Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

По Логике и основам алгоритмизации в инженерных задачах

По лабораторной работе №1

На тему «Простые структуры данных»

Выполнили

Студенты группы 24ВВВ3:

Осипов Е.А.

Давыдов А.Р.

Приняли:

Юрова О.В.

Деев М.В.

Пенза 2024

**Цель работы:** вспомнить алгоритмы работы с простыми структурами данных.

**Ход работы:**

1. Листинг программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <locale.h>

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <limits.h>

#include <string.h>

#include <ctype.h>

struct Student {

char name[50];

char surname[50];

int course;

};

void createStudent(struct Student\* s) {

printf("Введите имя студента: ");

scanf("%s", s->name);

printf("Введите фамилию студента: ");

scanf("%s", s->surname);

printf("Введите курс студента: ");

scanf("%d", &s->course);

}

void printStudent(const struct Student\* s) {

printf("Студент: %s %s, курс: %d\n", s->name, s->surname, s->course);

}

void toLowerCase(char\* str) {

for (int i = 0; str[i]; i++) {

str[i] = tolower(str[i]);

}

}

void findStudentByNamePartial(const struct Student students[], int size, const char\* searchTerm) {

printf("\nРезультаты поиска по части имени/фамилии '%s':\n", searchTerm);

int found = 0;

char searchLower[50];

strcpy(searchLower, searchTerm);

toLowerCase(searchLower);

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

char nameLower[50], surnameLower[50];

strcpy(nameLower, students[i].name);

strcpy(surnameLower, students[i].surname);

toLowerCase(nameLower);

toLowerCase(surnameLower);

if (strstr(nameLower, searchLower) != NULL || strstr(surnameLower, searchLower) != NULL) {

printStudent(&students[i]);

found = 1;

}

}

if (!found) {

printf("Студенты по запросу '%s' не найдены\n", searchTerm);

}

}

void findStudentByName(const struct Student students[], int size, const char\* name) {

printf("\nРезультаты поиска по точному имени '%s':\n", name);

int found = 0;

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

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

printStudent(&students[i]);

found = 1;

}

}

if (!found) {

printf("Студент с именем '%s' не найден\n", name);

}

}

void findStudentBySurname(const struct Student students[], int size, const char\* surname) {

printf("\nРезультаты поиска по точной фамилии '%s':\n", surname);

int found = 0;

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

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

printStudent(&students[i]);

found = 1;

}

}

if (!found) {

printf("Студент с фамилией '%s' не найден\n", surname);

}

}

void findStudentByCourse(const struct Student students[], int size, int course) {

printf("\nРезультаты поиска по курсу %d:\n", course);

int found = 0;

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

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

printStudent(&students[i]);

found = 1;

}

}

if (!found) {

printf("Студенты на курсе %d не найдены\n", course);

}

}

void searchMenu(const struct Student students[], int size) {

int choice;

char searchTerm[50];

char searchSurname[50];

int searchCourse;

do {

printf("\n=== МЕНЮ ПОИСКА СТУДЕНТОВ ===\n");

printf("1. Поиск по части имени/фамилии (неполный)\n");

printf("2. Поиск по точному имени\n");

printf("3. Поиск по точной фамилии\n");

printf("4. Поиск по курсу\n");

printf("5. Вернуться в главное меню\n");

printf("Выберите вариант: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Введите часть имени или фамилии для поиска: ");

scanf("%s", searchTerm);

findStudentByNamePartial(students, size, searchTerm);

break;

case 2:

printf("Введите точное имя для поиска: ");

scanf("%s", searchTerm);

findStudentByName(students, size, searchTerm);

break;

case 3:

printf("Введите точную фамилию для поиска: ");

scanf("%s", searchSurname);

findStudentBySurname(students, size, searchSurname);

break;

case 4:

printf("Введите курс для поиска: ");

scanf("%d", &searchCourse);

findStudentByCourse(students, size, searchCourse);

break;

case 5:

printf("Возврат в главное меню...\n");

break;

default:

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

break;

}

} while (choice != 5);

}

int main() {

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int array[10];

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

array[i] = rand() % 100;

}

printf("Первый массив:\n");

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

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

}

printf("\n");

printf("Разница между максимальным и минимальным элементом: ");

int min = INT\_MAX;

int max = 0;

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

if (array[i] > max) max = array[i];

}

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

if (array[i] < min) min = array[i];

}

int res = max - min;

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

int\* arrayP;

int size;

printf("Введите размер массива:\n");

scanf("%d", &size);

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

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

arrayP[i] = rand() % 100;

}

printf("Массив заданного размера:\n");

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

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

}

free(arrayP);

printf("\nРандомная матрица:\n");

int matrix[6][6];

int rowsize[6] = { 0 };

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

matrix[i][j] = rand() % 100;

printf("%3d ", matrix[i][j]);

}

printf("\n");

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

}

printf("Сумма каждой строки: \n");

for (int i = 0; i < 6; i++) {

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

}

printf("\n");

printf("\n=== РАБОТА СО СТРУКТУРОЙ STUDENT ===\n");

struct Student students[3];

printf("\nВведите данные для 3 студентов:\n");

for (int i = 0; i < 3; i++) {

printf("\nСтудент %d:\n", i + 1);

createStudent(&students[i]);

}

printf("\nВсе студенты:\n");

for (int i = 0; i < 3; i++) {

printStudent(&students[i]);

}

searchMenu(students, 3);

printf("\nВведите количество строк:");

int rows;

scanf("%d", &rows);

printf("\nВведите количество столбцов:");

int cols;

scanf("%d", &cols);

int\*\* matr = (int\*\*)malloc(rows \* sizeof(int\*));

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

{

matr[i] = (int\*)malloc(cols \* sizeof(int\*));

}

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

{

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

{

matr[i][j] = rand() % (21 - 10);

}

}

int count = 0;

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

{

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

{

if (matr[i][j] == 0) count++;

}

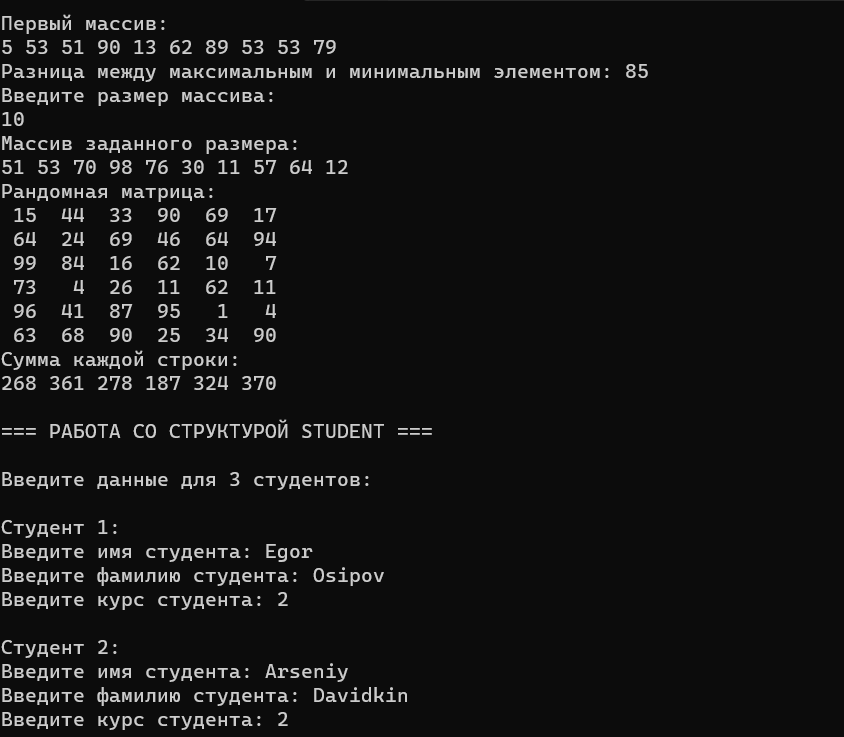
}

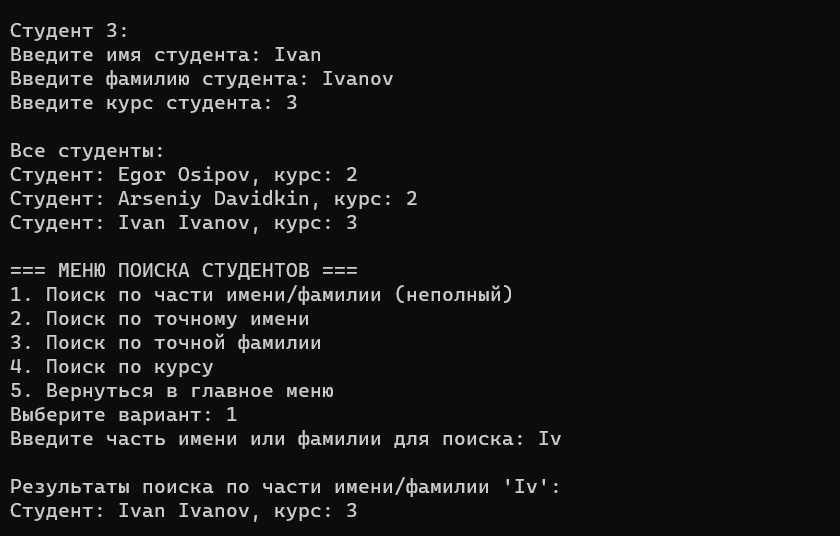
printf("\nКоличество нулевых элементов: %d", count);

return 0;

}

1. Результат выполнения программы:





**Вывод:** в ходе данной лабораторной работы были повторены алгоритмы для работы с простыми структурами данных, алгоритмы генерации случайных чисел и работа с памятью.