# НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №*8*

із дисципліни «Програмування»

на тему

*«Робота з бінарними та текстовими*

*файлами»*

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *асистент Громова В. В.* |
| *Артеменко Я.К.* |  |

Київ — 2017

# **ЗМІСТ**

[1 ПОСТАНОВКА ЗАДАЧІ 2](#_Toc478382182)

[1.1 Мета роботи 2](#_Toc478382183)

[1.2 Завдання на лабораторну роботу 2](#_Toc478382184)

[2 ОПИС ПРОГРАМИ 4](#_Toc478382185)

[ВИСНОВКИ 6](#_Toc478382186)

[Додаток А (блок-схема) 6](#_Toc478382187)

[Додаток Б (відповіді на контрольні запитання) 8](#_Toc478382188)

[Додаток В (текст програми) 9](#_Toc478382189)

# **1 ПОСТАНОВКА ЗАДАЧІ**

* 1. Мета роботи

Вивчити файлові типи даних, набути практичних навичок створення й опрацювання текстових і бінарних файлів.

* 1. Завдання на лабораторну роботу

Варіант №3

Робота з текстовими файлами:

1. Скопіювати у файл F2 рядки з F1, що починаються й закінчуються на одну й ту саму літеру.
2. Підрахувати кількість символів в F2.

Робота з бінарними файлами:

1. Структура «Покупець»:

* Прізвище, ім’я, по батькові;
* Домашня адреса;
* Номер телефону;
* Номер кредитної картки;
* Статус запису;

1. Видалити три елементи з початку файлу.
2. Додати три елементи в кінець файлу.

**Постановка завдання**

Завдання складається з двох частин:

* програмування роботи з текстовим доступом до файлу;
* програмування роботи з бінарним доступом до файлу.

Для обох програм повинно бути підготовлено вихідні файли:

* для текстового файлу — не менше за 10 рядків;
* для бінарного файлу — не менше за 10 структур, що відповідають конкретному варіанту.

Для роботи з файлами повинно бути розроблено меню, пункти якого реалізовано тільки за допомогою функцій. Пункти меню повинні бути такі:

* створення нового файлу;
* відкриття файлу;
* перегляд файлу (перегортання вперед, назад, у кінець файлу, у початок файлу);
* корекція файлу — дозапис, виправлення, видалення даних;
* збереження файлу;
* збереження файлу як…;
* закриття файлу.

Меню можна доповнювати додатковими пунктами, залежно від варіанту.

Імена файлів потрібно уводити за запитом із клавіатури.

# **ОПИС ПРОГРАМИ**

Дана програма містить 23 функцій: menu\_main, table, print\_struct, input\_note, new\_file, open\_file, show\_file, add, edit, del, save\_file, save\_file\_name, close\_file, task\_1, save\_without, save\_with, add\_3,del\_3, menu, menu\_1, menu\_save, menu\_task main. Ще програма має структуру buyer з полями, вказаними в п. 1.2.

Функція **menu\_main** виводить на екран дані про виконавця та надає можливість користувачу обрати завдання для виконання. Обрати завдання можна за допомогою стрілок «вверх» і «вниз» і за допомогою клавіші «Enter» перейти до його виконання. Пункти, які ми можемо обрати підсвічені білим кольором, а умови завдань виділяються сірим кольором. За допомогою оператора розгалуження «if – else» зроблено можливість переміщення «стрілки вибору» між пунктами меню та за допомогою десяткових індексів функціональних клавіш («80» – для стрілки вниз, а «72» – для стрілки вверх) зроблено можливим використання клавіш для вибору. За допомогою циклу «while» очікується натиснення клавіші Enter, після чого ця функція повертає значення вибраного пункту в функцію main.

Функція **table** задає шапку таблиці, в яку буде записуватись структура.

Функція **print\_struct** виводить на екран структуру.

Функція **input\_struct** організовує введення структури.

Функція **new\_file** створює файл з ім’ям, яке вводиться з клавіатури.

Функція **open\_file** відкриває файл з ім’ям, яке вводиться з клавіатури.

Функція **show\_file** показує вміст відкритого файлу.

Функція **add** додає у файл запис в залежності від того, як відкритий файл: текстовий чи бінарний.

Функція **edit** організовує заміну рядка за номером, який вводиться з клавіатури.

Функція **del** організовує видалення запису чи рядка з файлу за номером, введеним з клавіатури.

Функція **save\_file** зберігає текстовий файл файл.

Функція **save\_file\_name** забезпечує зміну імені відкритого файлу.

Функція **close\_file** організовує закриття файлу.

Функція **task\_1** виконує перше завдання.

Функція **save\_without** організовує збереження бінарного файлу без ущільнення.

Функція **save\_with** організовує збереження бінарного файлу з ущільненням.

Функція **add\_3** додає у кінець файлу три записа.

Функція **del\_3** видаляє з бінарного файлу три записа з початку.

Функція **menu** надає можливість вибирати дію, яку виконати з файлом.

Функція **menu\_1** надає можливість обрати метод корекції файлу.

Функція **menu\_save** надає можливість обрати метод збереження бінарного файлу.

Функція **menu\_task** надає можливість обрати підзадачу з другої задачі.

Функція **main** є головною і в ній викликаються всі функції і забезпечується робота меню за допомогу оператора «case».

# **ВИСНОВКИ**

На цій лабораторній роботі здобуто навички по роботі з бінарними та текстовими файлами та організовано пункти для роботи з ними.

# **Додаток А (блок-схема)**

Функція menu\_main:

Початок

Printf(“Умова завдання”)

Printf(“Інформація про виконавця”)

System(“cls”);

ні

If key = {0}

так

Cout<<”-> Продовжити”

Cout<<” Продовжити”

If key = {1}

ні

так

Cout<<”-> Вихід”

Cout<<” Вихід”

Кінець

Функція input\_note:

Початок

while (c != 'y')

pot->uah[0] = valid\_bit(10000, 1);

scanf("%s", &pot->name\_product);

printf("Введите название магазина(Вводите текст): ");

scanf("%s", &pot->name\_shop);

printf("Введите название продукта(Вводите текст): ");

cout << "Введите к-во гривен: ";

cout << "Введите к-во копеек: ";

pot->uah[1] = valid\_bit(10000, 1);

# 

Кінець

# **Додаток Б (відповіді на контрольні запитання)**

1. Тестові файли складаються з послідовності символів, розбитих на рядки.

Двійкові(бінарні) складаються з послідовності байтів між якими немає розділювачів.

1. Відкриття відбувається заданням режиму відкриття, а закриття виконує закриття файлу.
2. Fgets, fputs.
3. Кінець файлу за допомогою feof, а кінець рядка sizeof.
4. Так
5. Для зчитування, запису.
6. Зчитування, запис.
7. fseek надає можливість переміщувати вказівник поточної позиції на потрібний байт.

Ftell повертає поточну позицію у файл.

1. Бінарний зберігається у вигляді послідовності байтів без розділювача.
2. Fread i fwrite
3. Бінарні
4. Rename, remove.
5. Ferror – помилка, feof – кінець файлу.
6. -

# **Додаток В (текст програми)**

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <stdlib.h>

#include <cstdlib>

#include <iostream>

#include <locale.h>

#include <string>

#include <iomanip>

#include <cstring>

#include <cstdio>

#include <algorithm>

using namespace std;

int menu\_task();

int menu\_save();

int menu\_1();

int main();

char ch;

int task;

FILE\* F;

char\* str;

char filename[21] = { 0 };

struct buyer

{

char name[15];

char surname[15];

char father[15];

char house[30];

char number[15];

char card[16];

bool status;

};

void table()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" #| Фамилия Имя Отчество | Адрес | Номер | Номер карты | Статус |\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

void print\_struct(buyer number)

{

printf(" |%15s %15s %15s|%30s|%15s|%16s|%8i|\n", number.surname, number.name, number.father, number.house, number.number, number.card, number.status);

}

struct buyer input\_note()

{

struct buyer number;

printf("Введите фамилию (Вводите только буквы!): ");

cin >> number.surname;

printf("Введите имя (Вводите только буквы!): ");

cin >> number.name;

printf("Введите отчество (Вводите только буквы!): ");

cin >> number.father;

printf("Введите адрес: ");

cin >> number.house;

cin.clear();

printf("Введите номер телефона (Вводите только цифры!): ");

cin >> number.number;

cin.clear();

printf("Введите номер карты (Вводите только цифры!): ");

cin >> number.card;

number.status = FALSE;

return number;

}

int menu\_main()

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

char key = 0;

int code;

do

{

system("cls");

printf("\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t\t\t \* Лабораторная работа №7 \*\n");

printf("\t\t\t\t \* Работа с бинарными и \*\n");

printf("\t\t\t\t \* текстовыми файлами \*\n");

printf("\t\t\t\t \* Вариант №3 \*\n");

printf("\t\t\t\t \* Группа KM-63 Артеменко Я.К.\*\n");

printf("\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n");

key = (key + 3) % 3;

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 8));

printf("\t\t\t\tИНСТРУКЦИЯ!!! ДЛЯ НАВИГАЦИИ ПО МЕНЮ ИСПОЛЬЗУЙТЕ СТРЕЛКИ ВВЕРХ, ВНИЗ И КЛАВИШУ ENTER\n");

printf("\t\t\t\t1.1 Скопировать у файл F2 строки с F1, что начинаются и заканчиваются\n");

printf("\t\t\t\t на одну и ту самую букву. \n");

printf("\t\t\t\t1.2. Посчитать количество символов в F2.\n\n");

printf("\t\t\t\t2. Структура <<Покупатель>>: \n");

printf("\t\t\t\t - фамилия; \n");

printf("\t\t\t\t - имя; \n");

printf("\t\t\t\t - отчество; \n");

printf("\t\t\t\t - домашний адрес; \n");

printf("\t\t\t\t - номер телефона \n");

printf("\t\t\t\t - номер кредитки; \n");

printf("\t\t\t\t - статус записи; \n\n\n");

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 15));

if (key == 0) cout << "\t\t\t\t -> Работа с текстовым файлом \n" << endl;

else cout << "\t\t\t\t Работа с текстовым файлом\n" << endl;

if (key == 1) cout << "\t\t\t\t -> Работа с бинарным файлом \n\n\n\n" << endl;

else cout << "\t\t\t\t Работа с бинарным файлом\n\n\n\n" << endl;

if (key == 2) cout << "\t\t\t\t -> Выход\n" << endl;

else cout << "\t\t\t\t Выход.\n" << endl;

code = \_getch();

if (code == 224)

{

code = \_getch();

if (code == 80) key++;

if (code == 72) key--;

}

} while (code != 13);

system("cls");

return key;

}

void new\_file()

{

cout << "Введите имя файла: ";

//char filename[21] = { 0 };

cin >> filename;

FILE\* file = fopen(filename, "w");

cout << (file != NULL ? "Файл создан\n" : "Файл не создан\n");

fclose(file);

}

void open\_file()

{

cout << "Введите имя файла: ";

//char filename[21] = { 0 };

cin >> filename;

F = fopen(filename, (task ? "rb+" : "r+"));

cout << (F != NULL ? "Файл открыт\n" : "Файл не открыт\n");

}

void show\_file()

{

if (F)

{

fseek(F, 0, SEEK\_SET);

if (task == 0)

{

while (1)

{

char t[256];

fgets(t, 255, F);

if (feof(F))

break;

cout << t;

}

}

else

{

table();

int i = 1;

while (1)

{

buyer t;

fread(&t, sizeof(buyer), 1, F);

if (feof(F))

break;

cout << i++;

print\_struct(t);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

}

}

else

cout << "Файл не открыт!!" << endl;

cout << endl;

}

void add()

{

rewind(stdin);

if (task == 0)

{

char str\_file[255] = { 0 };

cin.getline(str\_file, 255);

fseek(F, 0, SEEK\_END);

if (F)

fprintf\_s(F, "\n%s", str\_file);

else

cout << "Файл не открыт!!" << endl;

}

else

{

fseek(F, 0, SEEK\_END);

buyer t = input\_note();

fwrite(&t, sizeof(buyer), 1, F);

}

}

void edit()

{

int n,i;

char t[256];

struct buyer tb;

cout << "Введите номер: ";

cin >> n;

fseek(F, 0, SEEK\_SET);

FILE\* TEMP = fopen("temp.txt", "w");

if (task == 0)

{

for (i = 1; i < n; i++)

{

if (!feof(F))

{

fgets(t, 255, F);

fprintf\_s(TEMP, "%s", t);

}

else

{

cout << "Такой строки нету!" << endl;

break;

}

}

if (i == n)

{

fgets(t, 255, F);

cout << "Текущая строка: " << t << endl;;

cout << "Заменить на: ";

rewind(stdin);

cin.getline(t, 255);

fprintf\_s(TEMP, "%s\n", t);

}

while (!feof(F))

{

fgets(t, 255, F);

fprintf\_s(TEMP, "%s", t);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

else

{

for (i = 1; i < n; i++)

{

if (!feof(F))

{

fread(&tb, sizeof(buyer), 1, F);

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

else

{

cout << "Такой строки нету!" << endl;

break;

}

}

if (i == n)

{

fread(&tb, sizeof(buyer), 1, F);

cout << "Текущая запись: \n";

print\_struct(tb);

cout << "Заменить на: \n";

//cin.getline(t, 255);

tb=input\_note();

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

while (1)

{

fread(&tb, sizeof(buyer), 1, F);

if (feof(F))

break;

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

}

void del()

{

int n;

char t[256];

struct buyer tb;

cout << "Введите номер: ";

cin >> n;

fseek(F, 0, SEEK\_SET);

FILE\* TEMP = fopen("temp.txt", "w");

if (task == 0)

{

for (int i = 1; i < n; i++)

{

if (!feof(F))

{

fgets(t, 255, F);

fprintf\_s(TEMP, "%s", t);

}

else

{

cout << "Такой строки нету!" << endl;

break;

}

}

fgets(t, 255, F);

while (!feof(F))

{

fgets(t, 255, F);

fprintf\_s(TEMP, "%s", t);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

else

{

for (int i = 1; i < n; i++)

{

if (!feof(F))

{

fread(&tb,sizeof(buyer),1, F);

fwrite(&tb,sizeof(buyer),1, TEMP);

}

else

{

cout << "Такой строки нету!" << endl;

break;

}

}

fread(&tb, sizeof(buyer), 1, F);

tb.status = TRUE;

fwrite(&tb, sizeof(buyer), 1, TEMP);

while (1)

{

fread(&tb, sizeof(buyer), 1, F);

if (feof(F))

break;

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

}

void save\_file()

{

cout << "Файл успешно сохранен." << endl;

}

void save\_file\_name()

{

cout << "Изменить название файла: " << filename << endl;

cout << "На имя файла: ";

char filename\_1[21] = { 0 };

cin >> filename\_1;

fclose(F);

rename(filename, filename\_1);

strcpy(filename, filename\_1);

F = fopen(filename, (task ? "rb+" : "r+"));

}

void close\_file()

{

fclose(F);

F = NULL;

}

void task\_1()

{

FILE\* F2 = fopen("f2.txt", "w+");

fseek(F, 0, SEEK\_SET);

char t[256];

int count = 0;

while (!feof(F))

{

fgets(t, 255, F);

int index;

if (t[strlen(t) - 1] == '\n')

index = strlen(t) - 2;

else

index = strlen(t) - 1;

if (t[0] == t[index] )

{

fprintf\_s(F2, "%s", t);

count += (index+1);

}

}

fprintf\_s(F2, "\nКоличество символов: %i", count);

fclose(F2);

}

void save\_without()

{

cout << "Файл успешно сохранен";

}

void save\_with()

{

FILE\* TEMP = fopen("temp.txt", "w");

fseek(F, 0, SEEK\_SET);

struct buyer tb;

while (1)

{

fread(&tb, sizeof(buyer), 1, F);

if (feof(F))

break;

if (tb.status==FALSE)

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

void add\_3()

{

rewind(stdin);

for (int i = 1; i <= 3; i++)

{

cout << "Введите " << i << " запись:" << endl;

fseek(F, 0, SEEK\_END);

buyer t = input\_note();

fwrite(&t, sizeof(buyer), 1, F);

}

}

void del\_3()

{

struct buyer tb;

FILE\* TEMP = fopen("temp.txt", "w");

fseek(F, 0, SEEK\_SET);

for (int i = 1; i <= 3; i++)

{

fread(&tb, sizeof(buyer), 1, F);

tb.status = TRUE;

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

while (1)

{

fread(&tb, sizeof(buyer), 1, F);

if (feof(F))

break;

fwrite(&tb, sizeof(buyer), 1, TEMP);

}

fclose(F);

fclose(TEMP);

remove(filename);

rename("temp.txt", filename);

F = fopen(filename, (task ? "rb+" : "r+"));

}

int menu()

{

while (1)

{

while (1)

{

system("cls");

printf("Выберите один из вариантов:\n\n");

printf("1. Создать новый файл\n");

printf("2. Открыть файл\n");

printf("3. Просмотр файла\n");

printf("4. Коррекция файла\n");

printf("5. Сохранение файла\n");

printf("6. Сохранить файл как...\n");

printf("7. Закрыть файл\n");

printf("8. Выполнить задачу\n");

printf("9. Вернуться в предыдущее меню\n\n");

printf("Введите номер пунка:");

ch = \_getch();

printf("\n");

if (ch == '1' || ch == '2' || ch == '3' || ch == '4' || ch == '5' || ch == '6' || ch == '7' || ch == '8' || ch=='9')

break;

system("cls");

printf("Вы ввели неправильный номер.\n\n");

}

switch (ch)

{

case '1':

{

system("cls");

new\_file();

system("pause");

break;

}

case '2':

{

system("cls");

open\_file();

system("pause");

break;

}

case '3':

{

system("cls");

show\_file();

system("pause");

break;

}

case '4':

system("cls");

if (F)

menu\_1();

else

cout << "Файл не открыт!!" << endl;

system("pause");

break;

case '5':

{

system("cls");

if (F)

{

if (task == 0)

save\_file();

else

menu\_save();

}

else

cout << "Файл не открыт!!" << endl;

system("pause");

break;

}

case '6':

{

system("cls");

if (F)

{

save\_file\_name();

}

else

cout << "Файл не открыт!!" << endl;

system("pause");

break;

}

case '7':

{

system("cls");

if (F)

close\_file();

else

cout << "Файл не открыт!!" << endl;

system("pause");

break;

}

case '8':

{

system("cls");

if (F)

{

if (task == 0)

task\_1();

else

menu\_task();

}

else

cout << "Файл не открыт!!" << endl;

system("pause");

break;

}

case '9': {main(); break; }

}

}

\_getch();

}

int menu\_1()

{

while (1)

{

while (1)

{

system("cls");

printf("Выберите один из вариантов:\n\n");

printf("1. Дописать. \n");

printf("2. Исправить.\n");

printf("3. Удалить данные\n");

printf("4. Вернутся в предыдущее меню.\n\n");

printf("Введите номер пунка:");

ch = \_getch();

printf("\n");

if (ch == '1' || ch == '2' || ch == '3' || ch == '4')

break;

system("cls");

printf("Вы ввели неправильный номер.\n\n");

}

switch (ch)

{

case '1':

{

system("cls");

add();

system("pause");

break;

}

case '2':

{

system("cls");

edit();

system("pause");

break;

}

case '3':

{

system("cls");

del();

system("pause");

break;

}

case '4': {menu(); break; }

}

}

\_getch();

}

int menu\_save()

{

while (1)

{

while (1)

{

system("cls");

printf("Выберите один из вариантов:\n\n");

printf("1. Сохранить без уплотнения. \n");

printf("2. Сохранить с уплотнением.\n");

printf("3. Вернутся в предыдущее меню.\n\n");

printf("Введите номер пунка:");

ch = \_getch();

printf("\n");

if (ch == '1' || ch == '2' || ch == '3')

break;

system("cls");

printf("Вы ввели неправильный номер.\n\n");

}

switch (ch)

{

case '1':

{

system("cls");

save\_without();

system("pause");

break;

}

case '2':

{

system("cls");

save\_with();

system("pause");

break;

}

case '3': {menu(); break; }

}

}

\_getch();

}

int menu\_task()

{

while (1)

{

while (1)

{

system("cls");

printf("Выберите один из вариантов:\n\n");

printf("1. Добавить 3 элемента в конец файла. \n");

printf("2. Удалить три элемента с начала файла.\n");

printf("3. Вернутся в предыдущее меню.\n\n");

printf("Введите номер пунка:");

ch = \_getch();

printf("\n");

if (ch == '1' || ch == '2' || ch == '3')

break;

system("cls");

printf("Вы ввели неправильный номер.\n\n");

}

switch (ch)

{

case '1':

{

system("cls");

add\_3();

system("pause");

break;

}

case '2':

{

system("cls");

del\_3();

system("pause");

break;

}

case '3': {menu(); break; }

}

}

\_getch();

}

int main()

{

while (1)

{

setlocale(LC\_ALL, "RUS");

task = menu\_main();

if (task == 2)

exit(0);

menu();

}

}