МИНИСТЕРВСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине: "Логика и основы алгоритмизации в инженерных задачах"

на тему: "Простые структуры данных"

Выполнил:

Студент группы 23ВВВ2

Кочегин Валерий

Принял:

Митрохин М. А.

Юрова О. В.

Пенза 2024

**Цель**

Изучение и повторение простух 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**

**Файл 1-zadan.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <limits.h>

int main() {

//---------------1-----------------

int mi = INT\_MAX, ma = INT\_MIN;

int a[10] = { 0 };

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

a[i] = i;

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

}

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

if (a[i] < mi) { mi = a[i]; };

if (a[i] > ma) { ma = a[i]; };

}

printf("\n%d", ma - mi);

printf("\n%d", ma);

printf("\n%d", mi);

return 0;

}

**Файл 2-zadan.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

int main() {

//---------------2-----------------

int mi2 = 0, ma2 = 100, m = 0;

int b[10] = { 0 };

srand(time(NULL));

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

m = mi2 + rand() % (ma2 - mi2 + 1);

b[i] = m;

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

}

return 0;

}

**Файл 3-zadan.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

#include <limits.h>

#include <time.h>

int main() {

// ---------------3-----------------

int\* c;

int n, z = 0;

printf("Введите размер массива: ");

scanf\_s("%d", &n);

c = (int\*)malloc(n \* sizeof(int));

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

printf("Введите %d эл массива: ", i+1);

scanf\_s("%d", &z);

c[i] = z;

}

printf("\n");

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

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

}

return 0;

}

**Файл 4-zadan.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

#include <limits.h>

#include <time.h>

int main() {

// ---------------4-----------------

setlocale(LC\_ALL, "Russian");

int mi3 = 0, ma3 = 5;

int sum\_str = 0, sum\_sto = 0;

int n, m, k;

printf("Введите размер массива через пробел): ");

scanf\_s("%d %d", &n, &m);

int\*\* h = (int\*\*)malloc(n \* sizeof(int\*));

srand(time(NULL));

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

h[i] = (int\*)malloc(m \* sizeof(int));

}

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

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

h[i][j] = mi3 + rand() % (ma3 - mi3 + 1);

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

}

printf("\n");

}

printf("\n");

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

sum\_str = 0;

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

sum\_str += h[i][j];

}

printf("Сумма %d строки: %d\n", i + 1, sum\_str);

}

printf("\n");

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

sum\_sto = 0;

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

sum\_sto += h[i][j];

}

printf("Сумма %d столбца: %d\n", j + 1, sum\_sto);

}

return 0;

}

**Файл 5-zadan.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <windows.h>

#include <limits.h>

#include <time.h>

#define NAME 15

#define MAX\_S 5

// ---------------5-----------------

struct student {

char Name[NAME];

char Name2[NAME];

int Age, Nomzach;

}students[MAX\_S] = {

{"Иван", "Пичаев", 18, 12345},

{"Сергей", "Чупраков", 19, 12346},

{"Валерий", "Кочегин", 18, 12347},

{"Илья", "Ваняшин", 19, 12348},

{"Никита", "Лунтев", 19, 12349}

};

void searchByName(struct student students[], int c, const char\* Name, const char\* Name2) {

int found = 0;

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

if (strcmp(students[i].Name, Name) == 0 && strcmp(students[i].Name2, Name2) == 0) {

printf("Ваш студент: %s %s. Его возраст: %d. Номер зачетки: %d\n", students[i].Name, students[i].Name2, students[i].Age, students[i].Nomzach);

found = 1;

}

}

if (!found) {

printf("Такого студента нет :/\n");

}

}

void searchByAge(struct student students[], int c, int age) {

int found = 0;

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

if (students[i].Age == age) {

printf("Ваш студент: %s %s. Его возраст: %d. Номер зачетки: %d\n", students[i].Name, students[i].Name2, students[i].Age, students[i].Nomzach);

found = 1;

}

}

if (!found) {

printf("Студента с таким возрастом нет :/\n");

}

}

void searchByNomzach(struct student students[], int c, int Nomzach) {

int found = 0;

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

if (students[i].Nomzach == Nomzach) {

printf("Ваш студент: %s %s. Его возраст: %d. Номер зачетки: %d\n", students[i].Name, students[i].Name2, students[i].Age, students[i].Nomzach);

found = 1;

}

}

if (!found) {

printf("Студента с таким возрастом нет :/\n");

}

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int c = MAX\_S;

//for (int i = 0; i < c; i++) {

// printf("Имя студента %d: ", i + 1);

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

// printf("Фамилия студента %d: ", i + 1);

// scanf("%s", students[i].Name2);

// printf("Возраст студента %d: ", i + 1);

// scanf("%d", &students[i].Age);

// printf("Ночер зачетки студента %d: ", i + 1);

// scanf("%d", &students[i].Nomzach);

//}

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

printf("Студент: %s %s. Возраст: %d. Номер зачетки: %d.\n", students[i].Name, students[i].Name2, students[i].Age, students[i].Nomzach);

}

char choice;

do {

printf("\nВыберите метод поиска:\n");

printf("1. Поиск по имени и фамилии\n");

printf("2. Поиск по возрасту\n");

printf("3. Поиск по номеру зачетки\n");

printf("4. Выход\n");

printf("Ваш выбор: ");

scanf("%s", &choice);

switch (choice) {

case '1': {

char search1[NAME];

char search2[NAME];

printf("Введите имя для поиска: ");

scanf("%s", search1);

printf("Введите фамилию для поиска: ");

scanf("%s", search2);

searchByName(students, c, search1, search2);

break;

}

case '2': {

int searchAge;

printf("Введите возраст для поиска: ");

scanf("%d", &searchAge);

searchByAge(students, c, searchAge);

break;

}

case '3': {

int searchNomzach;

printf("Введите номер зачетки для поиска: ");

scanf("%d", &searchNomzach);

searchByNomzach(students, c, searchNomzach);

break;

}

case '4':

printf("Выход из программы.\n");

break;

default:

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

break;

}

} while (choice != '4');

return 0;

}

**Пояснительный текст к программе**

Задание 1

Программа создает массив из 10 элементов и заполняет его цифрами от 0 до 9. Программа находит минимальное и максимальное значение в массиве и вычисляет их разницу и выводит результат.

Задание 2

Программа создает массив из 10 элементов и заполняет его случайными значениями от 0 до 100 с использованием функции srand().

Задание 3

Программа запрашивает размер массива и его элементы, затем выводит полученный массив.

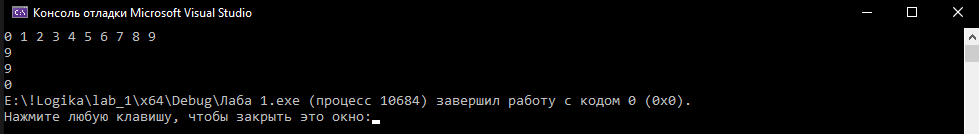
Задание 4

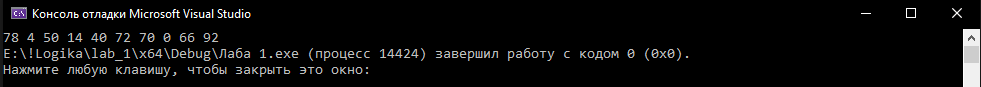
Программа запрашивает размер двумерного массива, заполняет его случайными значениями от 0 до 5. После программа вычисляет сумму элементов в каждом столбце и в каждой строке и выводит результат.

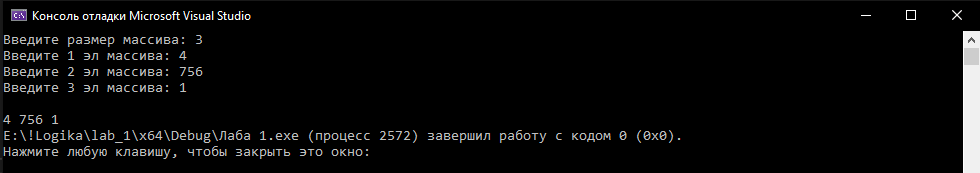
Задание 5

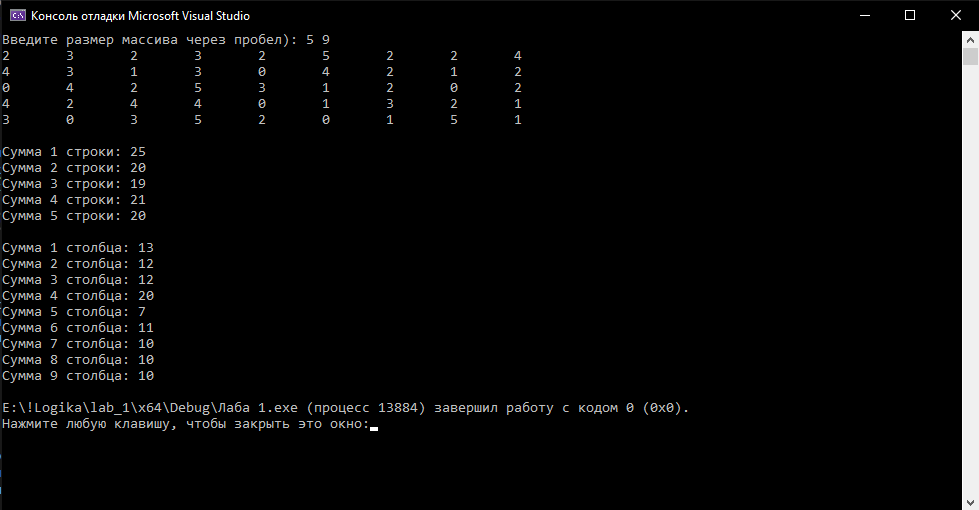
Программа содержит информацию о 5 студентах. С помощью меню поиска можно выбрать метод поиска студента. 1. С помощью имени и фамилии студента. 2. С помощью возраста студента. 3. С помощью номера зачетной книжки студента. Предусмотрен выход из программы в меню.

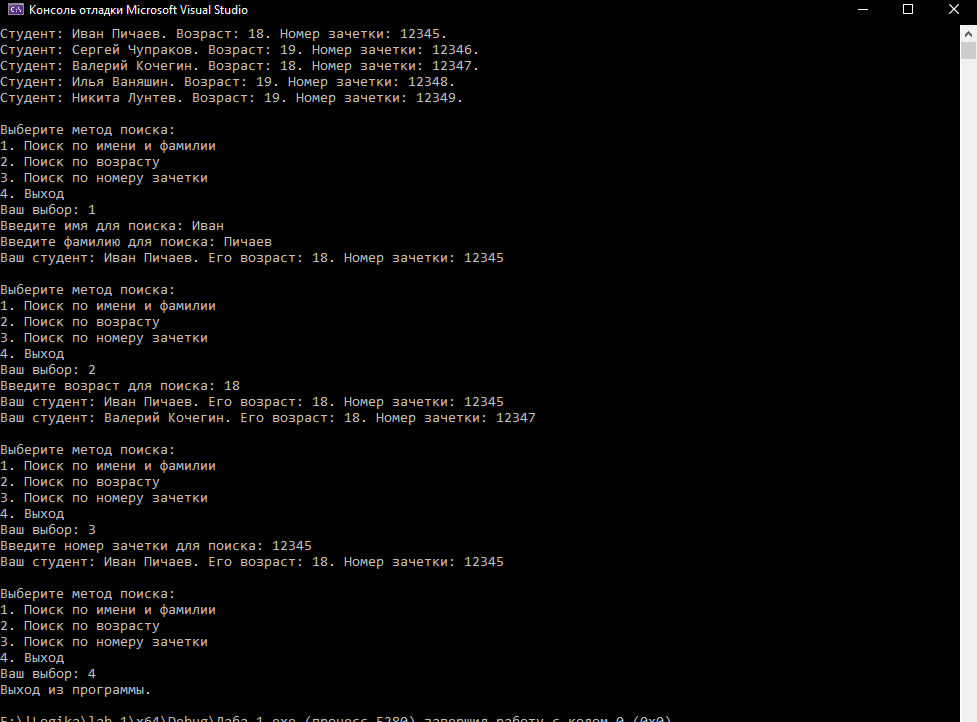
**Результат программы**

****1 Рис. - Результат работы программы 1-zadan.cpp

2 Рис. - Результат работы программы 2-zadan.cpp

3 Рис. - Результат работы программы 3-zadan.cpp

4 Рис. - Результат работы программы 4-zadan.cpp

5 Рис. - Результат работы программы 5-zadan.cpp

**Вывод**

В ходе выполнения лабораторной работы были разработаны программы для выполнения заданий Лабораторной работы №1. В процессе выполнения работы были использованы знания о простейших структурах данных.