Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

по лабораторной работе №1

по курсу “Логика и основы алгоритмизации в инженерных задачах”

на тему “простые структуры данных”

Выполнили:

Студенты группы 24ВВВ4

Васин И.В.

Анисимов А.Ю.

Приняли:

Юрова О.В.

Акифьев И.В.

Пенза 2025

**Название**

Простые структуры данных

**Цель работы**

Вспомнить материал за прошлый курс и разработать программу, работая с массивами и структурами

**Лабораторное задание**

1. написать программу, вычисляющую разницу между максимальным и минимальным элементами массива
2. написать программу, реализующую инициализацию массива случайными числами
3. написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры
4. написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива
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 <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <locale.h>

#include <wchar.h>

#define MAX\_STUDENTS 100

#define MAX\_NAME\_LENGTH 50

#define MAX\_ROWS 10

#define MAX\_COLS 10

typedef struct {

char surname[MAX\_NAME\_LENGTH];

char name[MAX\_NAME\_LENGTH];

int age;

float average\_grade;

} Student;

void array\_range\_difference();

void random\_array\_init();

void custom\_size\_array();

void matrix\_row\_column\_sum();

void search\_student();

void print\_array(int arr[], int size);

int get\_int\_input(const char\* prompt, int min\_val, int max\_val);

float get\_float\_input(const char\* prompt, float min\_val, float max\_val);

int main() {

setlocale(LC\_ALL, "");

int choice;

srand(time(NULL));

do {

printf("\n=== МЕНЮ ПРОГРАММЫ ===\n");

printf("1. Разница max и min\n");

printf("2. Массив, случайные числа\n");

printf("3. Динамичный массив\n");

printf("4. Сумма значений строки/столбцы\n");

printf("5. Ввод данных студентов\n");

printf("0. Выход\n");

printf("Выберите пункт меню: ");

if (scanf("%d", &choice) != 1) {

printf("Ошибкаввода! Пожалуйста, введите число.\n");

while (getchar() != '\n');

continue;

}

switch (choice) {

case 1:

array\_range\_difference();

break;

case 2:

random\_array\_init();

break;

case 3:

custom\_size\_array();

break;

case 4:

matrix\_row\_column\_sum();

break;

case 5:

search\_student();

break;

case 0:

printf("Выход из программы...\n");

break;

default:

printf("Неверный выбор! Попробуйте снова.\n");

}

while (getchar() != '\n');

} while (choice != 0);

return 0;

}

int get\_int\_input(const char\* prompt, int min\_val, int max\_val) {

int value;

int valid\_input;

do {

printf("%s", prompt);

valid\_input = scanf("%d", &value);

if (valid\_input != 1) {

printf("Ошибкаввода! Пожалуйста, введите целое число.\n");

while (getchar() != '\n');

continue;

}

if (value <min\_val || value >max\_val) {

printf("Ошибка! Значение должно быть от %d до %d.\n", min\_val, max\_val);

valid\_input = 0;

}

} while (valid\_input != 1 || value <min\_val || value >max\_val);

return value;

}

float get\_float\_input(const char\* prompt, float min\_val, float max\_val) {

float value;

int valid\_input;

do {

printf("%s", prompt);

valid\_input = scanf("%f", &value);

if (valid\_input != 1) {

printf("Ошибкаввода! Пожалуйста, введите число.\n");

while (getchar() != '\n');

continue;

}

if (value <min\_val || value >max\_val) {

printf("Ошибка! Значение должно быть от %.1f до %.1f.\n", min\_val, max\_val);

valid\_input = 0;

}

} while (valid\_input != 1 || value <min\_val || value >max\_val);

return value;

}

void array\_range\_difference() {

printf("\n=== Разница между max и min элементами массива ===\n");

int size = get\_int\_input("Введитеразмермассива (1-100): ", 1, 100);

int\* arr = (int\*)malloc(size \* sizeof(int));

if (arr == NULL) {

printf("Ошибкавыделенияпамяти!\n");

return;

}

printf("Введите %d элементовмассива:\n", size);

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

printf("Элемент %d: ", i + 1);

while (scanf("%d", &arr[i]) != 1) {

printf("Ошибка ввода! Пожалуйста, введите целое число: ");

while (getchar() != '\n');

}

}

int min = arr[0], max = 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: ");

print\_array(arr, size);

printf("Минимальный элемент: %d\n", min);

printf("Максимальный элемент: %d\n", max);

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

free(arr);

}

void random\_array\_init() {

printf("\n=== Инициализация массива случайными числами ===\n");

int size = get\_int\_input("Введитеразмермассива (1-100): ", 1, 100);

intmin\_val = get\_int\_input("Введите минимальное значение: ", -1000, 1000);

intmax\_val = get\_int\_input("Введите максимальное значение: ", min\_val, 1000);

int\* arr = (int\*)malloc(size \* sizeof(int));

if (arr == NULL) {

printf("Ошибкавыделенияпамяти!\n");

return;

}

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

arr[i] = rand() % (max\_val - min\_val + 1) + min\_val;

}

printf("Сгенерированныймассив: ");

print\_array(arr, size);

free(arr);

}

void custom\_size\_array() {

printf("\n=== Создание массива произвольного размерa ===\n");

int size = get\_int\_input("Введитеразмермассива (1-100): ", 1, 100);

int\* arr = (int\*)malloc(size \* sizeof(int));

if (arr == NULL) {

printf("Ошибкавыделенияпамяти!\n");

return;

}

printf("Введите %d элементовмассива:\n", size);

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

printf("Элемент %d: ", i + 1);

while (scanf("%d", &arr[i]) != 1) {

printf("Ошибка ввода! Пожалуйста, введите целое число: ");

while (getchar() != '\n');

}

}

printf("Созданный массив: ");

print\_array(arr, size);

free(arr);

}

void matrix\_row\_column\_sum() {

printf("\n=== Сумма значений в строках и столбцах матрицы ===\n");

introws = get\_int\_input("Введите количество строк (1-10): ", 1, MAX\_ROWS);

intcols = get\_int\_input("Введите количество столбцов (1-10): ", 1, MAX\_COLS);

intmin\_val = get\_int\_input("Введите минимальное значение для случайных чисел: ", -1000, 1000);

intmax\_val;

do {

max\_val = get\_int\_input("Введите максимальное значение для случайных чисел: ", min\_val, 1000);

if (max\_val<min\_val) {

printf("Максимальное значение не может быть меньше минимального!\n");

}

} while (max\_val<min\_val);

int matrix[MAX\_ROWS][MAX\_COLS];

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

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

matrix[i][j] = rand() % (max\_val - min\_val + 1) + min\_val;

}

}

printf("\nСгенерированнаяматрица %dx%d:\n", rows, cols);

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

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

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

}

printf("\n");

}

printf("\nСуммыстрок:\n");

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

int row\_sum = 0;

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

row\_sum += matrix[i][j];

}

printf("Строка %d: %d\n", i + 1, row\_sum);

}

printf("\nСуммыстолбцов:\n");

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

int col\_sum = 0;

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

col\_sum += matrix[i][j];

}

printf("Столбец %d: %d\n", j + 1, col\_sum);

}

}

void search\_student() {

printf("\n=== Вводданныхстудентов и поиск ===\n");

Student students[MAX\_STUDENTS];

int count;

count = get\_int\_input("Введите количество студентов (1-100): ", 1, MAX\_STUDENTS);

printf("Введите данные %d студентов (используя английские символы):\n", count);

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

printf("\nСтудент %d:\n", i + 1);

printf("Фамилия: ");

scanf("%s", students[i].surname);

printf("Имя: ");

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

students[i].age = get\_int\_input("Возраст: ", 1, 100);

students[i].average\_grade = get\_float\_input("Среднийбалл: ", 1, 100);

}

printf("\nВведенные данные студентов:\n");

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

printf("%d. %s %s, возраст: %d, средний балл: %.1f\n",

i + 1, students[i].surname, students[i].name,

students[i].age, students[i].average\_grade);

}

int search\_choice;

do {

printf("\n=== ПОИСК СТУДЕНТОВ ===\n");

printf("1 - Поиск по фамилии\n");

printf("2 - Поиск по имени\n");

printf("3 - Поиск по возрасту\n");

printf("4 - Поиск по среднему баллу\n");

printf("0 - Вернуться в главное меню\n");

search\_choice = get\_int\_input("Выберитетиппоиска: ", 0, 4);

int found = 0;

switch (search\_choice) {

case 1: {

char search\_surname[MAX\_NAME\_LENGTH];

printf("Введите фамилию для поиска: ");

scanf("%s", search\_surname);

printf("\nРезультатыпоискапофамилии '%s':\n", search\_surname);

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

if (strcmp(students[i].surname, search\_surname) == 0) {

printf("Найден: %s %s, возраст: %d, средний балл: %.1f\n",

students[i].surname, students[i].name,

students[i].age, students[i].average\_grade);

found = 1;

}

}

break;

}

case 2: {

char search\_name[MAX\_NAME\_LENGTH];

printf("Введите имя для поиска: ");

scanf("%s", search\_name);

printf("\nРезультатыпоискапоимени '%s':\n", search\_name);

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

if (strcmp(students[i].name, search\_name) == 0) {

printf("Найден: %s %s, возраст: %d, средний балл: %.1f\n",

students[i].surname, students[i].name,

students[i].age, students[i].average\_grade);

found = 1;

}

}

break;

}

case 3: {

int search\_age = get\_int\_input("Введитевозрастдляпоиска: ", 1, 100);

printf("\nРезультаты поиска по возрасту %d:\n", search\_age);

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

if (students[i].age == search\_age) {

wprintf(L"Найден: %ls %ls, возраст: %d, средний балл: %.1f\n",

students[i].surname, students[i].name,

students[i].age, students[i].average\_grade);

found = 1;

}

}

break;

}

case 4: {

float search\_grade = get\_float\_input("Введитесреднийбаллдляпоиска: ", 1, 100);

printf("\nРезультаты поиска по среднему баллу %.1f:\n", search\_grade);

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

if (students[i].average\_grade == search\_grade) {

wprintf(L"Найден: %ls %ls, возраст: %d, средний балл: %.1f\n",

students[i].surname, students[i].name,

students[i].age, students[i].average\_grade);

found = 1;

}

}

break;

}

case 0:

printf("Возврат в главноеменю...\n");

break;

}

if (search\_choice>= 1 &&search\_choice<= 4 && !found) {

printf("Студенты с указанными параметрами не найдены.\n");

}

while (getchar() != '\n');

} while (search\_choice != 0);

}

void print\_array(int arr[], int size) {

printf("[");

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

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

if (i< size - 1) printf(", ");

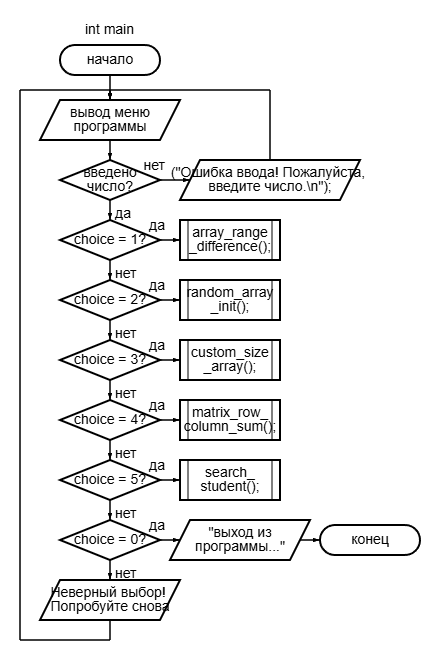
}

printf("]\n");

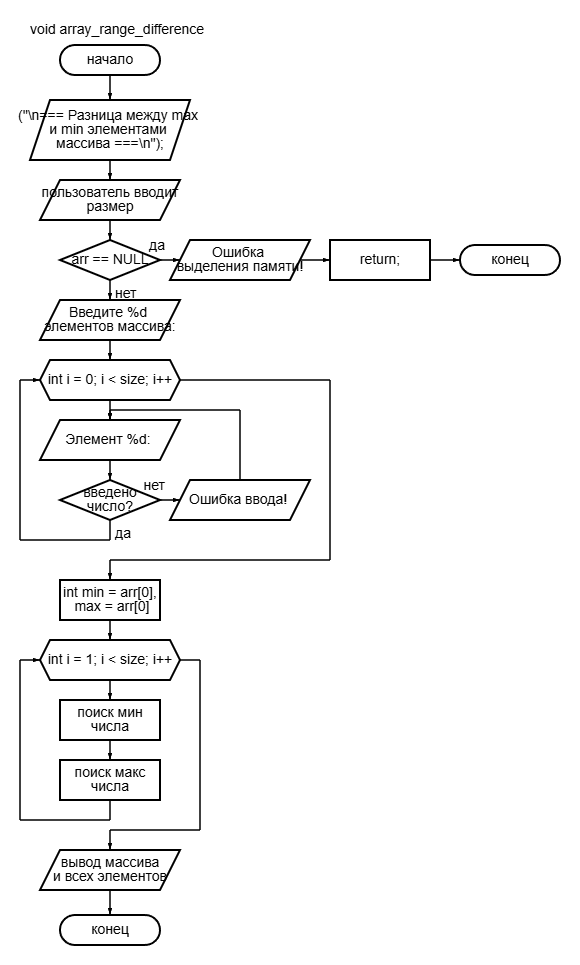
}

**Схема программы**

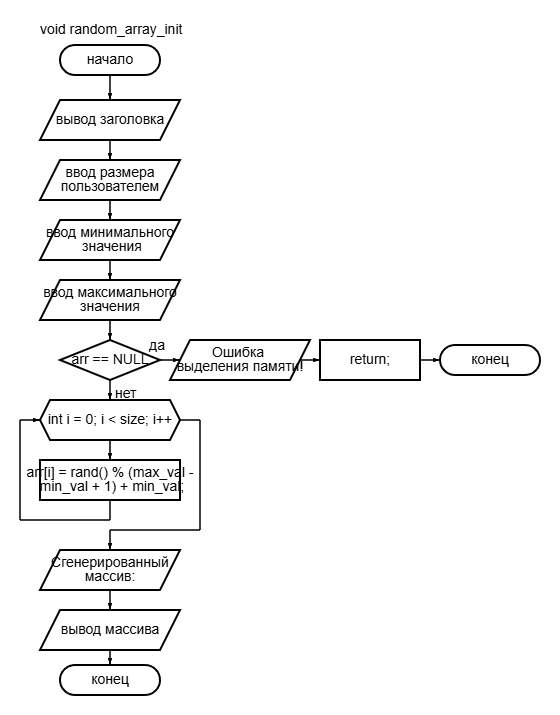
**Int main**



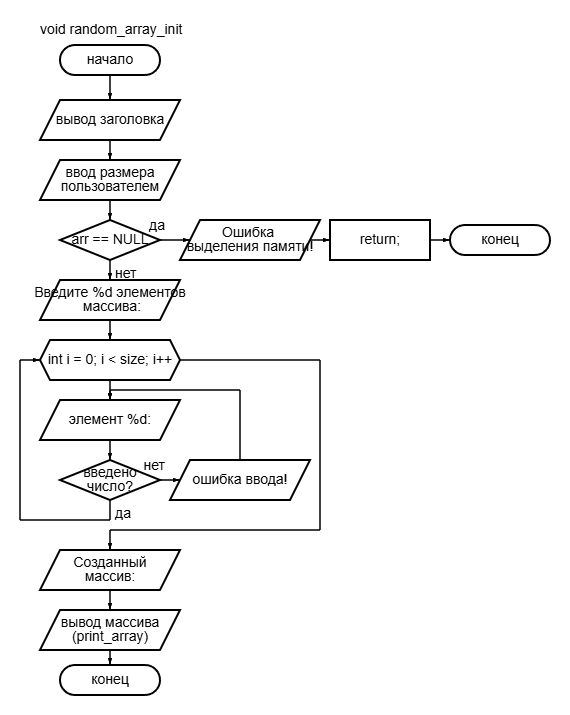
**voidarray\_range\_difference();**

****

**void random\_array\_init();**

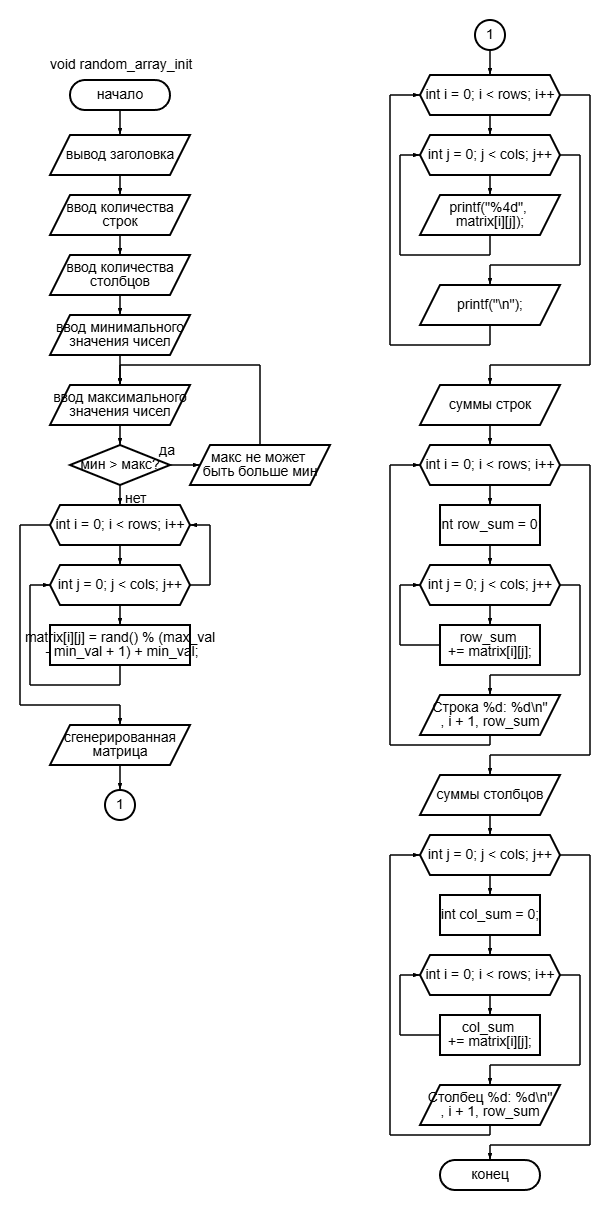
****

**void custom\_size\_array();**

****

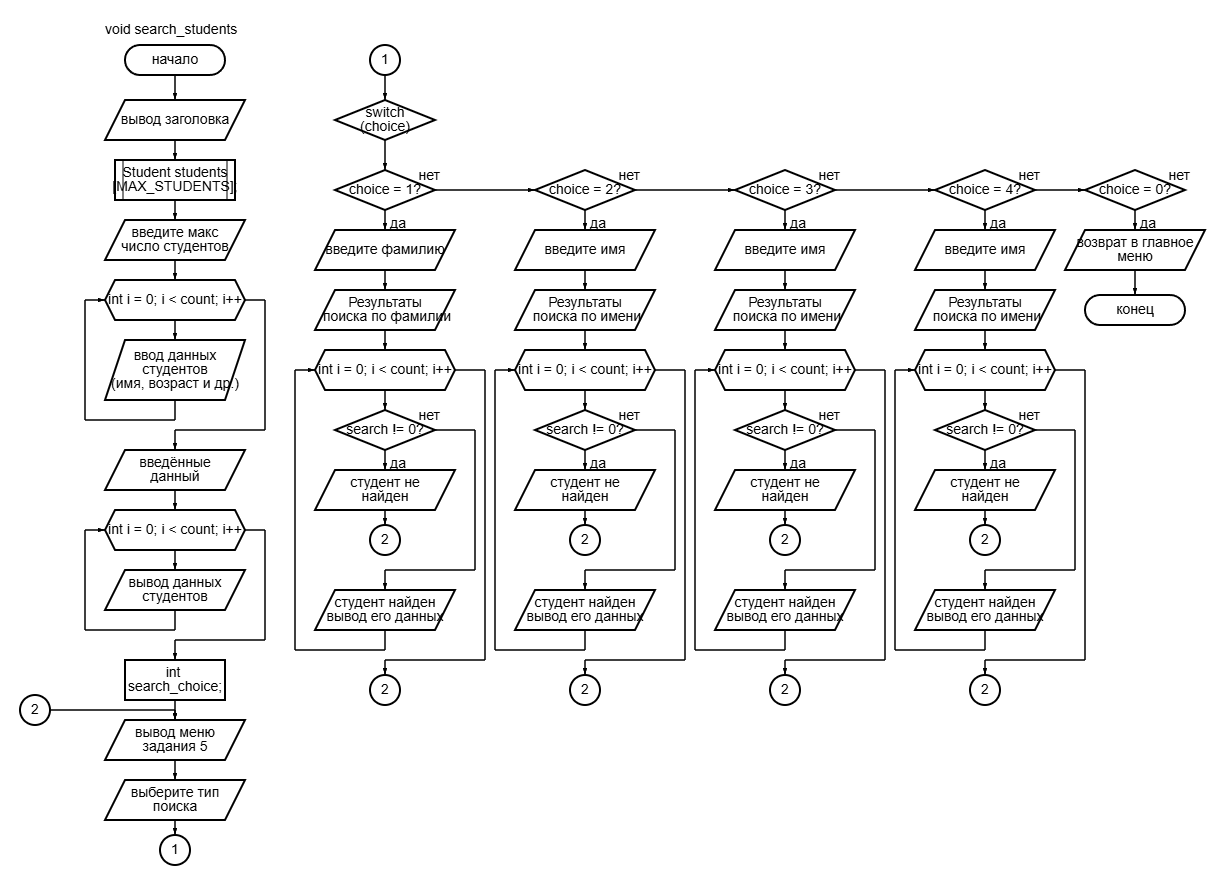
**void matrix\_row\_column\_sum();**



****

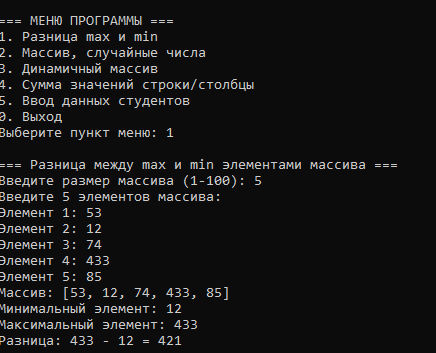


**voidsearch\_student();**

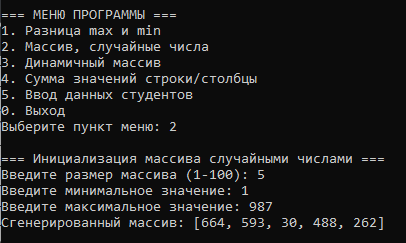
****

**Результат работы программы**

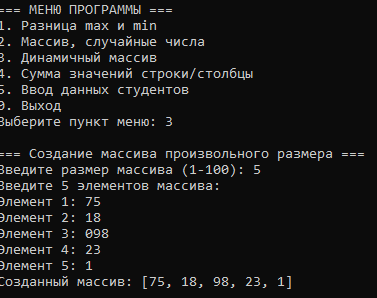
**Задание 1**



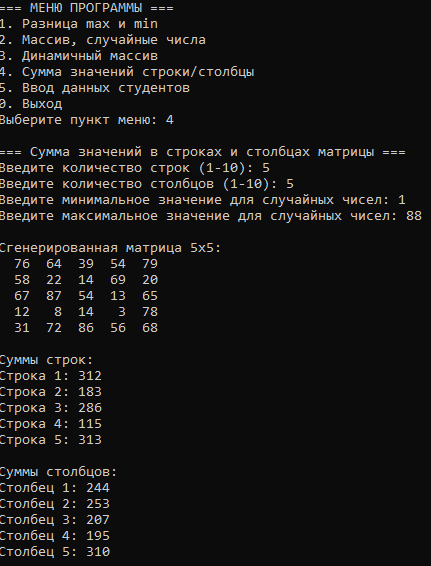
**Задание 2**



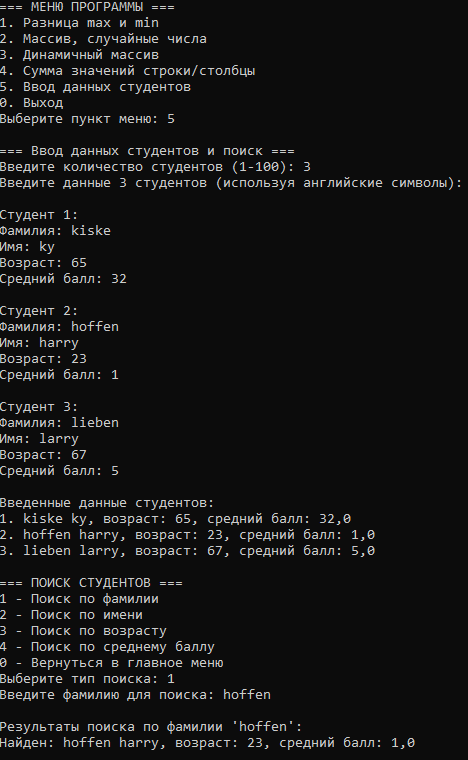
**Задание 3**



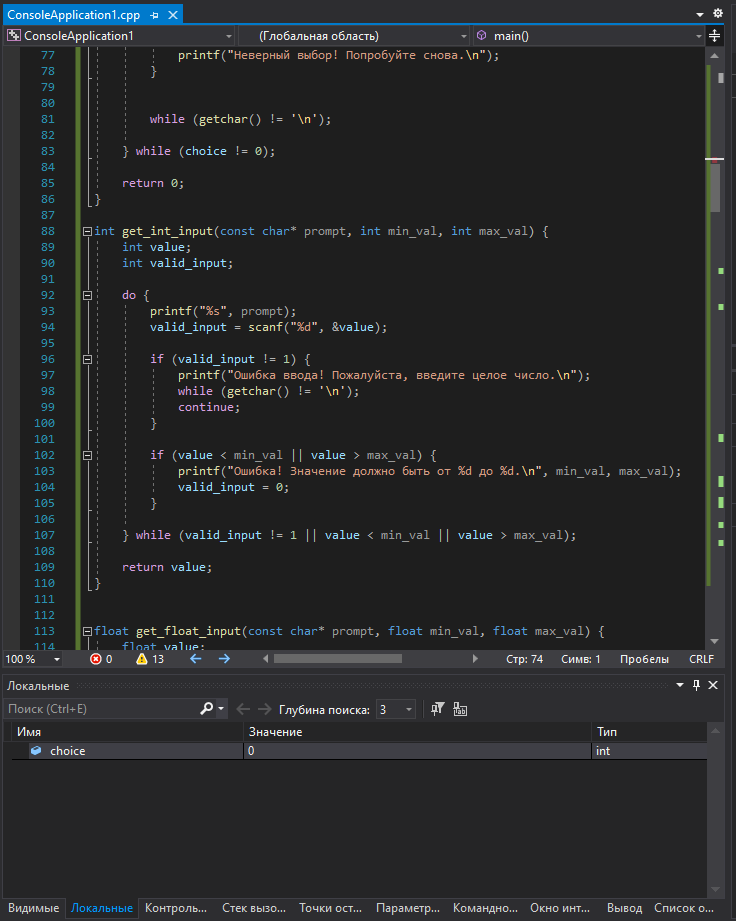
**Задание 4**



**Задание 5**



**Протокол трассировки программы**

****

**Результат работы программы совпал с результатами трассировки**

**Выводы**

В ходе выполнения лабораторной работы были разработаны программы, выполняющие указанные в лабораторной работе задачи. Результаты работ программ совпали с результатами трассировок, следовательно, программы работают без ошибок.

Получили опыт в создании проектов в среде MicrosoftVisualStudio.