Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра прикладной математики и кибернетики

Отчет по лабораторной работе

по дисциплине «Операционные системы»

Лабораторная работа №1

Выполнил: студент 3 курса группы ИП-811 Мироненко К. А.

Проверил: профессор кафедры ПМиК Малков Е. А.

**ОГЛАВЛЕНИЕ**

[1. Постановка задачи 3](#_Toc53520694)

[2. Примеры работы программы 6](#_Toc53520696)

[*Приложение* Листинг 7](#_Toc53520697)

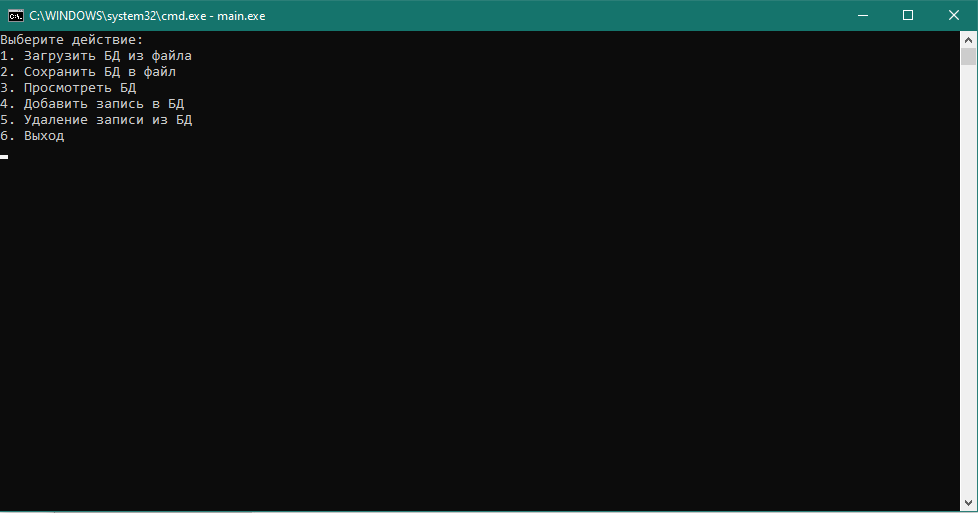
# Постановка задачи

**Цель**: проверка знаний, необходимых для усвоения курса Операционные системы.

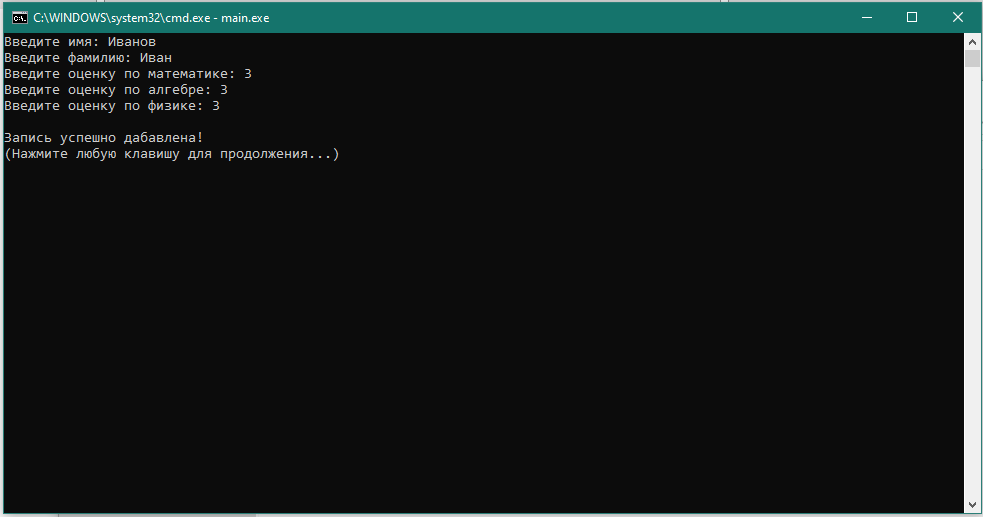
**Задание**: написать программу для манипуляции данными на основе рекурсивных структур, реализовать функции вставки, удаления и навигации для списка; реализовать сериализацию списка. (Например: электронный журнал успеваемости студентов.)

**Инструментарий**: любая среда разработки C/C++..

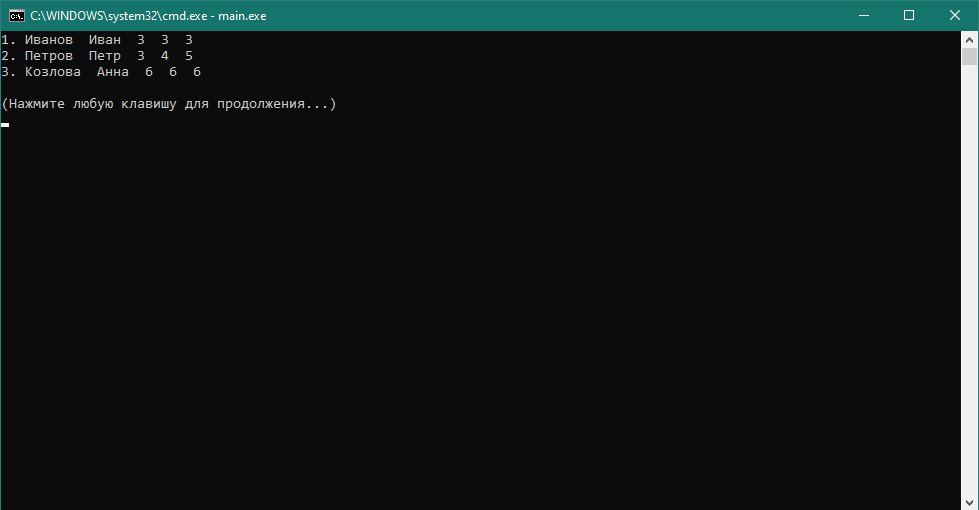
# Примеры работы программы



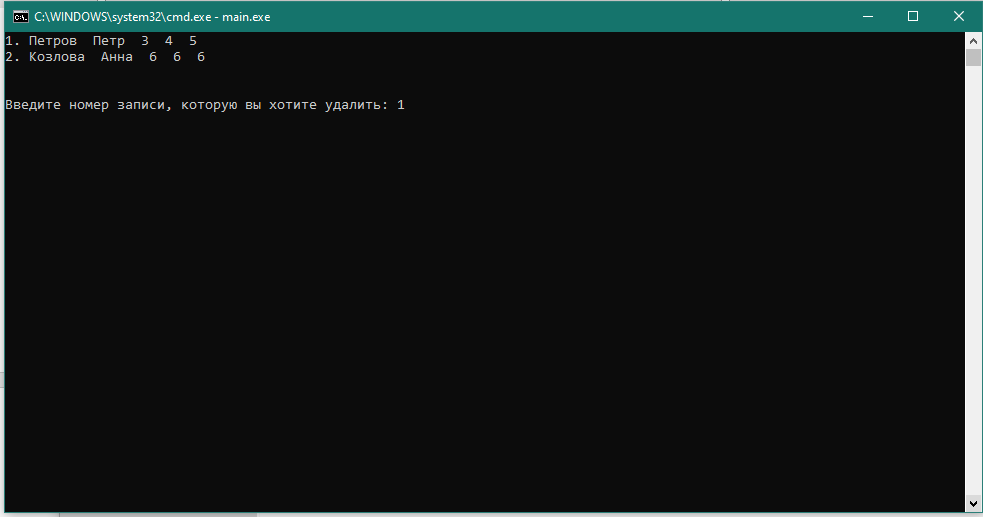
(Меню программы)



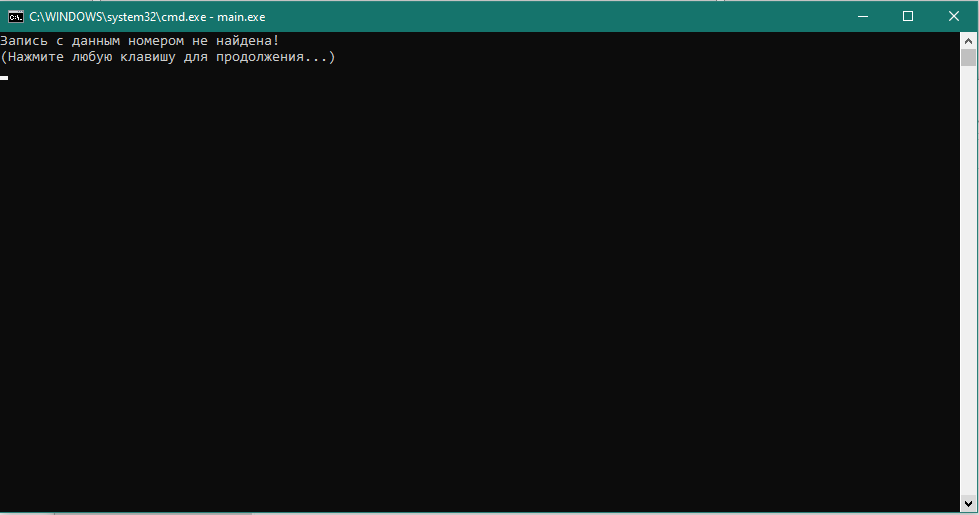
(Добавление записи)



(Вывод базы)



(Удаление записи)



(Удаление, при вводе некорректного значения)

# Приложение. Листинг

**main.c**

#include <windows.h>

#include <stdio.h>

#include <locale.h>

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "");

char systemPath[MAX\_PATH], tempPath[MAX\_PATH];

SYSTEM\_INFO systemInformation;

MEMORYSTATUSEX memory;

memory.dwLength = sizeof(memory);

GetWindowsDirectory(systemPath, MAX\_PATH);

GetTempPath(MAX\_PATH, tempPath);

GetSystemInfo(&systemInformation);

GlobalMemoryStatusEx(&memory);

printf("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("\n Системный каталог Windows: %s", systemPath);

printf("\n Каталог временных файлов Windows: %s", tempPath);

printf("\n\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("\n Количество ядер процессора: %d", systemInformation.dwNumberOfProcessors);

printf("\n Архитектура процессора: ");

switch (systemInformation.wProcessorArchitecture)

{

case 0:

printf("x86");

break;

case 9:

printf("x64");

break;

default:

printf("неизвестно");

break;

}

printf("\n\n Нижняя граница пользовательского адресного пространства: %p", systemInformation.lpMinimumApplicationAddress);

printf("\n Верхняя граница пользовательского адресного пространства: %p", systemInformation.lpMaximumApplicationAddress);

printf("\n\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("\n Физическая память:");

printf("\n Всего: %llu Мб", memory.ullTotalPhys / 1048576);

printf("\n Доступно: %llu Мб", memory.ullAvailPhys / 1048576);

printf("\n Используется: %llu Мб", (memory.ullTotalPhys - memory.ullAvailPhys) / 1048576);

printf("\n\n Виртуальная память:");

printf("\n Всего: %llu Мб", memory.ullTotalVirtual / 1048576);

printf("\n Доступно: %llu Мб", memory.ullAvailVirtual / 1048576);

printf("\n Используется: %llu Мб", (memory.ullTotalVirtual - memory.ullAvailVirtual) / 1048576);

printf("\n\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

return 0;

}