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 23ВВВ2:**

Пырков Д. А.

Родионов А. А.

**Приняли:**

Митрохин М.А

Юрова О.В

Пенза 2024

**Цель:** научиться создавать структуры и массивы, а также работать с ними.

**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.).

**Ход работы**

**Задание 1**

**Листинг**

Файл LOG\_L\_1\_1.cpp

#define MASSIVE\_SIZE 10

#include <stdio.h>

#include <limits.h>

#include <time.h>

#include <random>

#define \_CRT\_SECURE\_NO\_WARNINGS\_

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int maxValue = INT\_MIN;

int minValue = INT\_MAX;

int massive[MASSIVE\_SIZE] = { 0 };

int\* p = massive;

for (int i = 0; i < MASSIVE\_SIZE; i++) {

\*p = rand() % 100;

printf("%d\n", \*p);

if (\*p > maxValue) maxValue = \*p;

if (\*p < minValue) minValue = \*p;

p++;

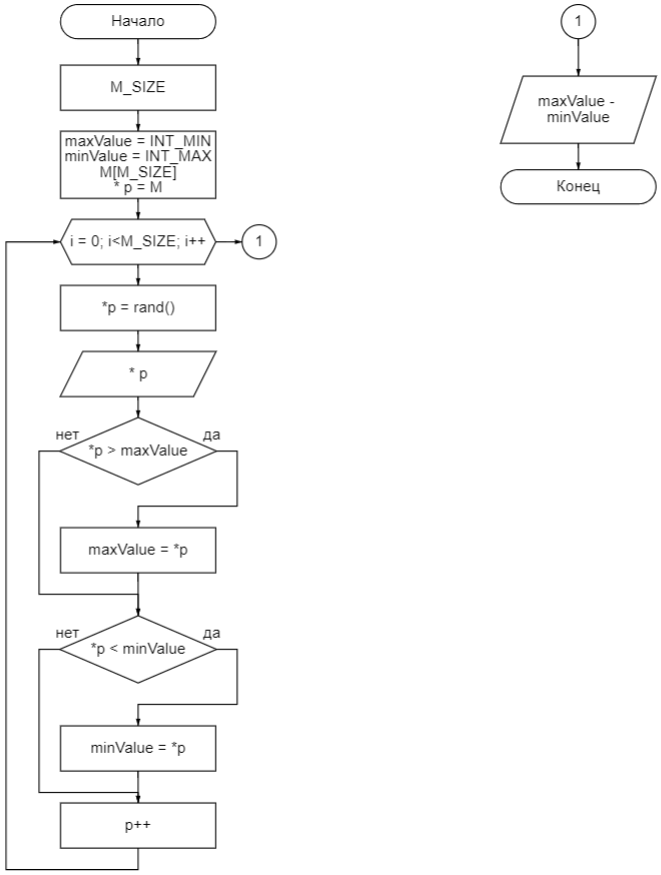
}

printf("\n%s%d\n", "Разница между максимальным и минимальным значением: ", maxValue - minValue);

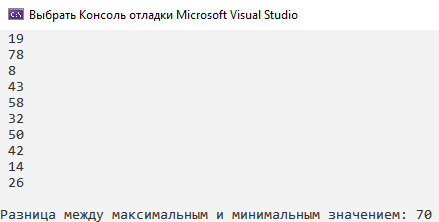
return 0;

}

**Схема программы**



**Результат работы программы**



**Задание 2**

**Листинг**

Файл LOG\_L\_1\_2.cpp

#define MASSIVE\_SIZE 20

#include <stdio.h>

#include <time.h>

#include <random>

#define \_CRT\_SECURE\_NO\_WARNINGS

int main() {

srand(time(NULL));

int massive[MASSIVE\_SIZE] = { 0 };

int\* p = massive;

for (int i = 0; i < MASSIVE\_SIZE; i++) {

\*p = rand() % 100; // 0 - 99

printf("%d ", \*p);

p++;

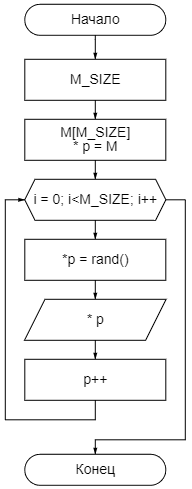
}

printf("\n");

return 0;

}

**Схема программы**



**Результат работы программы**



**Задание 3**

**Листинг**

Файл LOG\_L\_1\_3.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

int\* createMassive(int size);

int fillMassive(int size, int\* p);

int main() {

setlocale(LC\_ALL, "Russian");

int size = 0;

int result;

int\* p;

printf("Введите размер массива: ");

result = scanf("%d", &size);

if (result != 1 && size <= 0) {

printf("\nОшибка: Размер массива должен быть больше нуля\n");

return 1;

}

else {

p = createMassive(size);

result = fillMassive(size, p);

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

printf("%d ", \*p);

p++;

}

printf("\n");

}

return result;

}

int\* createMassive(int size) {

int\* p;

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

return p;

}

int fillMassive(int size, int\* p) {

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

printf("\n%s%d%s", "Введите элемент ", i+1, ": ");

int result = scanf("%d", p);

retry:

if (result != 1) {

printf("Ошибка: введены не десятичные символы");

return 1;

}

p++;

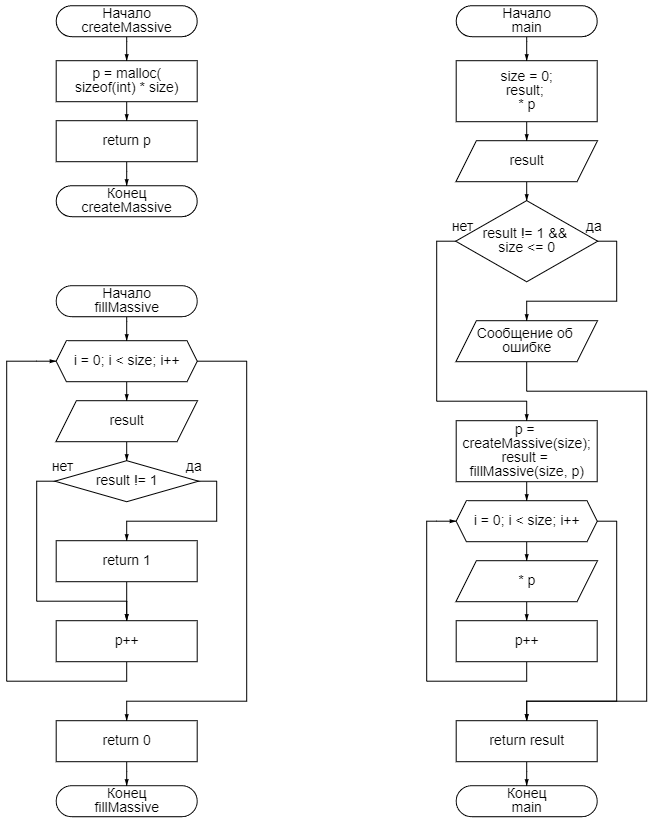
}

printf("\nВвод массива завершен. Полученный массив:\n");

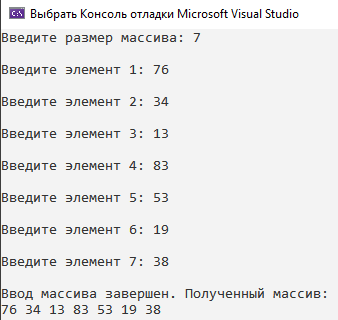
return 0;

}

**Схема программы**



**Результат работы программы**



**Задание 4**

**Листинг**

Файл LOG\_L\_1\_4.cpp

#define MATRIX\_ROWS 5

#define MATRIX\_COLS 8

#include <stdio.h>

#include <time.h>

#include <random>

#include <locale.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

int main() {

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

int M[MATRIX\_ROWS][MATRIX\_COLS] = {0};

int summ = 0;

for (int i = 0; i < MATRIX\_ROWS; i++) {

summ = 0;

for (int j = 0; j < MATRIX\_COLS; j++) {

M[i][j] = rand() % 50; // 0 - 49

summ += M[i][j];

printf("%-2d ", M[i][j]);

}

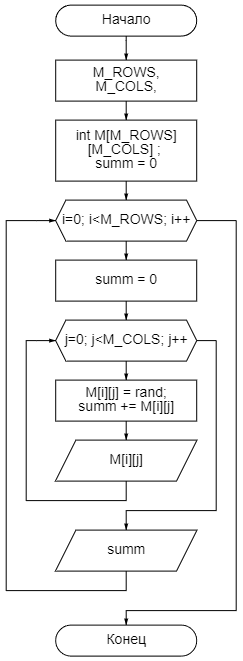
printf(" Сумма: %d\n", summ);

}

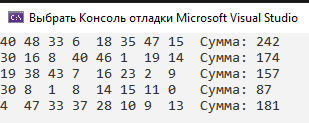
return 0;

}

**Схема программы**



**Результат работы программы**



**Задание 5**

**Листинг**

Файл LOG\_L\_1\_5.cpp

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

#include <string.h>

#define RECORDS\_COUNT 3

typedef struct {

char lastName[20];

char firstName[20];

char faculty[20];

int ZachetkaNum;

} base;

void printStruct(base student[], int id);

void search(base student[]);

int cmpString(char string1[20], char string2[20]);

void fillBase(base student[]);

int main()

{

int searchType = 0;

setlocale(LC\_ALL, "Russian");

base student[RECORDS\_COUNT];

fillBase(student);

search(student);

return 0;

}

void fillBase(base student[]) {

for (int i = 0; i < RECORDS\_COUNT; i++) {

printf("%s%d", "Студент №", i+1);

printf("\nВведите фамилию студента: ");

scanf("%19s", &student[i].lastName);

printf("\nВведите имя студента: ");

scanf("%19s", &student[i].firstName);

printf("\nВведите факультет студента: ");

scanf("%19s", &student[i].faculty);

printf("\nВведите номер зачетной книжки студента: ");

scanf("%d", &student[i].ZachetkaNum);

}

}

int cmpString(char string1[20], char string2[20]) {

int i = 0;

while (string1[i] != '\0' || string1[i] != '\0') {

if (string1[i] == string2[i]) {

i++;

continue;

}

else {

return 0;

}

}

return 1;

}

void printStruct(base student[], int id) {

printf("\n%s%s\n%s%s\n%s%s\n%s%d\n",

"Фамилия: ", student[id].lastName,

"Имя: ", student[id].firstName,

"Факультет: ", student[id].faculty,

"Номер зачетной книжки: ", student[id].ZachetkaNum);

}

void search(base student[]) {

int type = 0;

int success = 0;

int id;

char string[20];

int zachetkaSearch;

printf("\nПоиск по\n 1 - Фамилия\n 2 - Имя\n 3 - Факультет\n 4 - Номер зачетной книжки\n");

scanf("%1d", &type);

switch (type)

{

case 1:

printf("Введите запрос для поиска: ");

scanf("%19s", &string);

for (int i=0; i< RECORDS\_COUNT; i++){

success = cmpString(string, student[i].lastName);

if (success) {

id = i;

printStruct(student, id);

break;

}

else continue;

}

break;

case 2:

scanf("%19s", &string);

for (int i = 0; i < RECORDS\_COUNT; i++) {

success = cmpString(string, student[i].firstName);

if (success) {

id = i;

printStruct(student, id);

break;

}

else continue;

}

break;

case 3:

scanf("%19s", &string);

for (int i = 0; i < RECORDS\_COUNT; i++) {

success = cmpString(string, student[i].faculty);

if (success) {

id = i;

printStruct(student, id);

break;

}

else continue;

}

break;

case 4:

scanf("%d", &zachetkaSearch);

for (int i = 0; i < RECORDS\_COUNT; i++) {

if (zachetkaSearch == student[i].ZachetkaNum) {

id = i;

success = 1;

printStruct(student, id);

break;

}

else continue;

}

break;

default:

printf("\nОшибка: неверное значение\n");

search(student);

break;

if (!success) {

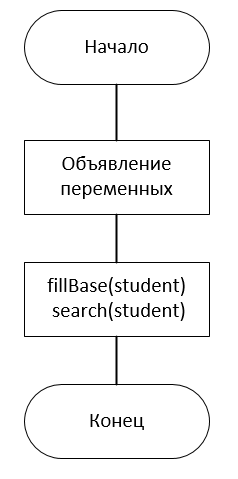
printf("Запись не найдена");

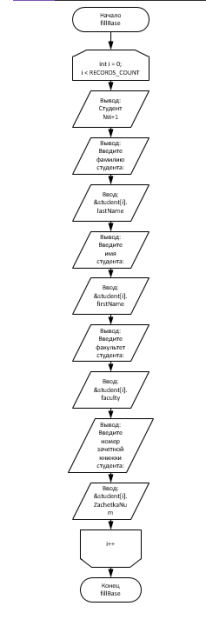
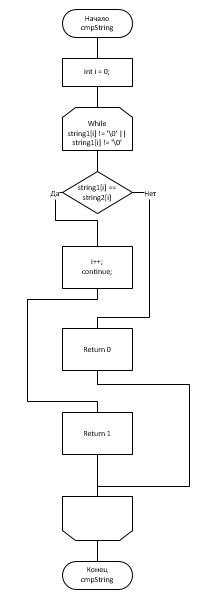
}

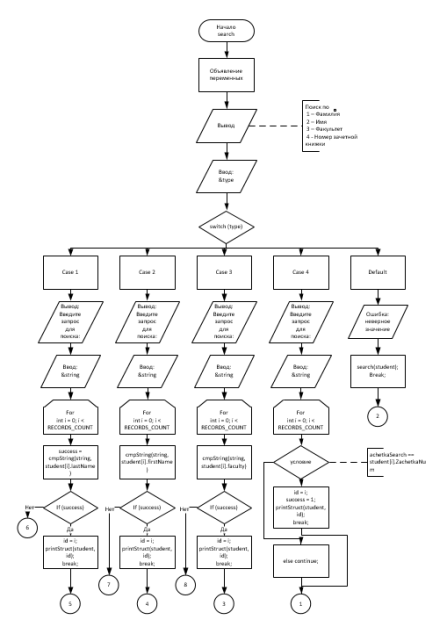
}

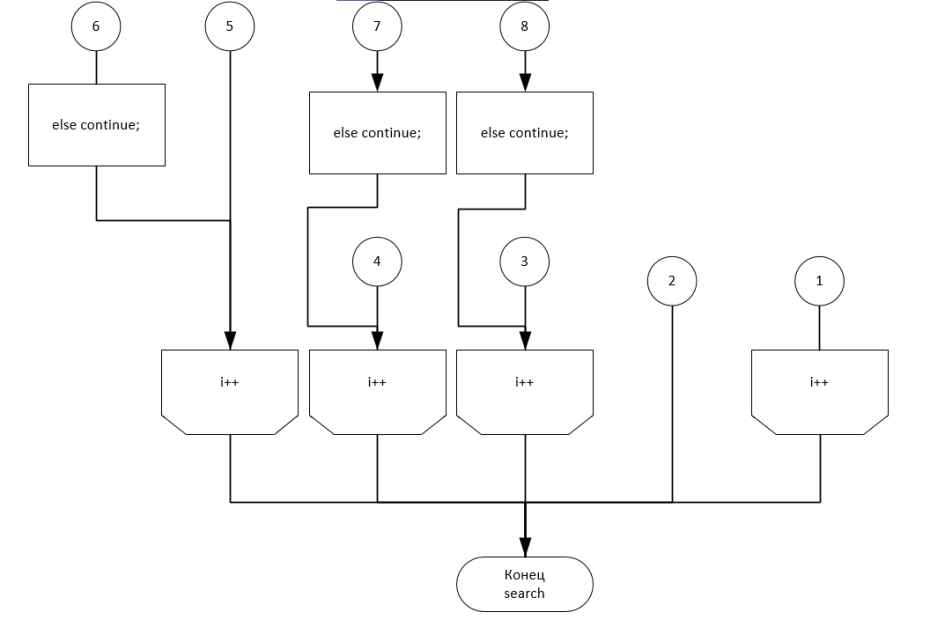
}

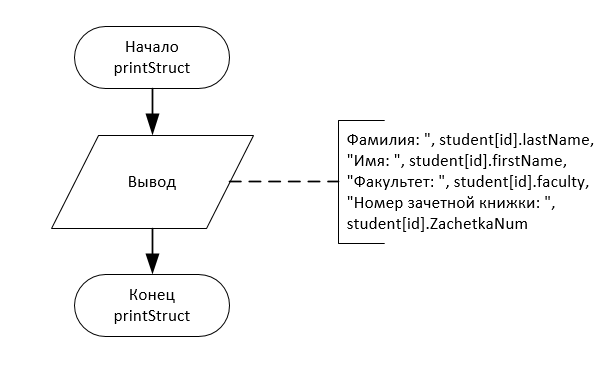
**Схема программы**



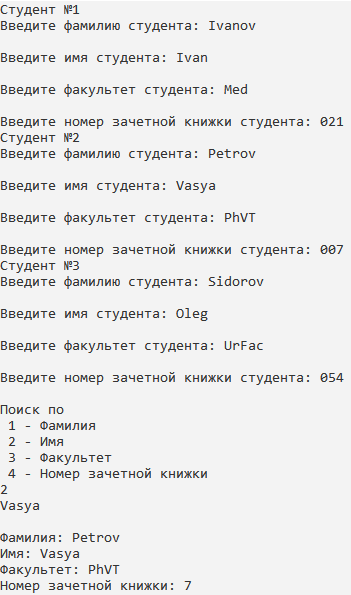






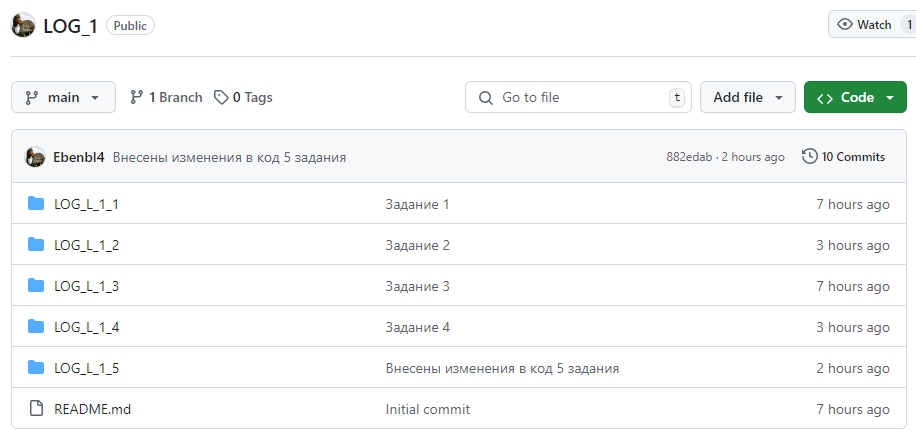


**Результат работы программы**



**Совместная работа**

Проекты всех заданий были загружены на GitHub.



**Вывод**

В ходе выполнения лабораторной работы были разработаны 5 программ, удовлетворяющие условию соответствующих заданий. Был получен опыт в написании программы с массивами и структурами данных на языке Си.