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»

Вариант 1

Выполнили:

Студенты группы 24ВВВ1

Тусков А.А.

Цыбузин Д.В.

Приняли:

к.т.н., доцент Юрова О.В.

  к.т.н., доцент Деев М.В.

Пенза 2025**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(уже с изменениями для дополнительных заданий):**

#include <iostream>

#include <climits>

#include <cstdlib>

#include <ctime>

#include <cstring>

#include <cstdio>

#include <windows.h>

#include <locale.h>

using namespace std;

// Задание 1

void task1() {

const int size = 10;

int\* arr = new int[size] { 5, 2, 8, 1, 9, 3, 7, 4, 6, 0 };

int min\_val = INT\_MAX;

int max\_val = INT\_MIN;

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

if (arr[i] < min\_val) min\_val = arr[i];

if (arr[i] > max\_val) max\_val = arr[i];

}

cout << "minimum things: " << min\_val << endl;

cout << "maksimum things: " << max\_val << endl;

cout << "differents: " << max\_val - min\_val << endl;

delete[] arr;

}

// Задание 2

void task2() {

srand(time(NULL));

const int size = 10;

int\* arr = new int[size];

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

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

}

cout << "massiv: ";

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

cout << arr[i] << " ";

}

cout << endl;

delete[] arr;

}

// Задание 3

void task3() {

int size;

cout << "size of: ";

cin >> size;

int\* arr = new int[size];

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

cin >> arr[i];

}

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

cout << arr[i] << " ";

}

cout << endl;

delete[] arr;

}

// Задание 4

void task4() {

const int rows = 4;

const int cols = 4;

int\*\* matrix = new int\* [rows];

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

matrix[i] = new int[cols];

}

int value = 1;

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

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

matrix[i][j] = value++;

}

}

cout << "Matrix:" << endl;

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

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

cout << matrix[i][j] << "\t";

}

cout << endl;

}

cout << "\nSumma po stolbcam:" << endl;

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

int col\_sum = 0;

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

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

}

cout << "Stolbec " << j + 1 << ": " << col\_sum << endl;

}

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

delete[] matrix[i];

}

delete[] matrix;

}

// Задание 5

struct Student {

char famil[20];

char name[20];

char facult[20];

int Nomzach;

};

void printStudents(Student\* students, int num) {

cout << "\nList of students:" << endl;

cout << "==============================================" << endl;

cout << "№ Surname Name facultet Zachetka" << endl;

cout << "--------------------------------------------------------------" << endl;

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

printf("%-2d %-12s %-12s %-12s %-10d\n",

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

students[i].facult, students[i].Nomzach);

}

cout << "==============================================" << endl;

}

void swapStudents(Student\* s1, Student\* s2) {

Student temp = \*s1;

\*s1 = \*s2;

\*s2 = temp;

}

void reorderStudents(Student\* students, int num) {

int index1, index2;

cout << "\nPoyadok studentov:" << endl;

printStudents(students, num);

cout << "Enter numbers of students (to 1 to " << num << "):" << endl;

cout << "First student: ";

cin >> index1;

cout << "Second student: ";

cin >> index2;

if (index1 < 1 || index1 > num || index2 < 1 || index2 > num) {

cout << "False numbers of students!" << endl;

return;

}

if (index1 == index2) {

cout << "It's a same student!" << endl;

return;

}

swapStudents(&students[index1 - 1], &students[index2 - 1]);

cout << "\nStudent pomenalis mestami!" << endl;

cout << "New List:" << endl;

printStudents(students, num);

}

void searchStudent(Student\* students, int num) {

char search[20];

int searchType;

cout << "\nChoice pole for search:" << endl;

cout << "1 - Po fami" << endl;

cout << "2 - Po imeni" << endl;

cout << "3 - Po facultetu" << endl;

cout << "4 - Po omeru zachetki" << endl;

cout << "Ваш выбор: ";

cin >> searchType;

cout << "Enter znachenie poiska: ";

cin >> search;

bool found = false;

cout << "\nPeruslt of search:" << endl;

cout << "==============================================" << endl;

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

bool match = false;

switch (searchType) {

case 1:

if (strcmp(students[i].famil, search) == 0) {

match = true;

}

break;

case 2:

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

match = true;

}

break;

case 3:

if (strcmp(students[i].facult, search) == 0) {

match = true;

}

break;

case 4:

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

match = true;

}

break;

default:

cout << "False choice!" << endl;

return;

}

if (match) {

cout << "Student №" << i + 1 << ":" << endl;

cout << "Surname: " << students[i].famil << endl;

cout << "Name: " << students[i].name << endl;

cout << "Facult: " << students[i].facult << endl;

cout << "Zachetka: " << students[i].Nomzach << endl;

cout << "-----------------" << endl;

found = true;

}

}

if (!found) {

cout << "Student not found" << endl;

}

}

void studentManagementMenu(Student\* students, int num) {

int choice;

do {

cout << "\n=== Menu ===" << endl;

cout << "1 - Show all students" << endl;

cout << "2 - Search students" << endl;

cout << "3 - Pomenyat mestami" << endl;

cout << "0 - exit" << endl;

cout << "Your choice: ";

cin >> choice;

switch (choice) {

case 1:

printStudents(students, num);

break;

case 2:

searchStudent(students, num);

break;

case 3:

reorderStudents(students, num);

break;

case 0:

cout << "Exit from menu..." << endl;

break;

default:

cout << "Nevernyi vibor!" << endl;

}

} while (choice != 0);

}

void task5() {

const int num\_students = 5;

setlocale(LC\_ALL, "Russian");

Student\* students = new Student[num\_students];

strcpy\_s(students[0].famil, "Ivanov");

strcpy\_s(students[0].name, "Ivan");

strcpy\_s(students[0].facult, "FVT");

students[0].Nomzach = 12345;

strcpy\_s(students[1].famil, "Petrov");

strcpy\_s(students[1].name, "Petr");

strcpy\_s(students[1].facult, "Math");

students[1].Nomzach = 12346;

strcpy\_s(students[2].famil, "Sidorov");

strcpy\_s(students[2].name, "Alex");

strcpy\_s(students[2].facult, "Fizika");

students[2].Nomzach = 12347;

strcpy\_s(students[3].famil, "Kuznetsova");

strcpy\_s(students[3].name, "Mari");

strcpy\_s(students[3].facult, "FVT");

students[3].Nomzach = 12348;

strcpy\_s(students[4].famil, "Smirnov");

strcpy\_s(students[4].name, "Dmitriy");

strcpy\_s(students[4].facult, "Math");

students[4].Nomzach = 12349;

studentManagementMenu(students, num\_students);

delete[] students;

}

int main() {

cout << "=== Task 1 ===" << endl;

task1();

cout << "\n=== Task 2 ===" << endl;

task2();

cout << "\n=== Task 3 ===" << endl;

task3();

cout << "\n=== Task 4 ===" << endl;

task4();

cout << "\n=== Task 5 ===" << endl;

task5();

return 0;

}

**Результат работы программ**

Результаты работы программы показаны на рисунке 1.

Изображение выглядит как текст, снимок экрана, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 1 — Результаты работы программы**

Результат второй работы программы показан на рисунке 2Изображение выглядит как текст, снимок экрана, меню, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 2 — Результаты работы программы**

**Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Рисунок 3 — Результаты работы программы**

**Протокол трассировки программы**

На рисунке 3 показан протокол трассировки обработки вводимых значений

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 4 — Протокол трассировки**

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

**Рисунок 5 — Требования дополнительного задания**

**Вывод:**

В ходе выполнения лабораторной работы разработали программу, вычисляющую разницу между максимальным и минимальным элементами массива, реализующую инициализацию массива случайными числами, реализующую создание массива произвольного размера, вводимого с клавиатуры, вычисляющую сумму значений в каждом столбце (строке) двумерного массива, осуществляющую поиск среди структур student структуру с нужными параметрами (фамилией, именем, факультетом и номером зачётной книжки).

Задание: выполнили поиск по подстроке, добавили сохранение матриц, добавили ввод букв в графе зачетная книжка.