Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему « Простые структуры данных»

Выполнили:

студенты группы 20ВВ2

Киреев Б.П.

Верховский М.В.

Лукин В.Д.

Приняли:

д.т.н. Митрохин М.А.

к.т.н. доцент Юрова О.В.

Пенза 2021

Название: Простые структуры данных.

Цель работы: Научиться работать с простыми структурами 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:

Файл Lab1\_1.cpp

#include <iostream>

#include <ctime>

#include <conio.h>

using namespace std;

int Mass1[10], Max, Min;

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0)); //по-настоящему случайная генерация массива

cout << "Генерируемый массив: "; // вывод

for (int i = 0; i < 10; i++) { // заполнение массива

Mass1[i] = 1 + rand() % 100; //1+ это от 0 до 100

cout << Mass1[i] << " ";//вывод массива

if (i == 0) { Max = Mass1[i]; Min = Mass1[i]; }//когда i = 0, назначаем максу и мину значения 0-го элем-та массива

else if (Max < Mass1[i]) { Max = Mass1[i]; }

else if (Min > Mass1[i]) { Min = Mass1[i]; }

}

cout << "\nМаксимум: " << Max << endl;

cout << "Минимум: " << Min << endl;

cout << "Разница между максимумом и минимумом: " << Max - Min << endl;

getch();

return 0;

}

Пояснительный текст к программе:

Данная программа создает массив и заполняет его случайными числами. При этом выполняет поиск максимального и минимального элементов, а также их разницы.

Результаты работы программы:

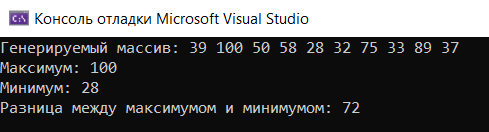


Рисунок №1 – Работа программы из задания №1

**Задание 2:**

Листинг:

Файл Lab1\_2.cpp

#include <iostream>

#include <time.h>

#include <conio.h>

using namespace std;

int Mass2[10];

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

cout << "\nГенерируемый массив: ";

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

Mass2[i] = -50 + rand() % 100;

cout << Mass2[i] << " ";

}

cout << endl;

getch();

return 0;

}

Пояснительный текст к программе:

Данная программа создает массив и заполняет его случайными числами.

Результаты работы программы:

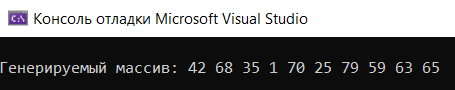


Рисунок №2 – Результат работы программы №2

**Задание 3:**

Листинг:

Файл Lab1\_3.cpp

#include <iostream>

#include <ctime>

#include <malloc.h>

#include <conio.h>

using namespace std;

int n, \*\*p = NULL; // n - к-во эл-тов в массиве, p -указатель на первый элемент в выделяемой памяти

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0)); //по-настоящему случайная генерация массива

cout << "\nВведите размерность массива: ";

cin >> n; // ввод к-ва эл-тов в массиве

cout << "Массив: \n";

p = (int\*\*)calloc(n, 3);

for(int i = 0; i < n; i++){p[i] = (int\*)calloc(n, 3);} // выделение места под n элементов с рамерностью 2 байта(int - 2 байта)

//присваиваем указателю адрес первого эл-та в памяти

cout.precision(3\*n);

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

for(int m = 0; m < n; m++){

p[i][m] = 1 + rand() % 100;

cout << p[i][m] << " ";

}

cout<<endl;

}

getch();

return 0;

}

Пояснительный текст к программе:

Данная программа выделяет место под массив и заполняет его случайными числами.

Результаты работы программы:

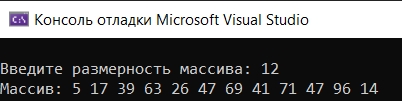


Рисунок №3 – Результат работы программы №3

**Задание 4:**

Листинг:

Файл Lab1\_4.cpp

#include <iostream>

#include <ctime>

using namespace std;

int Mass3[10][10], x = 10, y = 10, Sum = 0;// двумерный массив 10\*10

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(0));

for (int i = 0; i < x; i++) {//движение по оси х

for (int h = 0; h < y; h++) { // движение по оси у

Mass3[i][h] = 1 + rand() % 100; // проходим по всему у -ку и прибавляем к х единицу, а потом заново заполняем у

}

}

for (int i = 0; i < x; i++) {//проход по Х

Sum = 0;

cout << "Строка " << i << ": ";

for (int h = 0; h < y; h++) { // проход по У

Sum += Mass3[i][h]; // прибавление к сум каждого эл-та в строке

cout << Mass3[i][h] << " "; // отпечатываем элементы строки

}

cout << endl;//переход на новую строку в консоли

cout << "Сумма строки " << i << ": " << Sum << endl;

}

return 0;

}

Пояснительный текст к программе:

Данная программа заполняет квадратную матрицу случайными значениями и вычисляет сумму значений в каждой строке.

Результаты работы программы:

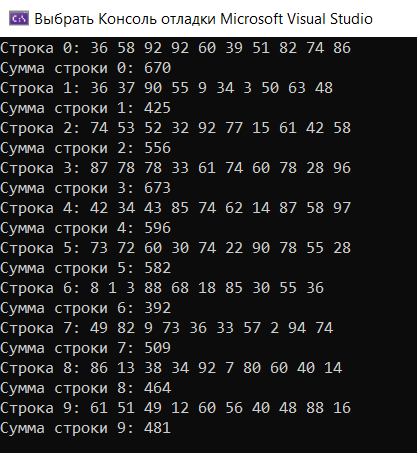


Рисунок №4 – Результат работы программы №4

**Задание 5:**

Листинг:

Файл Lab1\_5.cpp

#include <iostream>

#include <ctime>

#include <windows.h>

using namespace std;

struct student {

string Name = "-", Surname = "-", Group = "-";

int Curse = 0, TicketNum = 0;

} StudentList[5], StudentText; // список студентов и критерии поиска

bool stopper = false;

int choose;

int main() {

SetConsoleCP(1251); //Подключение кириллицы

SetConsoleOutputCP(1251); //Для считывания и вывода

for (int i = 0; i < 5; i++) { // цикл ввода студентов (заполнение каждого параметра структуры) (полностью заполняем массив)

cout << "Введите имя: ";

cin >> StudentList[i].Name;

cout << "Введите фамилию: ";

cin >> StudentList[i].Surname;

cout << "Введите группу: ";

cin >> StudentList[i].Group;

cout << "Введите курс: ";

cin >> StudentList[i].Curse;

cout << "Введите номер студенческого билета: ";

cin >> StudentList[i].TicketNum;

cout << endl;

}

//Заполняем структуру StudentText

cout << "Введите критерии поиска: " << endl;

cout << "Введите имя: ";

cin >> StudentText.Name;

cout << "Введите фамилию: ";

cin >> StudentText.Surname;

cout << "Введите группу: ";

cin >> StudentText.Group;

cout << "Введите курс: ";

cin >> StudentText.Curse;

cout << "Введите номер студенческого билета: ";

cin >> StudentText.TicketNum;

cout << endl;

for (int i = 0; i < 5; i++) {

if (StudentText.Name == StudentList[i].Name) { stopper = true; }

if (StudentText.Surname == StudentList[i].Surname) { stopper = true; }

if (StudentText.Group == StudentList[i].Group) { stopper = true; }

if (StudentText.Curse == StudentList[i].Curse) { stopper = true; }

if (StudentText.TicketNum == StudentList[i].TicketNum) { stopper = true; }

if (stopper == true) { cout << StudentList[i].Name << " " << StudentList[i].Surname << " " << StudentList[i].Group << " " << StudentList[i].Curse << " " << StudentList[i].TicketNum << endl; }

stopper = false;

}

return 0;

}

Пояснительный текст к программе:

Данная программа осуществляет поиск среди структур student структуру с заданными параметрами (фамилией, именем, группой, курсом, номером студ. билета).

Результаты работы программы:

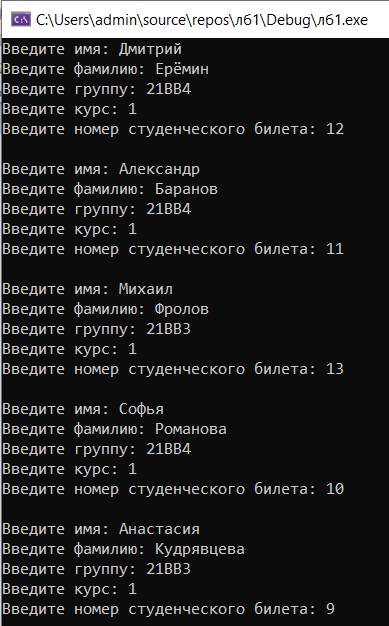


Рисунок №5 – Результат работы программы №5

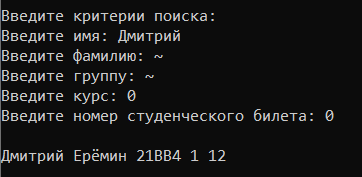


Рисунок №6 – Результат работы программы №5

Выводы: В ходе этой лабораторной работы, мы смогли изучить простые структуры данных.