Пензенский государственный университет Кафедра «Вычислительная техника»

**Отчет**о выполнении лабораторной работы №1 по дисциплине “Логика и основа алгоритмизации в инженерных задачах” на тему**:**

**Простые структуры данных**

Выполнили студенты гр. 19ВВ4:   
Хлыстов А.Ю.  
Привалов А.Э.

Проверили:

Юрова О.В.  
Митрохин М. А.

Пенза, 2020 г.

**Название**Простые структуры данных

**Цель работы**Вспомнить и применить на практике принципы работы с простыми структурами даных.

**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.).

**Код программы**

Lab 1.1:  
  
//Лабораторная работа номер 1 получаеца.

#include <limits.h>

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <Windows.h>

int max = INT\_MIN;

int min = INT\_MAX;

typedef struct result

{

int difference;

int\* the\_sum\_of\_the\_values\_rows;

int\* the\_sum\_of\_the\_values\_cols;

}result;

int random\_sign() {

srand(time(NULL));

int random = rand();

if (random % 2 == 0) {

return -1;

}

return 1;

}

int main()

{

srand(time(NULL));

result answer\_task;

int size\_x, size\_y;

int\*\* matrix;

int value;

printf("enter size of matrix:\n column =");

scanf\_s("%d", &size\_x);

printf("\n lines =");

scanf\_s("%d", &size\_y);

matrix = (int\*\*)malloc(size\_y \* sizeof(int\*));

answer\_task.the\_sum\_of\_the\_values\_cols = (int\*)calloc(size\_x, sizeof(int));

answer\_task.the\_sum\_of\_the\_values\_rows = (int\*)calloc(size\_y, sizeof(int));

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

matrix[i] = (int\*)malloc(size\_x \* sizeof(int));

for (int j = 0; j < size\_x; j++) {

matrix[i][j] = (rand() % 100) \* (random\_sign());

}

}

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

for (int j = 0; j < size\_x; j++) {

value = matrix[i][j];

if (value > max) {

max = value;

}

if (value < min) {

min = value;

}

answer\_task.the\_sum\_of\_the\_values\_rows[i] += value;

answer\_task.the\_sum\_of\_the\_values\_cols[j] += value;

}

}

answer\_task.difference = max - min;

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

printf("\n");

for (int j = 0; j < size\_x; j++) {

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

}

printf("= %d", answer\_task.the\_sum\_of\_the\_values\_rows[i]);

}

printf("\n");

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

printf("=\t");

}

printf("\n");

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

printf("%d\t", answer\_task.the\_sum\_of\_the\_values\_cols[i]);

}

printf("\n\nDifferent (max-min) = %d", answer\_task.difference);

for (int i = 0; i < size\_y; i++)

{

free(matrix[i]);

}

free(matrix);

return 0;

}

Lab 1.2:

[#define](https://vk.com/im?sel=273062832&st=%23define) \_CRT\_SECURE\_NO\_WARNINGS  
[#include](https://vk.com/im?sel=273062832&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=273062832&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=273062832&st=%23include) <string.h>  
[#include](https://vk.com/im?sel=273062832&st=%23include) <locale.h>  
  
int main() {  
setlocale(LC\_ALL, "Rus");  
struct student  
{  
char famil[20];  
char name[20], facult[20];  
int Nomzach;  
} stud[3];  
  
int full = -1;  
int part[4] = { -1 , -1, -1, -1};  
int ipart = 0;  
  
int i;  
int call = 0;  
int coincidences = 0;  
struct student find;  
  
for (i = 0; i < 3; i++)  
{  
printf("Введите фамилию студента\n"); scanf("%20s", stud[i].famil);  
}  
for (i = 0; i < 3; i++)  
{  
printf("Введите имя студента %s\n", stud[i].famil); scanf("%20s", stud[i].name);  
}  
for (i = 0; i < 3; i++)  
{  
printf("Введите название факультета студента %s %s\n", stud[i].famil, stud[i].name); scanf("%20s", stud[i].facult);  
}  
for (i = 0; i < 3; i++)  
{  
printf("Введите номер зачётной книжки студента %s %s\n", stud[i].famil, stud[i].name); scanf("%d", &stud[i].Nomzach);  
}  
  
printf("Ввдите данные для поиска:\n");  
  
printf("Введите фамилию студента\n"); scanf("%20s", find.famil);  
printf("Введите имя студента %s\n", find.famil); scanf("%20s", [find.name](https://vk.com/away.php?to=http%3A%2F%2Ffind.name&cc_key=));  
printf("Введите название факультета студента %s %s\n", find.famil, [find.name](https://vk.com/away.php?to=http%3A%2F%2Ffind.name&cc_key=)); scanf("%20s", find.facult);  
printf("Введите номер зачётной книжки студента %s %s\n", find.famil, [find.name](https://vk.com/away.php?to=http%3A%2F%2Ffind.name&cc_key=)); scanf("%d", &find.Nomzach);  
  
for (i = 0; i < 3; i++)  
{  
if (strcmp([find.name](https://vk.com/away.php?to=http%3A%2F%2Ffind.name&cc_key=), stud[i].name) == 0) {  
coincidences += 1;  
}  
if (strcmp(find.famil, stud[i].famil) == 0) {  
coincidences += 1;  
}  
if (strcmp(find.facult, stud[i].facult) == 0) {  
coincidences += 1;  
}  
if (find.Nomzach == stud[i].Nomzach) {  
coincidences += 1;  
}  
if (coincidences == 4) {  
full = i;  
break;  
}  
if (coincidences == 3) {  
part[ipart] = i;  
ipart++;  
coincidences = 0;  
}  
else {  
continue;  
}  
}  
if (full != -1) {  
printf("Найдена запись: \n");  
printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n", stud[full].famil, stud[full].name,  
stud[full].facult, stud[full].Nomzach);  
}  
else {  
if (part[0] != -1) {  
i = 0;  
printf("Возможно вы имели ввиду:");  
while (part[i] != -1) {  
printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n", stud[i].famil, stud[i].name,  
stud[i].facult, stud[i].Nomzach);  
i++;  
}  
}  
else {  
printf("Ничего не найдено\n");  
}  
}  
return 0;  
}

**Схема программы**

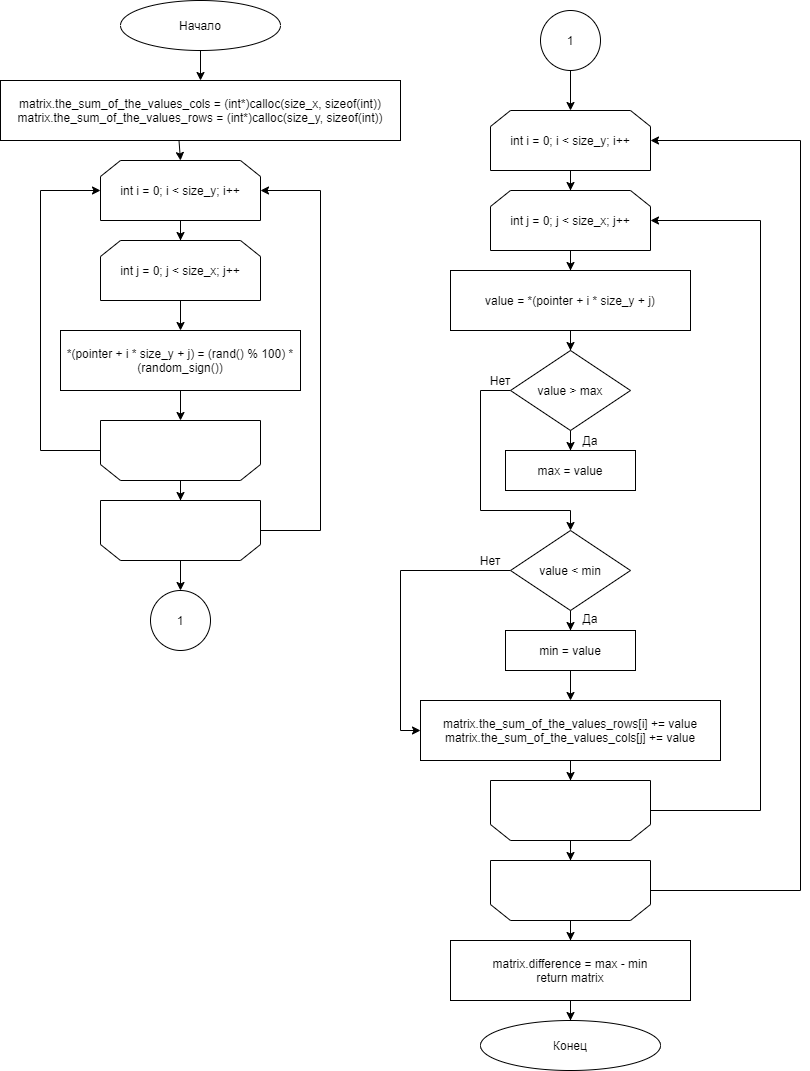
****

Рисунок Схема программы Lab1.1.

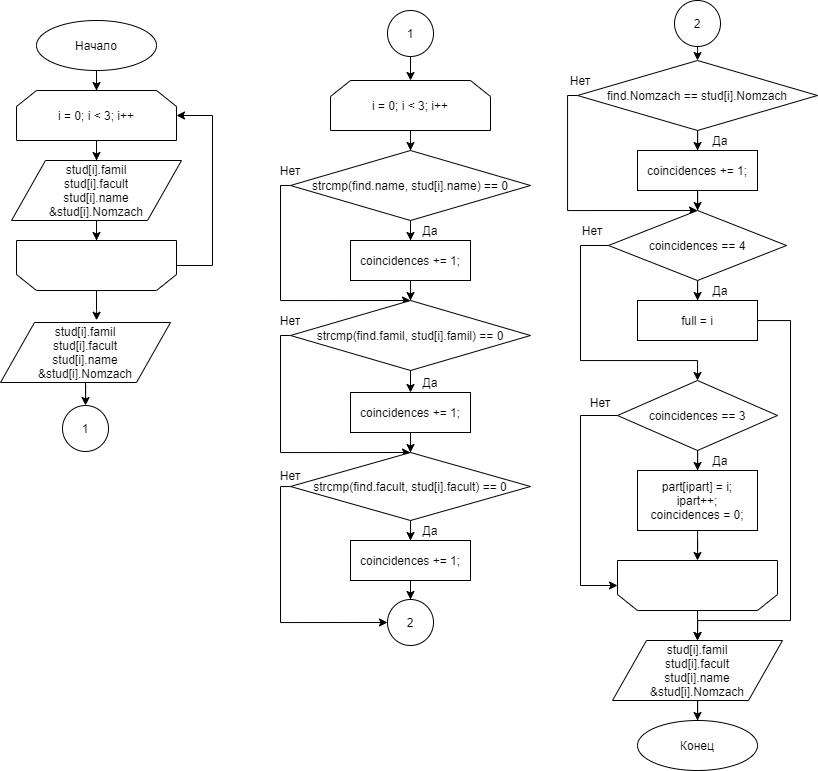
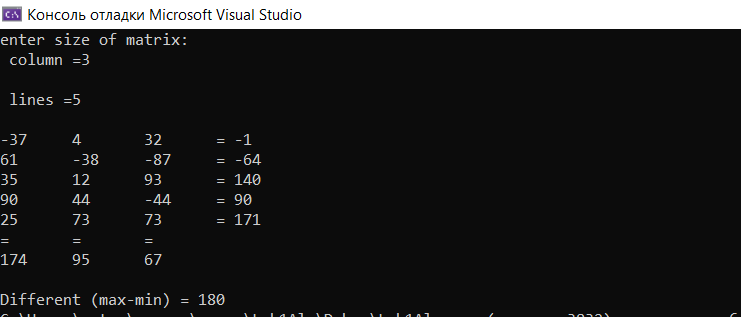
****

Рисунок Схема функции main документа Lab 1.2

**Результаты работы программы**



**Выводы**В ходе выполнения лабораторной работы была разработана программа, содержащая простые структуры данных и различные операции с ними.