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”



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 <iostream>

#include <time.h>

#include <locale>

using namespace std;

int\* func(int size) {

int\* arr = new int(size);

return arr;

}

struct Student {

int age;

string name;

};

bool Find(Student\* a, int age, string name, int size) {

for (int i = 0; i != size; i++) {

if (a[i].age == age && a[i].name == name) return 1;

}

return 0;

}

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

// задание 1

int arr[10];

for (int i = 0; i != 10; i++) { //задание 2

arr[i] = rand() % 101;

cout << arr[i] << endl;

}

int flag\_min = arr[0];

int flag\_max = arr[0];

for (int i = 1; i != 10; i++) {

if (flag\_min > arr[i]) {

flag\_min = arr[i];

}

if (flag\_max < arr[i]) {

flag\_max = arr[i];

}

}

cout << "Макс элемент = " << flag\_max << endl;

cout << "Мин элемент = " << flag\_min << endl;

return 0;

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1.



Рис. 1

Протокол трассировки программы показан на рисунке 2.

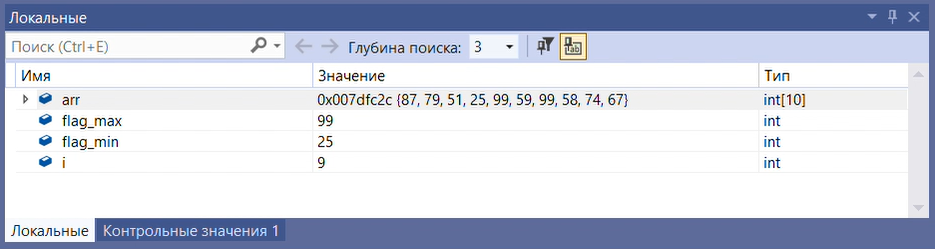


Рис. 2

Результат получился правильный, все действия совершены верно.

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

**Задание 2:**

**Листинг**

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <Windows.h>

int main(void)

{

SetConsoleOutputCP(1251);

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.

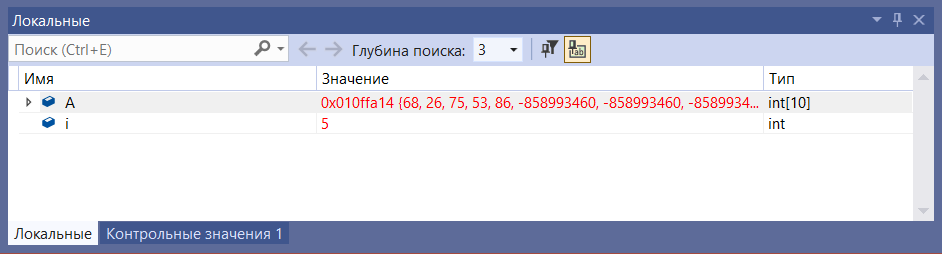


Рис. 2

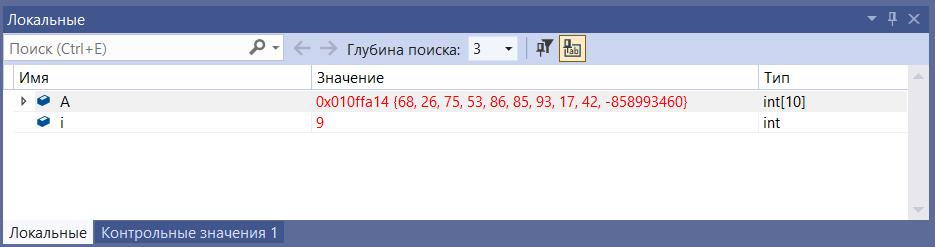


Рис. 3

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);

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.

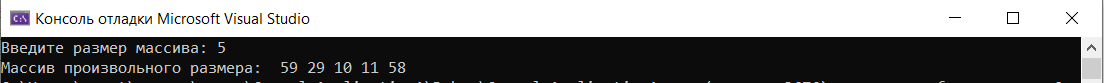


Рис. 1

Протокол трассировки программы показан на рисунке 2 и 3.

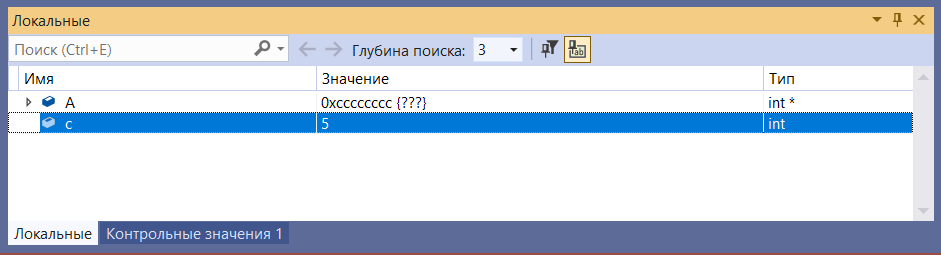


Рис. 2

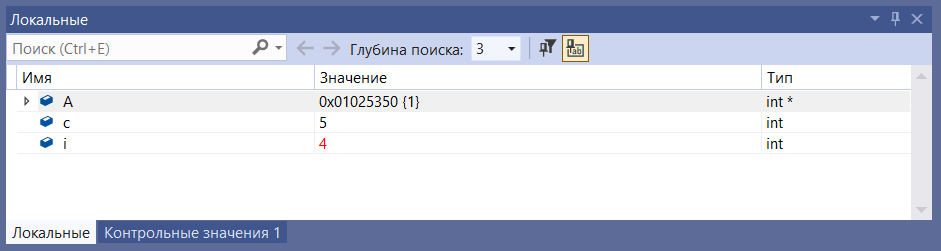


Рис. 3

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 <iostream>

#include <time.h>

#include <locale>

using namespace std;

int\* func(int size) {

int\* arr = new int(size);

return arr;

}

struct Student {

int age;

string name;

};

bool Find(Student\* a, int age, string name, int size) {

for (int i = 0; i != size; i++) {

if (a[i].age == age && a[i].name == name) return 1;

}

return 0;

}

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

int arr2[10][10];

for (int i = 0; i != 10; i++) {

for (int j = 0; j != 10; j++) {

arr2[i][j] = rand() % 101;

cout << arr2[i][j] << endl;

}

cout << endl;

}

int sum\_col = 0;

int sum\_row = 0;

for (int i = 0; i != 10; i++) {

for (int j = 0; j != 10; j++) {

sum\_col += arr2[i][j];

}

cout << "Строка№" << i << " = " << sum\_col << endl;

sum\_col = 0;

}

for (int i = 0; i != 10; i++) {

for (int j = 0; j != 10; j++) {

sum\_row += arr2[j][i];

}

cout << "Столбец №" << i << " = " << sum\_row << endl;

sum\_row = 0;

}

return 0;

}

**Результаты работы программы:**

Результаты работы программы показан на рисунке 1.

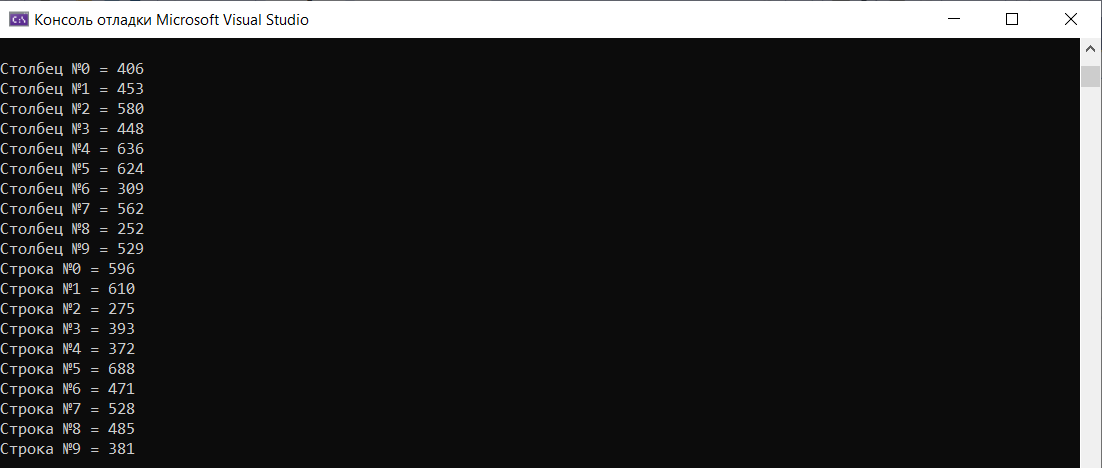


Рис. 1

Протокол трассировки программы показан на рисунке 2 и 3.

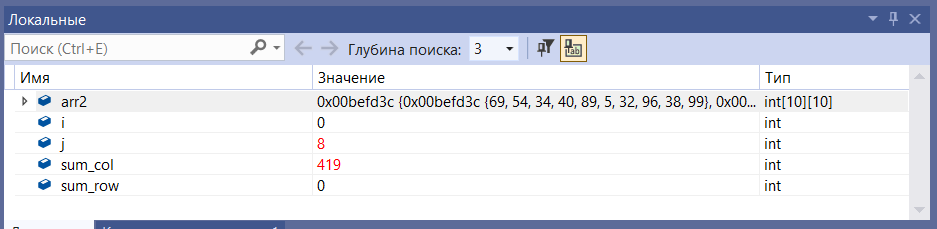


Рис. 2

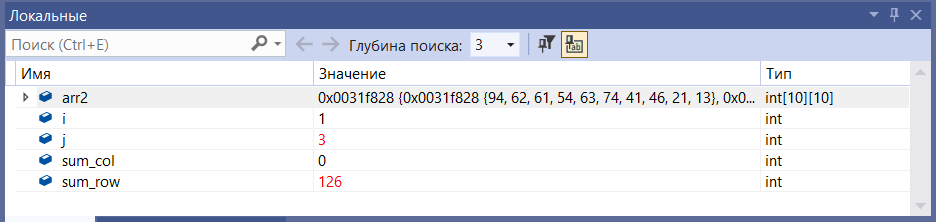


Рис. 3

Результат получился правильный, все действия совершены верно.

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

**Задание 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 имя студента:\n");

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

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

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

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

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

}

system("cls");

printf("Введите имя, фамилию и возраст, 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.

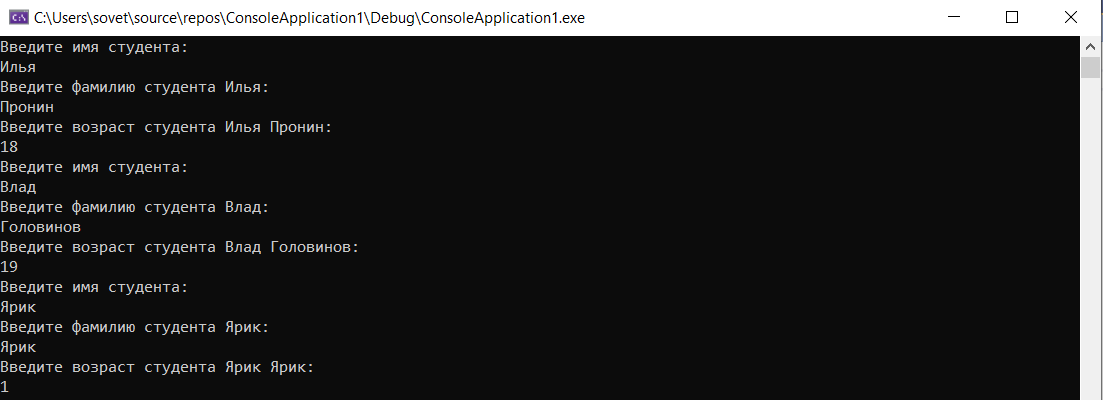


Рис. 1

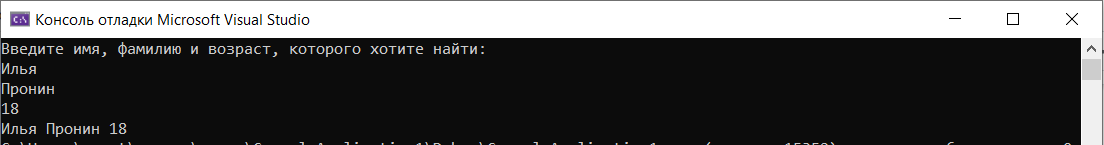


Рис. 2

Протокол трассировки программы показан на рисунке 3-4.

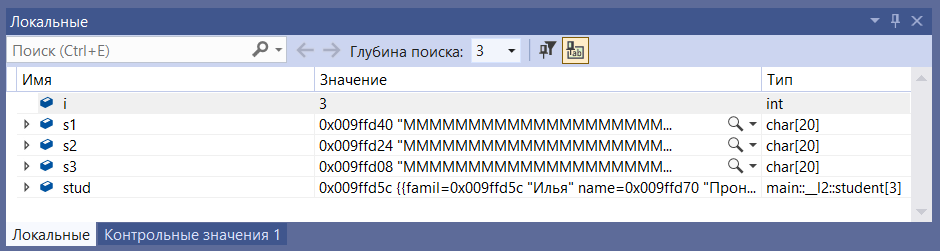


Рис. 3

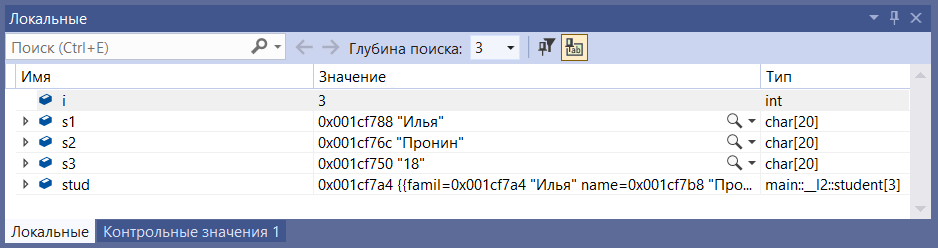


Рис. 4

Результат получился правильный, все действия совершены верно.

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

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