

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

Отчёт

по лабораторной работе №1
по курсу «Л и О А в ИЗ»

на тему «Простые структуры данных»

Выполнили студенты группы 24ВВВ4:

Кондратьев С.В.

Кошелев Р.Д.

Приняли к.т.н., доцент:

Юрова О.В.

доцент:

Акифьев И.В.

Пенза 2025

Цель работы: вспомнить приобретенные на 1 курсе навыки программирования простых структур 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.).

Листинг

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <windows.h>

//задание 5
struct student {
    char famil[50];
    char name[50];
    char facult[50];
    int Nomzach;
};

int main(void) {
```

```
setlocale(LC_ALL, "rus");
```

```
//задание 1
```

```
/*
```

```
int arr[] = { 2, 5, 7, 10, -34, 72, 0 };
```

```
int size = sizeof(arr) / sizeof(arr[0]);
```

```
int max = arr[0];
```

```
int min = arr[0];
```

```
for (int i = 1; i < size; i++) {
```

```
    if (arr[i] < min) {
```

```
        min = arr[i];
```

```
    }
```

```
    if (arr[i] > max) {
```

```
        max = arr[i];
```

```
    }
```

```
}
```

```
printf("Массив: ");
```

```
for (int i = 0; i < size; i++) {
```

```
    printf("%d ", arr[i]);
```

```
}
```

```
printf("\nМинимальный элемент: %d", min);
```

```
printf("\nМаксимальный элемент: %d", max);
```

```
printf("\nРазница: %d - %d = %d\n", max, min, max - min);
```

```
*/
```

```
//задание 2
```

```
/*
```

```
int arr[10];
```

```
srand(time(NULL));
```

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

```

        arr[i] = rand() % 101 - 50;
    }
    printf("Массив ");
    for (int i = 0; i < 10; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
*/

```

//задание 3

```

/*
int size;
printf("Введите размер массива: ");
while (scanf("%d", &size) != 1 or size <= 0) {
    while (getchar() != '\n');
    printf("Ошибка! Введите целое число больше 0: ");
}
if (size > 1000000) {
    printf("Слишком большой размер! Максимум: 1000000\n");
    return 1;
}
int* arr = (int*)malloc(size * sizeof(int));
if (arr == NULL) {
    printf("Ошибка выделения памяти!\n");
    return 1;
}
printf("Введите %d элементов массива:\n", size);
for (int i = 0; i < size; i++) {
    printf("Элемент %d: ", i + 1);
    while (scanf("%d", &arr[i]) != 1) {

```

```

        while (getchar() != '\n');
        printf("Ошибка! Введите целое число: ");
        printf("Элемент %d: ", i + 1);
    }
}

printf("\nСозданный массив:\n");
for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);
}

printf("\n");
free(arr);
*/

//задание 4
/*

int rows, cols;

printf("Введите количество строк: ");
while (scanf("%d", &rows) != 1 or rows <= 0) {
    while (getchar() != '\n');
    printf("Ошибка! Введите целое число больше 0: ");
}

printf("Введите количество столбцов: ");
while (scanf("%d", &cols) != 1 or cols <= 0) {
    while (getchar() != '\n');
    printf("Ошибка! Введите целое число больше 0: ");
}

int** arr = (int**)malloc(rows * sizeof(int*));
for (int i = 0; i < rows; i++) {
    arr[i] = (int*)malloc(cols * sizeof(int));
}

```

```

srand(time(NULL));
printf("\nСгенерированный массив %dx%d:\n", rows, cols);
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        arr[i][j] = rand() % 100;
        printf("%4d", arr[i][j]);
    }
    printf("\n");
}
printf("\nСуммы по столбцам:\n");
for (int i = 0; i < rows; i++) {
    int sum_row = 0;
    for (int j = 0; j < cols; j++) {
        sum_row += arr[i][j];
    }
    printf("Строка %d: %d\n", i + 1, sum_row);
}
printf("\nСуммы по строкам:\n");
for (int j = 0; j < cols; j++) {
    int sum_col = 0;
    for (int i = 0; i < rows; i++) {
        sum_col += arr[i][j];
    }
    printf("Строка %d: %d\n", j + 1, sum_col);
}
free(arr);
*/

//задание 5
/*

```

```
SetConsoleOutputCP(1251);
SetConsoleCP(1251);
setvbuf(stdin, NULL, _IONBF, 0);
setvbuf(stdout, NULL, _IONBF, 0);
int i;
struct student stud[3];
char search_famil[50];
printf("Введите данные 3 студентов:\n");
for (i = 0; i < 3; i++)
{
    printf("\nСтудент %d:\n", i + 1);
    printf("Фамилия: ");
    scanf("%49s", stud[i].famil);
    while (getchar() != '\n');
    printf("Имя: ");
    scanf("%49s", stud[i].name);
    while (getchar() != '\n');
    printf("Факультет: ");
    scanf("%49[^\n]", stud[i].facult);
    while (getchar() != '\n');
    printf("Номер зачетки: ");
    scanf("%d", &stud[i].Nomzach);
    while (getchar() != '\n');
}
printf("\nВведите фамилию для поиска: ");
scanf("%49s", search_famil);
int found = 0;
printf("\nРезультаты поиска:\n");
for(i = 0; i < 3; i++)
{
```

```

        if(strcmp(stud[i].famil, search_famil) == 0)
        {
            printf("Найден: %s %s, %s, зачетка №%d\n",
                stud[i].famil, stud[i].name, stud[i].facult, stud[i].Nomzach);
            found = 1;
        }
    }
    if(!found)
    {
        printf("Студент с фамилией '%s' не найден\n", search_famil);
    }

    /*
    return 0;
}

```

Схема программы

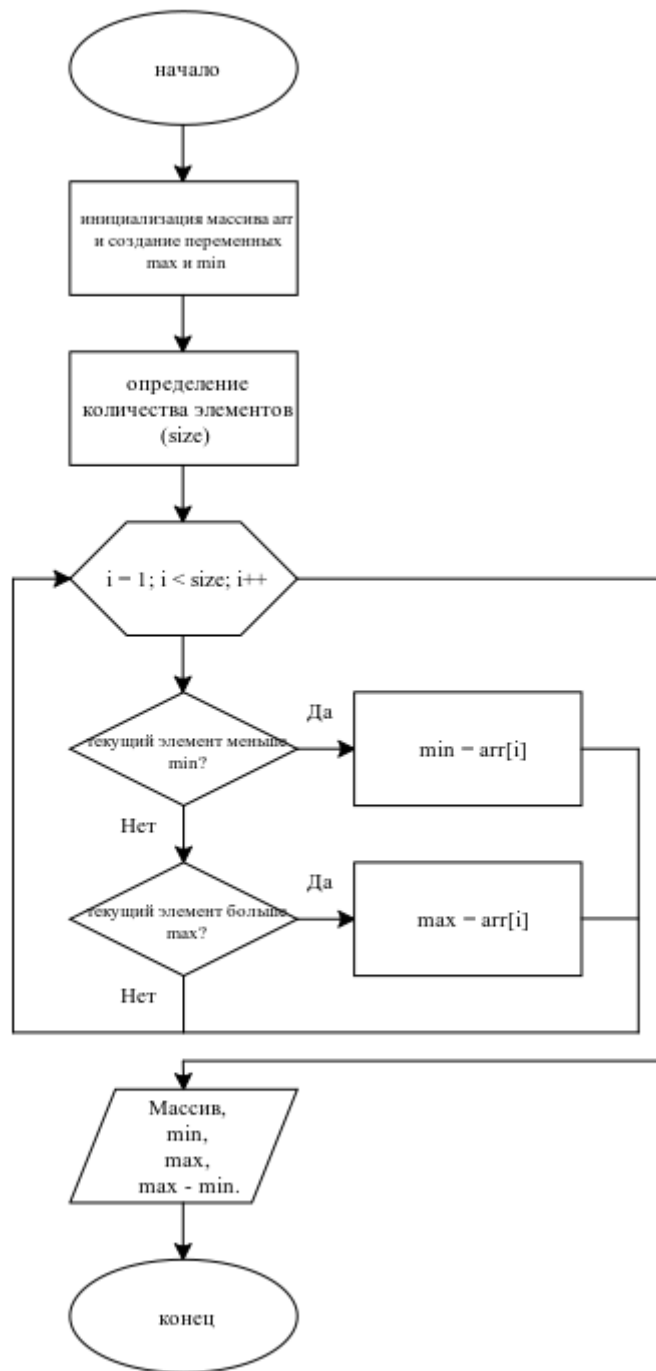


Рисунок 1 – Схема программы 1



Рисунок 2 – Схема программы 2

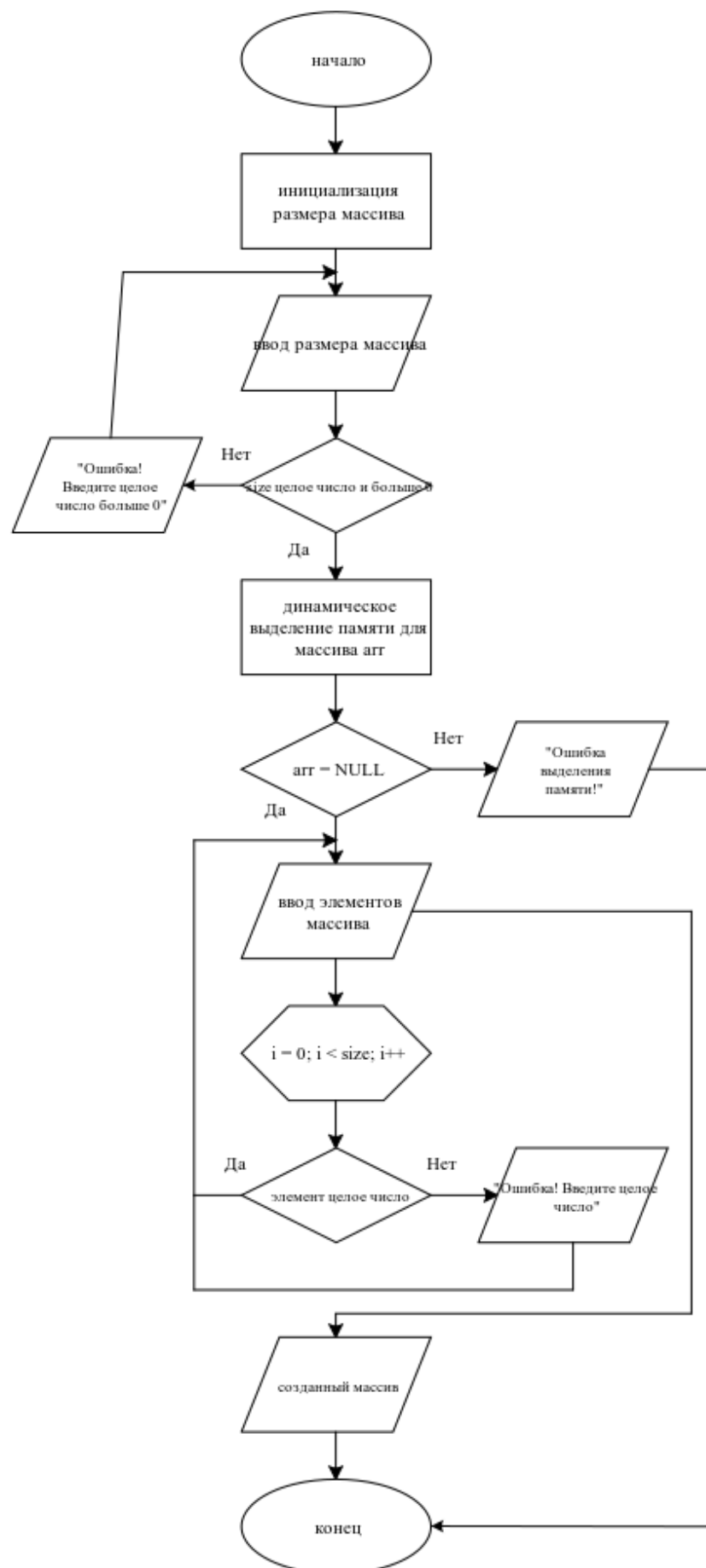
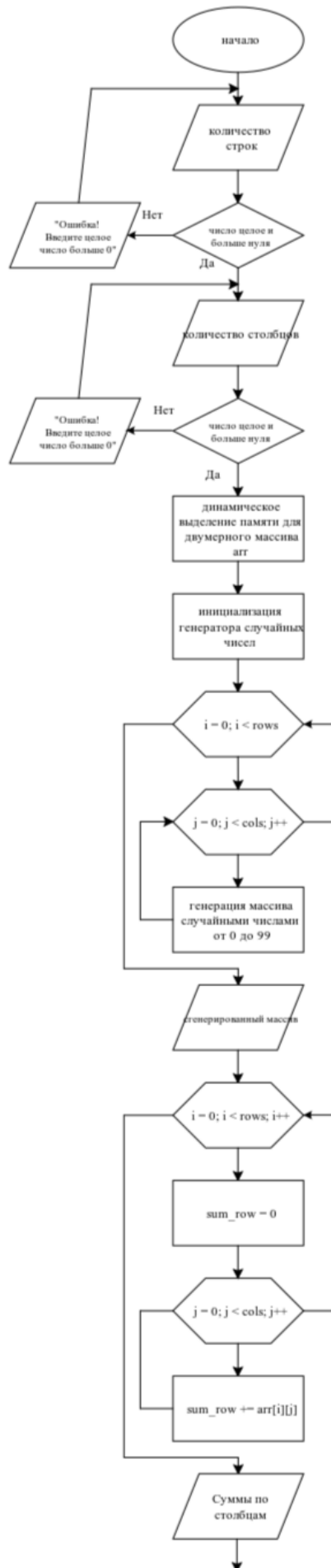


Рисунок 3 – Схема программы 3



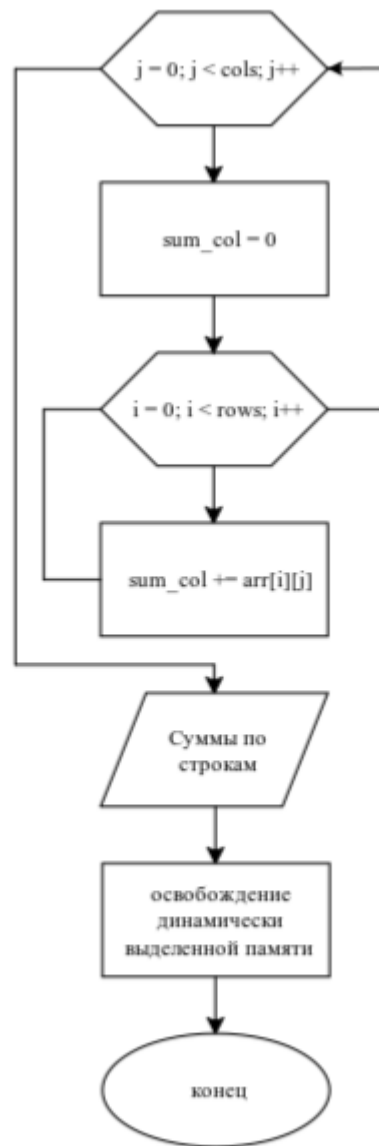


Рисунок 4 – Схема программы 4

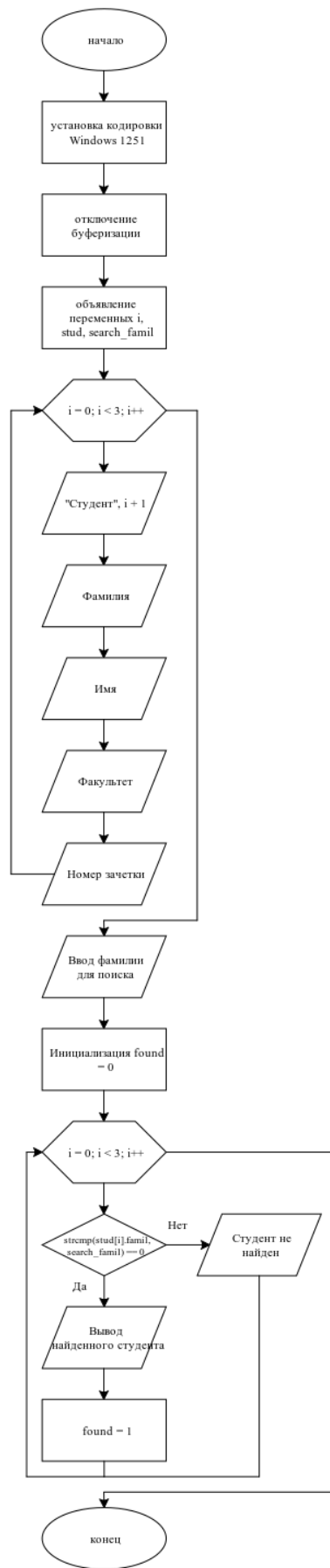
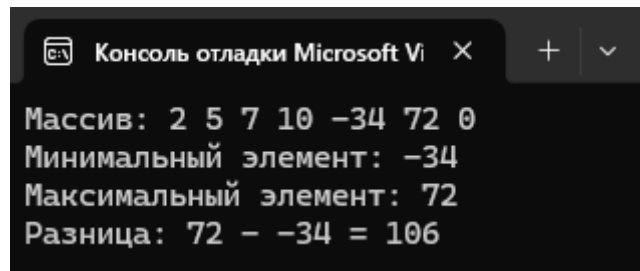


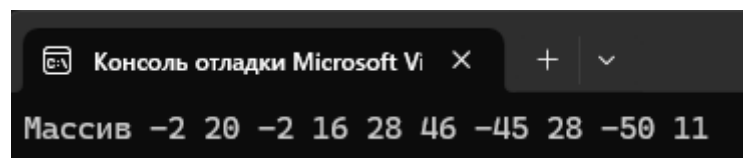
Рисунок 5 – Схема программы 5

Результаты работ программ



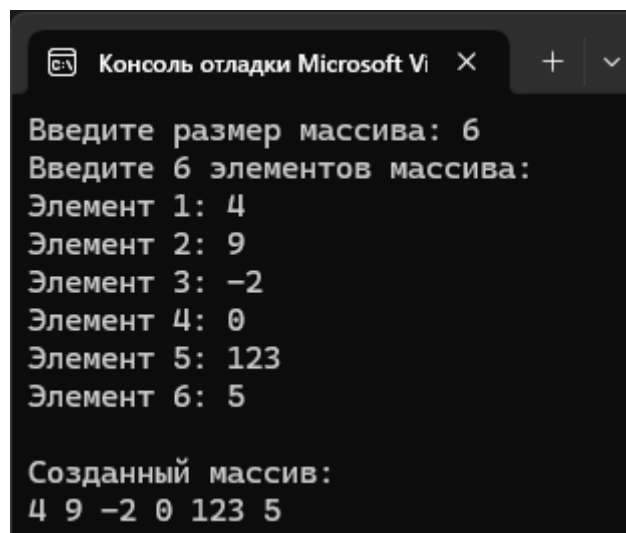
```
Консоль отладки Microsoft Vi X + v
Массив: 2 5 7 10 -34 72 0
Минимальный элемент: -34
Максимальный элемент: 72
Разница: 72 - -34 = 106
```

Рисунок 6 – Результат работы программы 1



```
Консоль отладки Microsoft Vi X + v
Массив -2 20 -2 16 28 46 -45 28 -50 11
```

Рисунок 7 – Результат работы программы 2



```
Консоль отладки Microsoft Vi X + v
Введите размер массива: 6
Введите 6 элементов массива:
Элемент 1: 4
Элемент 2: 9
Элемент 3: -2
Элемент 4: 0
Элемент 5: 123
Элемент 6: 5

Созданный массив:
4 9 -2 0 123 5
```

Рисунок 8 – Результат работы программы 3

```
Консоль отладки Microsoft Vi X + v
Введите количество строк: 5
Введите количество столбцов: 8

Сгенерированный массив 5x8:
16 26 51 23 84 38 41 86
82 72 6 28 91 55 52 12
93 15 14 49 67 44 9 11
56 15 1 71 61 42 18 15
72 23 24 13 48 32 9 98

Суммы по столбцам:
Строка 1: 365
Строка 2: 398
Строка 3: 302
Строка 4: 279
Строка 5: 319

Суммы по строкам:
Строка 1: 319
Строка 2: 151
Строка 3: 96
Строка 4: 184
Строка 5: 351
Строка 6: 211
Строка 7: 129
Строка 8: 222
```

Рисунок 9 – Результат работы программы 4

```
Консоль отладки Microsoft Vi X + v
Введите данные 3 студентов:

Студент 1:
Фамилия: Кошелев
Имя: Руслан
Факультет: Вычислительная техника
Номер зачетки: 4543534

Студент 2:
Фамилия: Кондратьев
Имя: Сергей
Факультет: Вычислительная техника
Номер зачетки: 233455

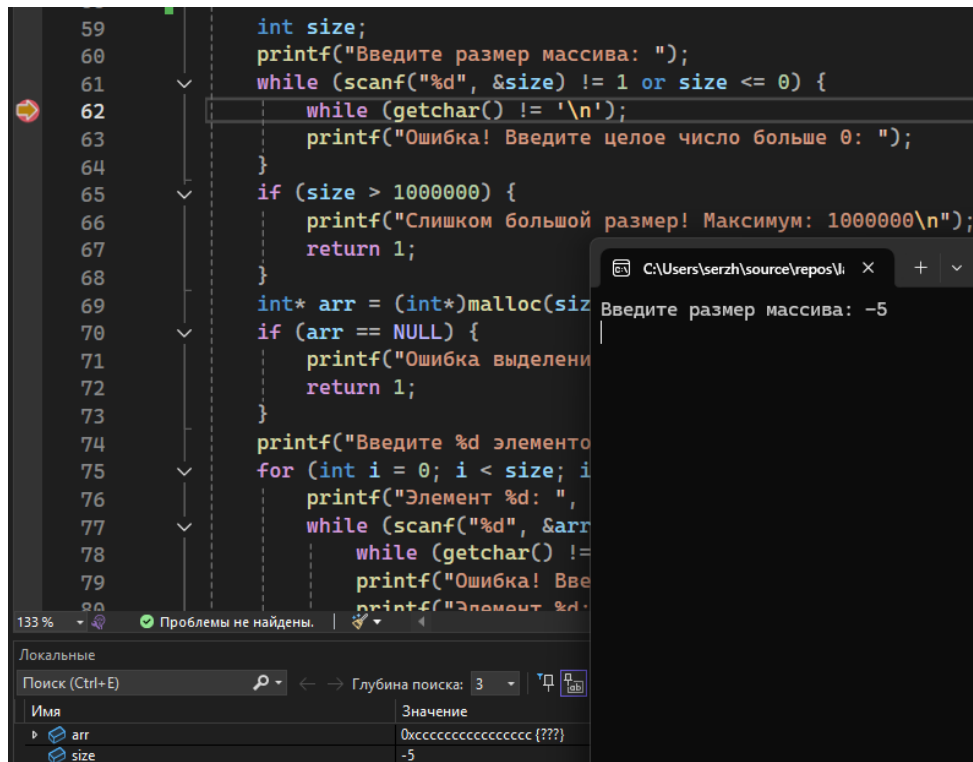
Студент 3:
Фамилия: Подвохов
Имя: Павел
Факультет: Вычислительная техника
Номер зачетки: 3454356

Введите фамилию для поиска: Кондратьев

Результаты поиска:
Найден: Кондратьев Сергей, Вычислительная техника, зачетка %233455
```

Рисунок 10 – Результат работы программы 5

Протокол трассировки программы



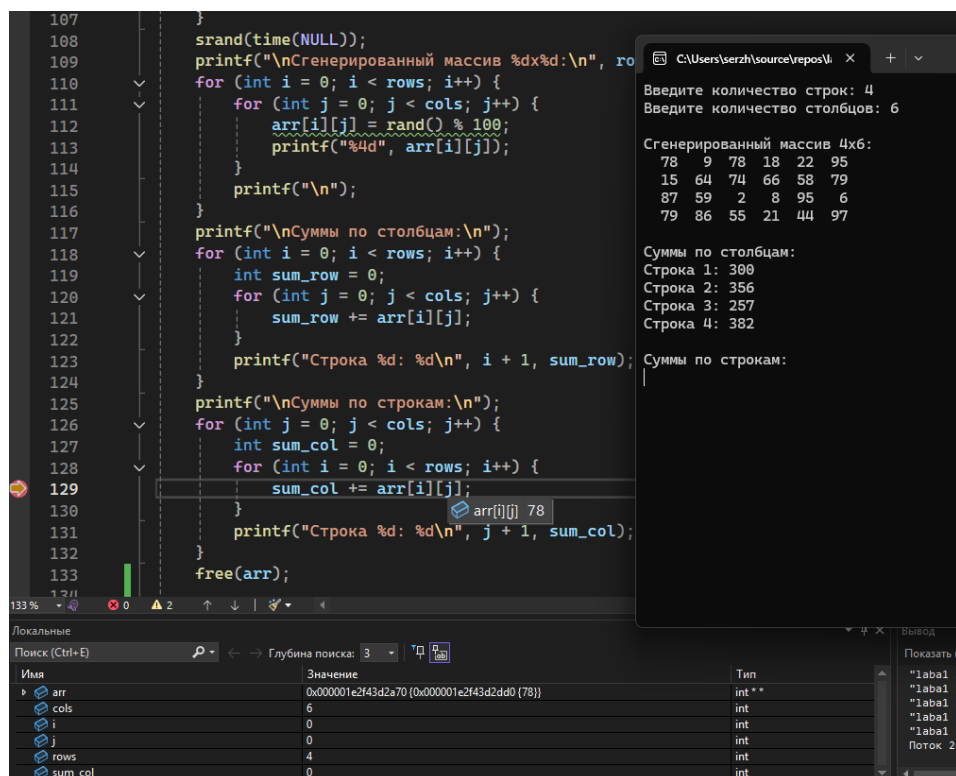
```
59     int size;
60     printf("Введите размер массива: ");
61     while (scanf("%d", &size) != 1 or size <= 0) {
62         while (getchar() != '\n');
63         printf("Ошибка! Введите целое число больше 0: ");
64     }
65     if (size > 1000000) {
66         printf("Слишком большой размер! Максимум: 1000000\n");
67         return 1;
68     }
69     int* arr = (int*)malloc(size * sizeof(int));
70     if (arr == NULL) {
71         printf("Ошибка выделения памяти\n");
72         return 1;
73     }
74     printf("Введите %d элементов\n", size);
75     for (int i = 0; i < size; i++) {
76         printf("Элемент %d: ", i);
77         while (scanf("%d", &arr[i]) != 1) {
78             while (getchar() != '\n');
79             printf("Ошибка! Введите целое число\n");
80         }
81     }
82     printf("Элементы массива: ");
83     for (int i = 0; i < size; i++) {
84         printf("%d ", arr[i]);
85     }
86     printf("\n");
87     free(arr);
88     return 0;
89 }
```

Локальные

Имя	Значение
arr	0хcccccccccccccccc {???
size	-5

Поиск (Ctrl+E) Глубина поиска: 3 Проблемы не найдены.

Рисунок 11 – Результат трассировки программы 3



```
107     srand(time(NULL));
108     printf("Сгенерированный массив %dx%d:\n", rows, cols);
109     for (int i = 0; i < rows; i++) {
110         for (int j = 0; j < cols; j++) {
111             arr[i][j] = rand() % 100;
112             printf("%4d", arr[i][j]);
113         }
114     }
115     printf("\n");
116
117     printf("\nСуммы по столбцам:\n");
118     for (int i = 0; i < cols; i++) {
119         int sum_col = 0;
120         for (int j = 0; j < rows; j++) {
121             sum_col += arr[j][i];
122         }
123         printf("Столбец %d: %d\n", i + 1, sum_col);
124     }
125
126     printf("\nСуммы по строкам:\n");
127     for (int j = 0; j < rows; j++) {
128         int sum_row = 0;
129         for (int i = 0; i < cols; i++) {
130             sum_row += arr[j][i];
131         }
132         printf("Строка %d: %d\n", j + 1, sum_row);
133     }
134     free(arr);
135     return 0;
136 }
```

Локальные

Имя	Значение	Тип
arr	0х000001e2f43d2a70 (0х000001e2f43d2dd0 [78])	int **
cols	6	int
i	0	int
j	0	int
rows	4	int
sum_col	0	int

Поиск (Ctrl+E) Глубина поиска: 3 Проблемы не найдены.

Вывод

Введите количество строк: 4
Введите количество столбцов: 6

Сгенерированный массив 4x6:

78	9	78	18	22	95
15	64	74	66	58	79
87	59	2	8	95	6
79	86	55	21	44	97

Суммы по столбцам:

Столбец 1:	300
Столбец 2:	356
Столбец 3:	257
Столбец 4:	382

Суммы по строкам:

Строка 1:	300
Строка 2:	356
Строка 3:	257
Строка 4:	382

Рисунок 11 – Результат трассировки программы 4

Вывод: вспомнили навыки программирования простых структур данных, приобретенных на 1 курсе.