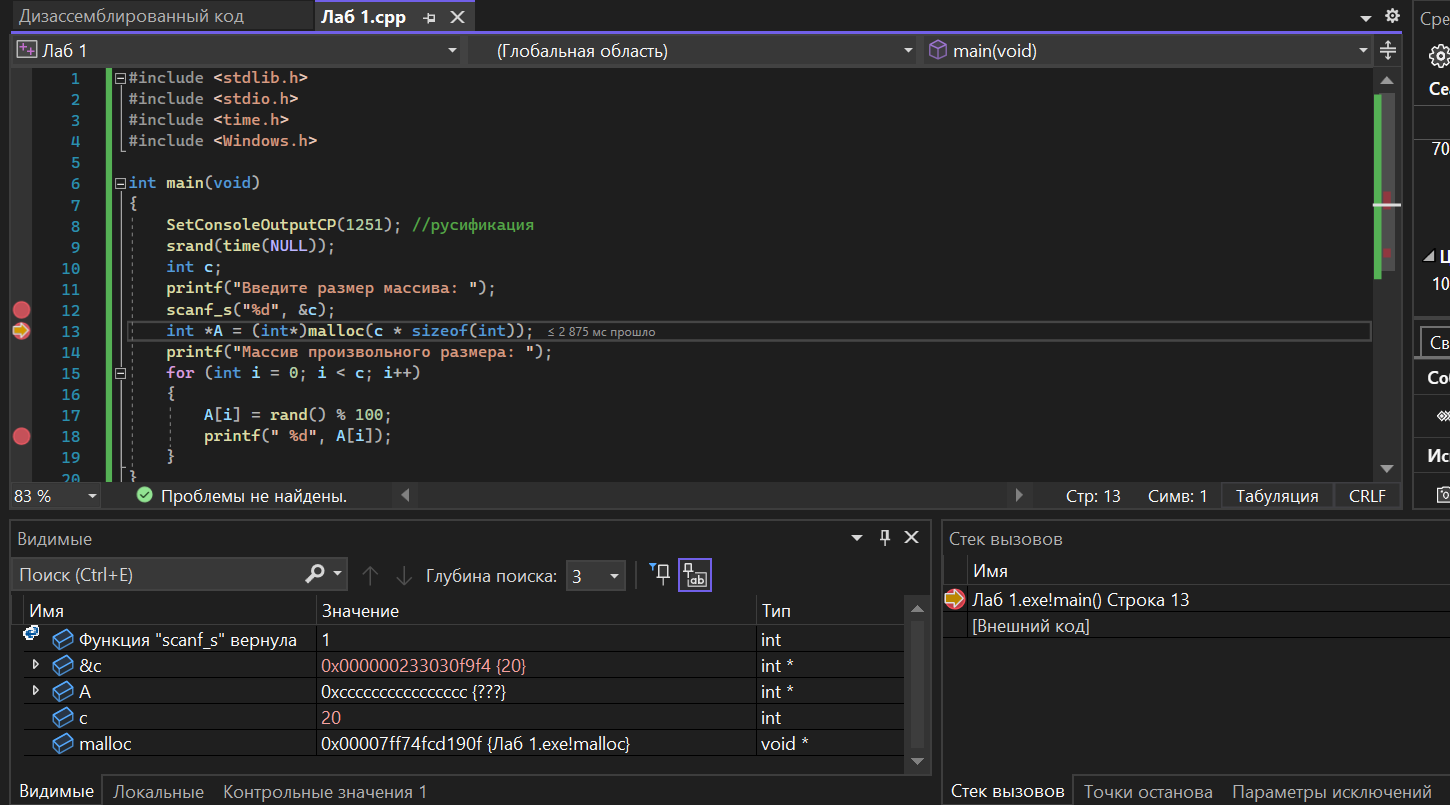


Рис. 3

Результат получился правильный, все действия совершены верно.

Результат работы программы, показанный на рисунке 1, совпал с результатами трассировки на рисунке 2 и 3.

**Задание 4:**

**Листинг**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <Windows.h>

int main(void)

{

SetConsoleOutputCP(1251); //русификация

srand(time(NULL));

int a, b;

printf("Введите размер двумерного массива:\n");

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

scanf\_s("%d", &b);

int\*\* A = (int\*\*)malloc(a \* sizeof(int\*));

for (int j = 0; j < b; j++)

{

A[j] = (int\*)malloc(b \* sizeof(int));

}

printf("Двумерный массив:\n");

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

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

A[i][j] = rand() % 2; //заполнение массива случайными числами

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

}

printf("\n");

}

int \*C = (int\*)malloc(a \* sizeof(int));

int sum = 0;

printf("Сумма элементов в каждой строке:\n");

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

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

sum += A[i][j];

C[i] = sum;

}

printf("%d\n", C[i]);

sum = 0;

}

free(C);

free(A);

return 0;

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1 и 2.

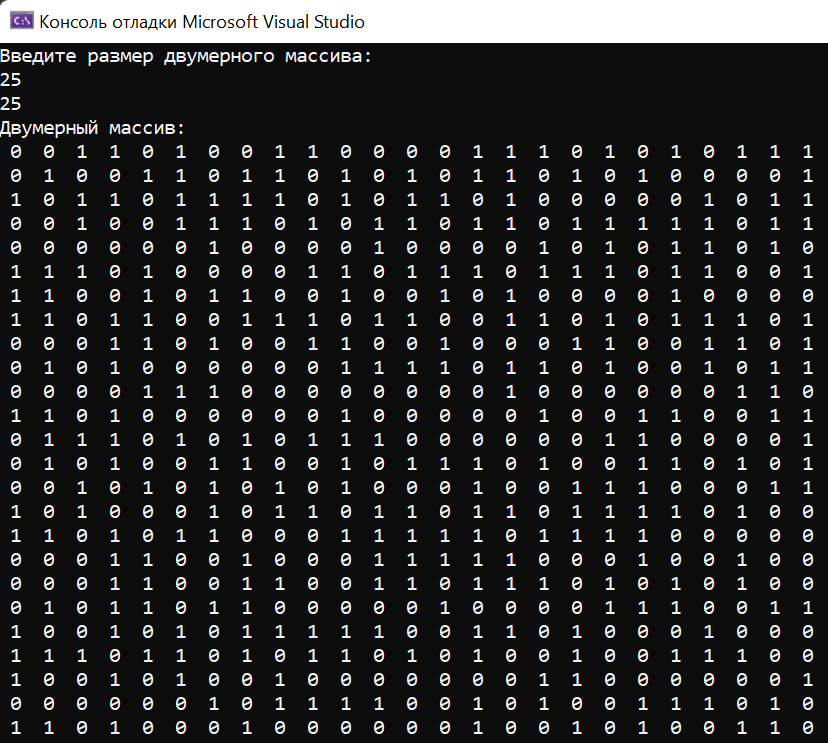
****

Рис. 1

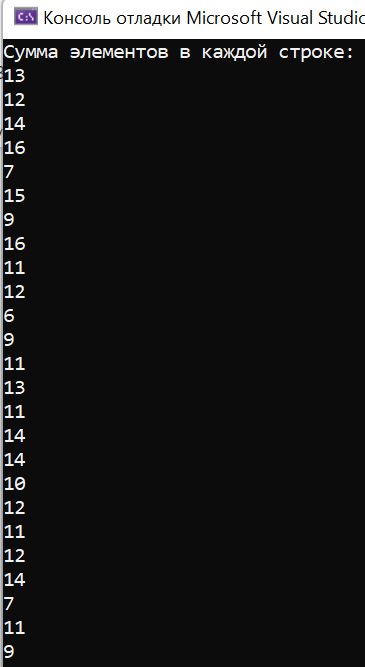


Рис. 2

Протокол трассировки программы показан на рисунке 3 и 4.

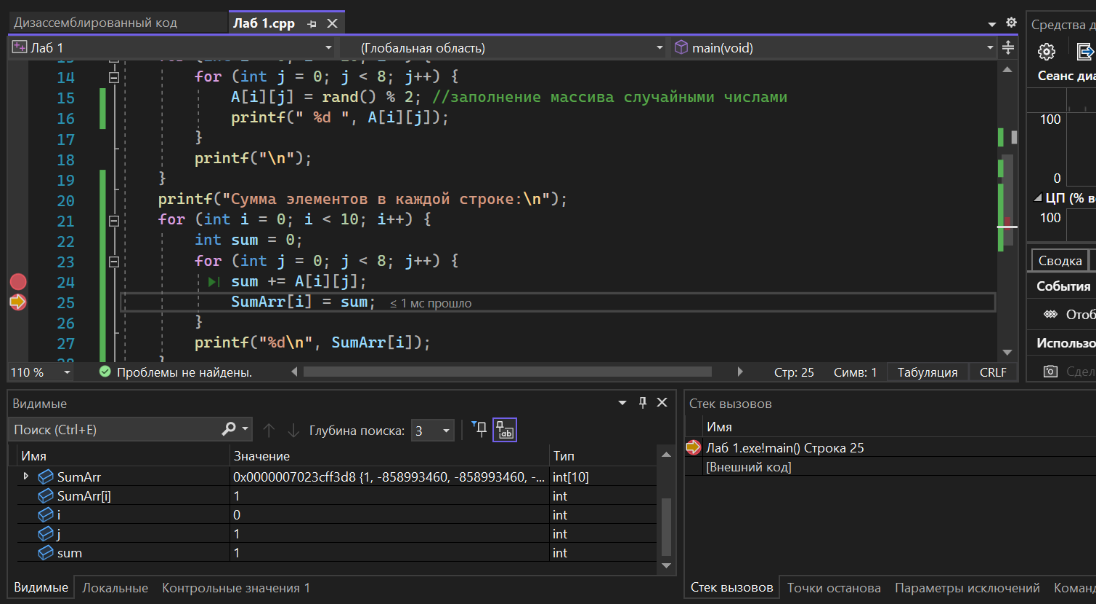


Рис. 3

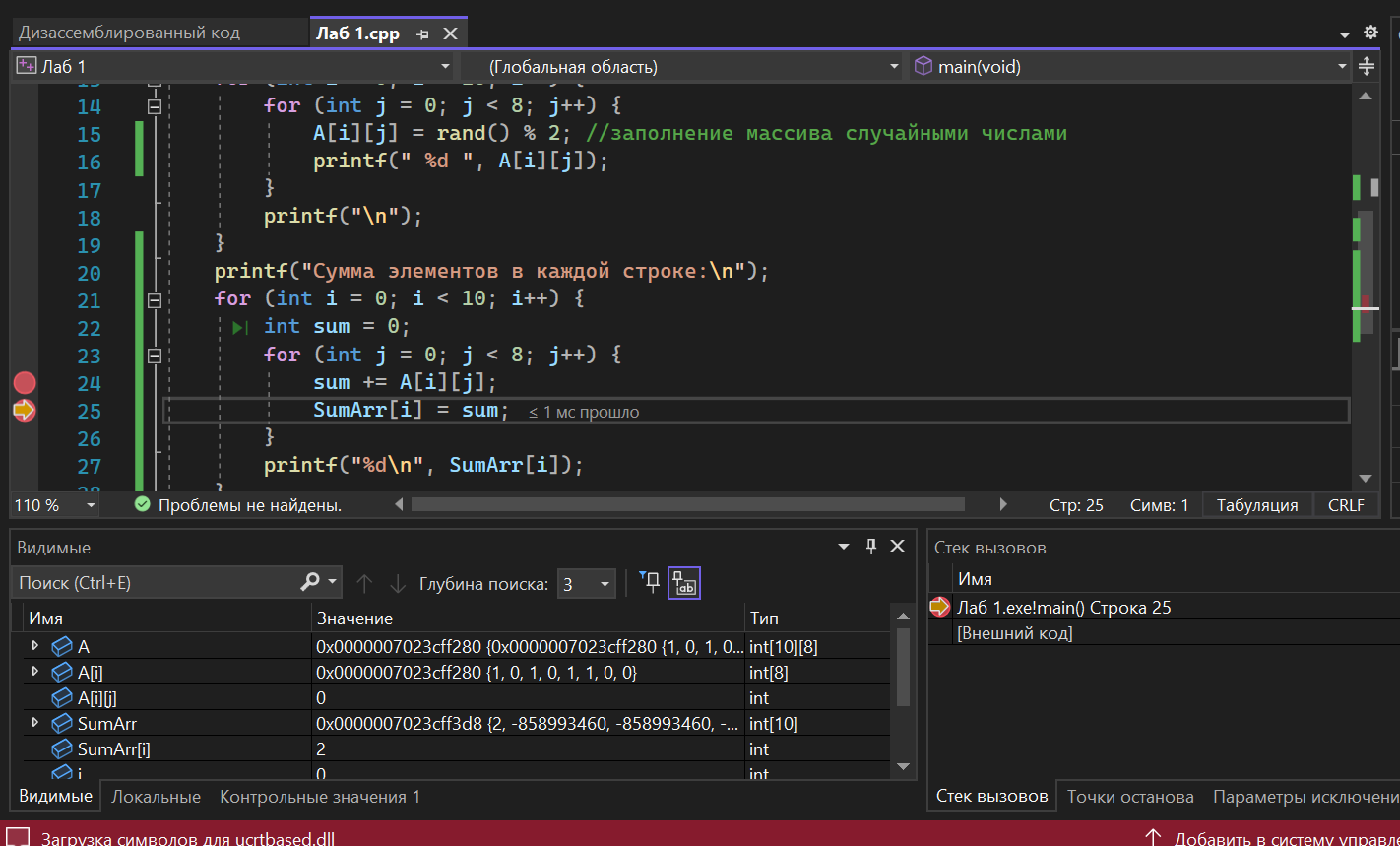


Рис. 4

Результат получился правильный, все действия совершены верно.

Результат работы программы, показанный на рисунке 1 и 2, совпал с результатами трассировки на рисунке 3 и 4.

**Задание 5:**

**Листинг**

#include <stdio.h>

#include <string.h>

#include <Windows.h>

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

struct student {

char famil[20], name[20], facult[20];

}stud[3];

char s1[20];

char s2[20];

char s3[20];

int i;

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

{

printf("Введите фамилию студента:\n");

scanf\_s("%s", &stud[i].famil, 20);

printf("Введите имя студента %s:\n", stud[i].famil);

scanf\_s("%s", &stud[i].name, 20);

printf("Введите название факультета студента %s %s:\n", stud[i].famil, stud[i].name);

scanf\_s("%s", &stud[i].facult, 20);

}

system("cls");

printf("Введите фамилию, имя и факультет студента, которого хотите найти:\n");

scanf\_s("%s", &s1, 20);

scanf\_s("%s", &s2, 20);

scanf\_s("%s", &s3, 20);

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

if (strcmp(s1, stud[i].famil) == 0)

{

if (strcmp(s2, stud[i].name) == 0)

{

if (strcmp(s3, stud[i].facult) == 0)

{

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

}

}

}

}

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1 и 2.

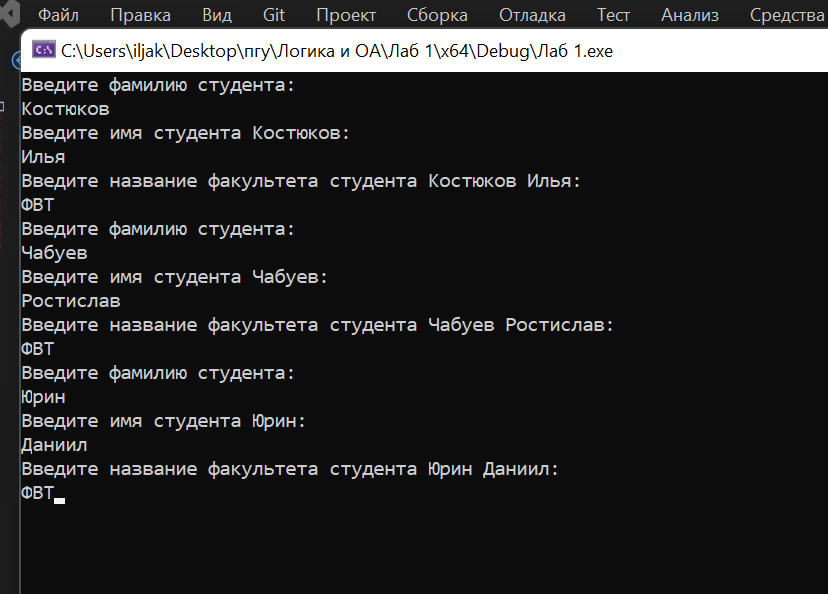


Рис. 1

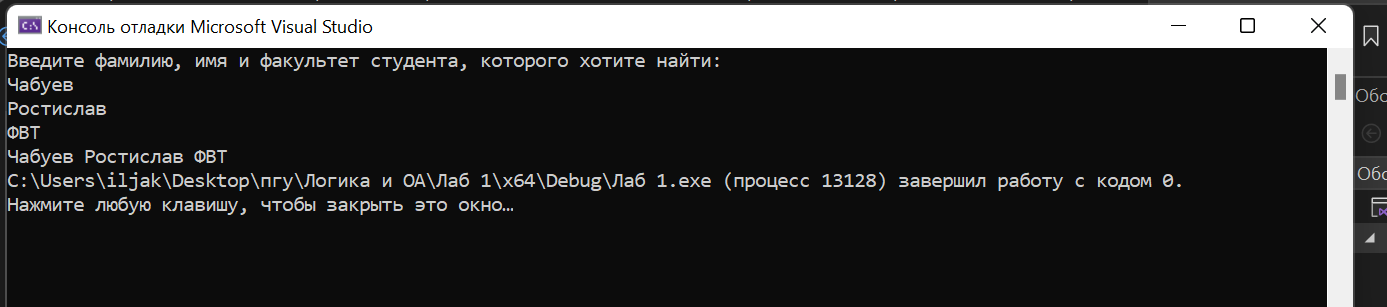


Рис. 2

Протокол трассировки программы показан на рисунке 3-5.

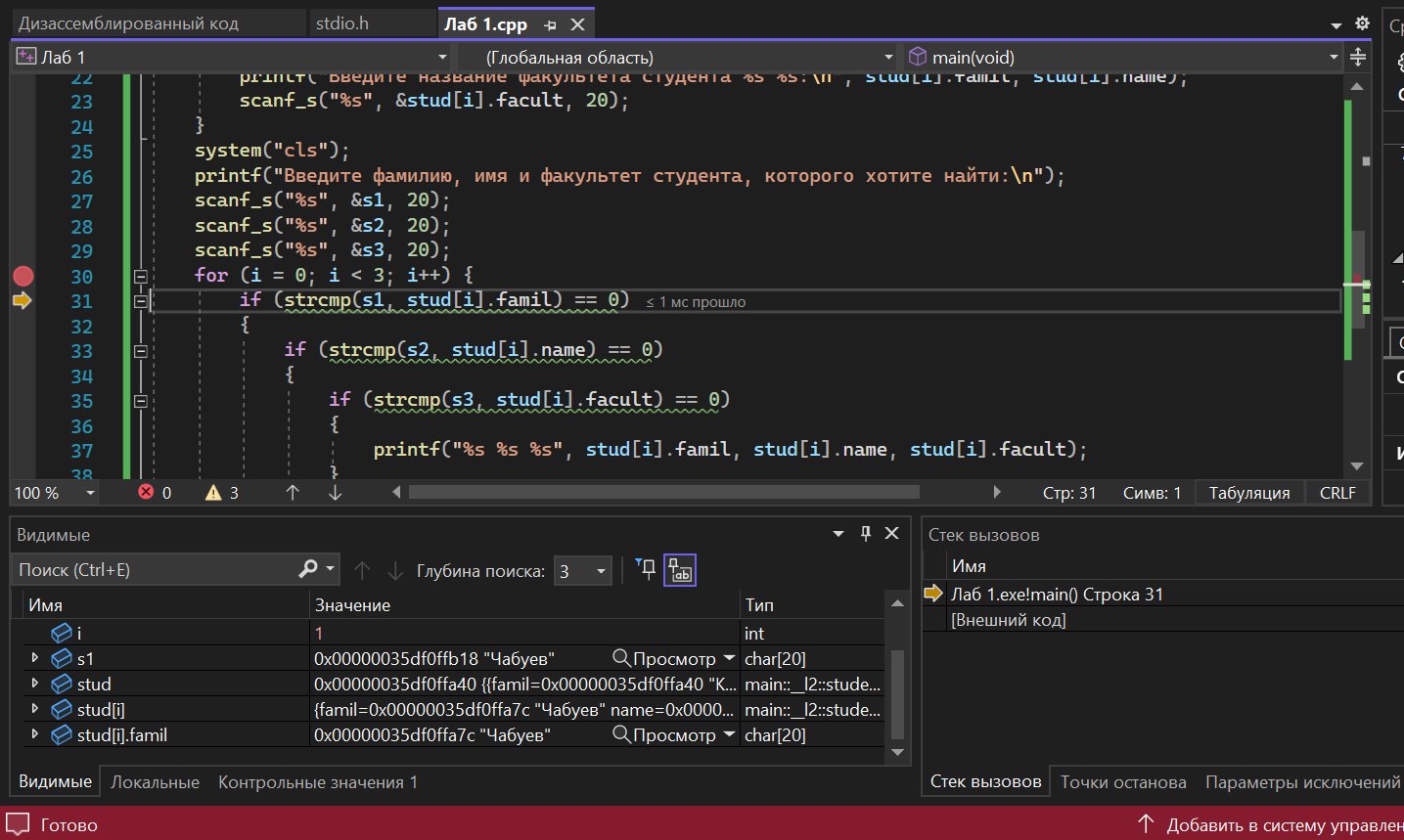


Рис. 3

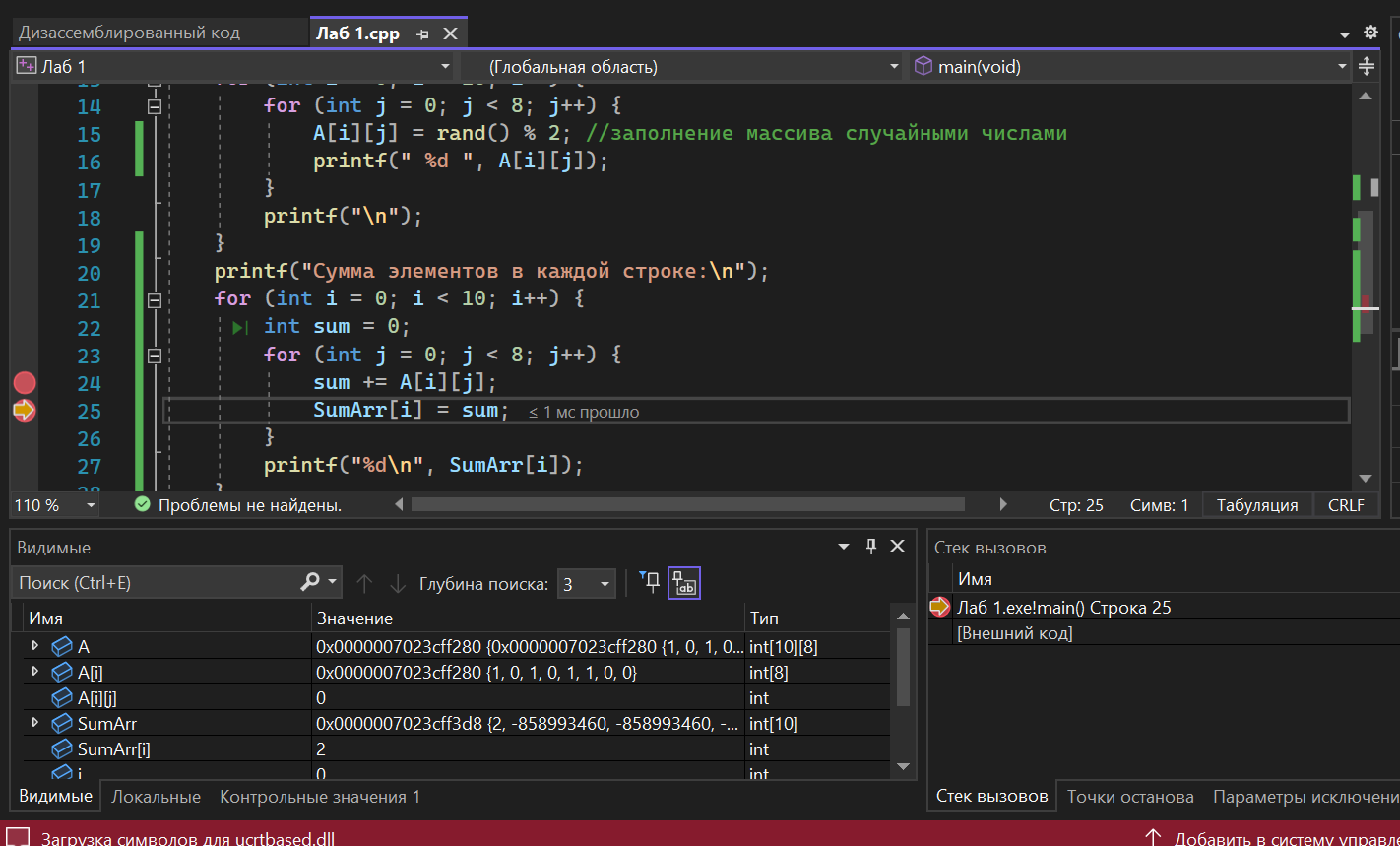


Рис. 4

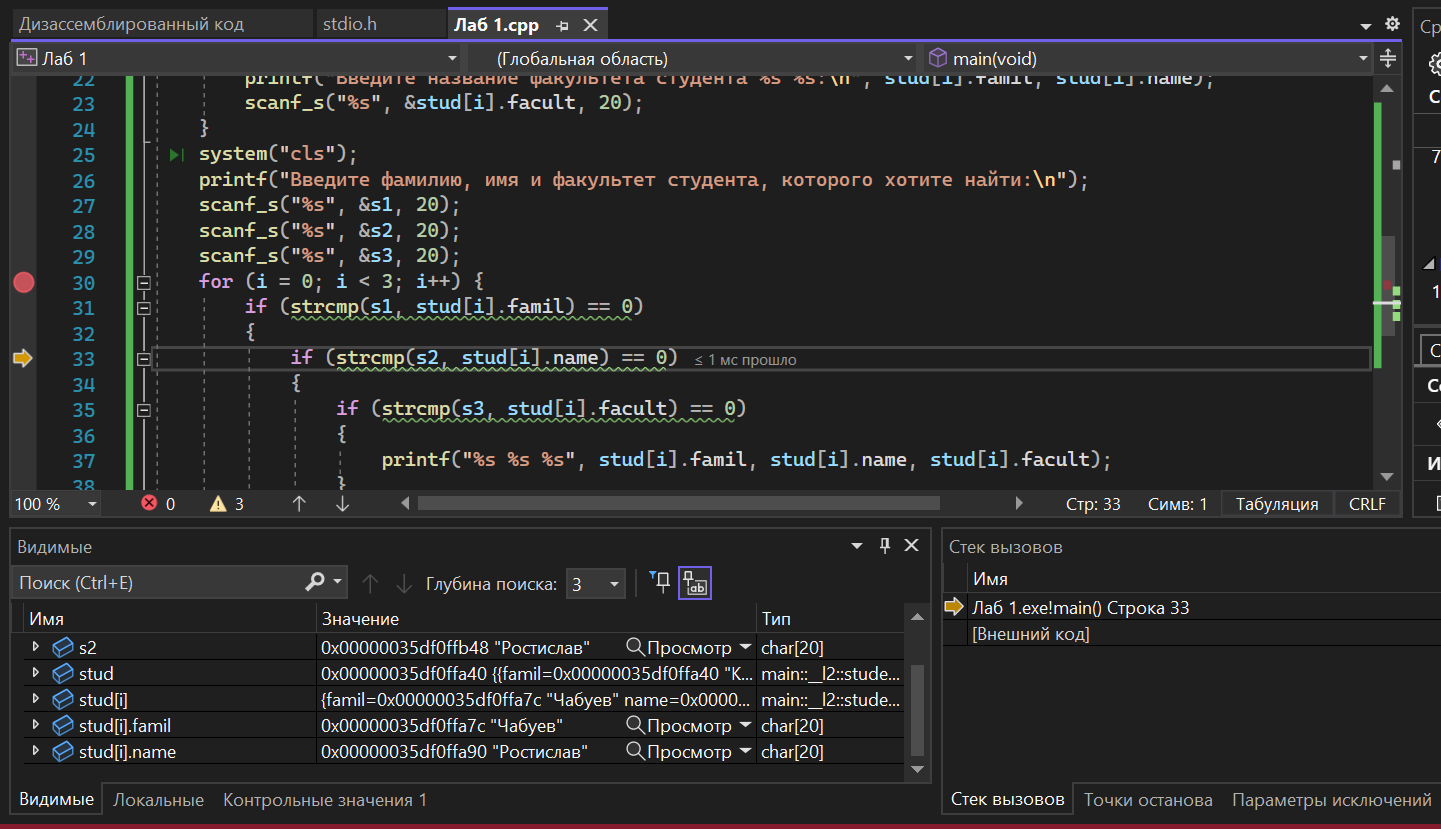


Рис. 5

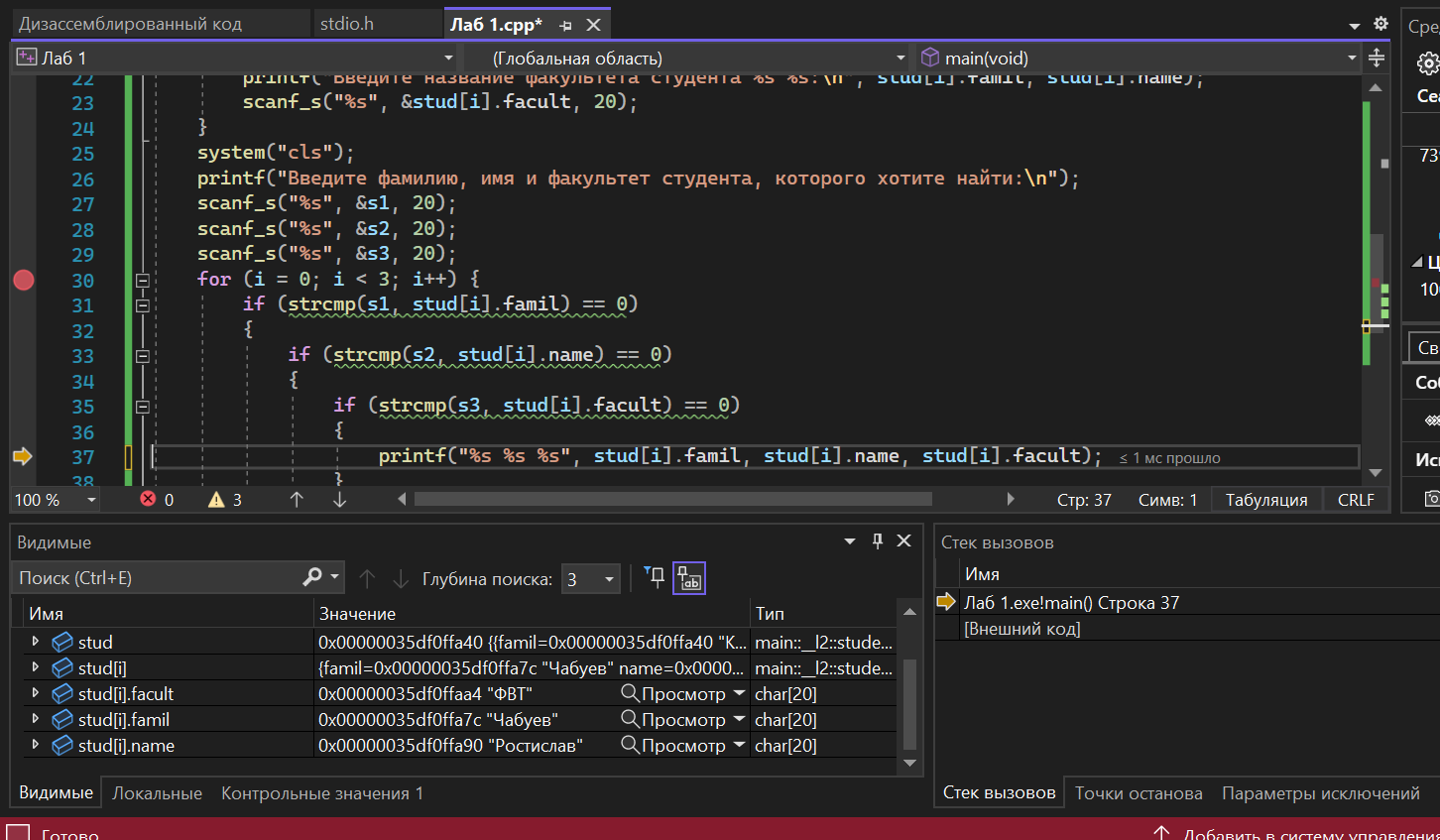


Рис. 6

Результат получился правильный, все действия совершены верно.

Результат работы программы, показанный на рисунке 1-2, совпал с результатами трассировки на рисунке 3-5.

**Вывод:** мы выполнили лабораторные указания 1-5 используя простые структуры данных, где повторили приобретенные навыки с предыдущего курса.