**6 ЛАБОРАТОРНАЯ РАБОТА №6**

**«ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД СТРУКТУРАМИ**

**И БИНАРНЫМИ ФАЙЛАМИ»**

**6.1 Цель работы**

Изучение способов описания структур данных на языке С. Исследование

особенностей обработки бинарных файлов, хранящих структурные типы данных.

**6.2 Вариант задания – 20**

Требуется описать структуру с именем ORDER, содержащую следующие поля:

- расчетный счет плательщика;

- расчетный счет получателя;

- перечисляемая сумма в грн.

Написать программу, выполняющую следующие действия с помощью функций:

- ввод с клавиатуры данных в файл, состоящий из элементов типа ORDER; записи должны быть размещены в алфавитном порядке по расчетным счетам плательщиков;

- чтение данных из этого файла;

- вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры; если такого расчетного счета нет, выдать на дисплей соответствующее сообщение

**6.3 Порядок выполнения работы**

6.3.1 Разработать алгоритм решения задачи.

6.3.2 Разработать программу на языке Си.

6.3.3 Разработать тестовые примеры. Выполнить тестирование используя разработанные тестовые примеры. Выполнить отладку.

6.3.4 Сделать выводы по проделанной работе.

**6.4 Ход работы**

6.4.1 Для задания был разработан алгоритм решения задачи и отражен на структурных схемах, которые представлены на рисунках 6.1 и 6.2.

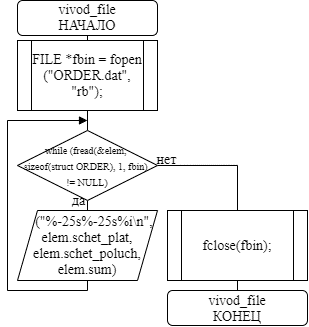
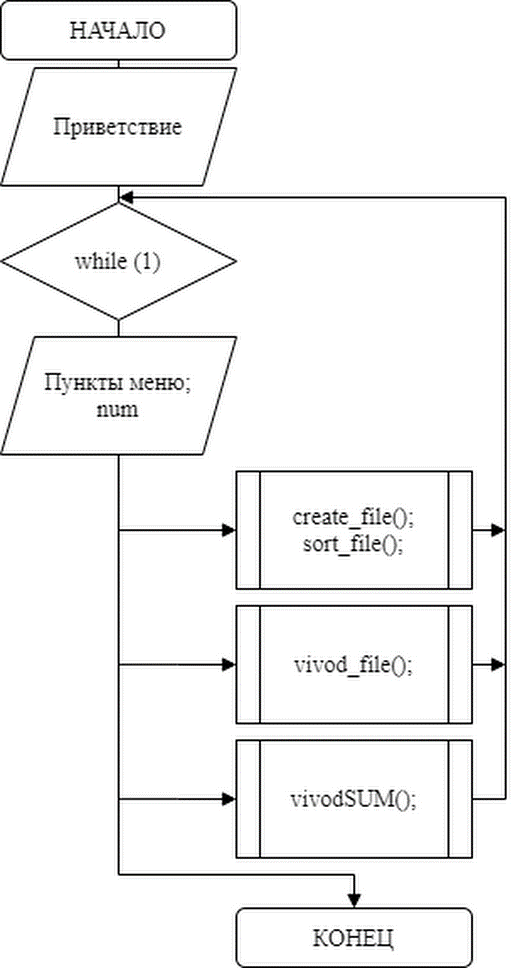
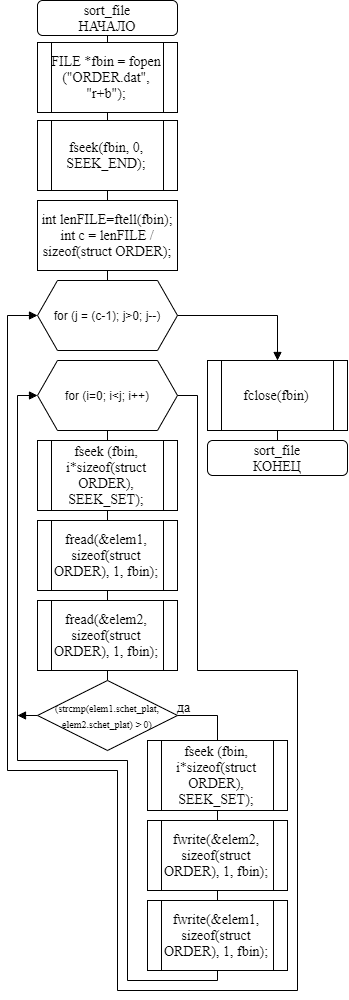


Рисунок 6.1 – Структурная схема основной программы, функции сортировки и функции вывода структуры

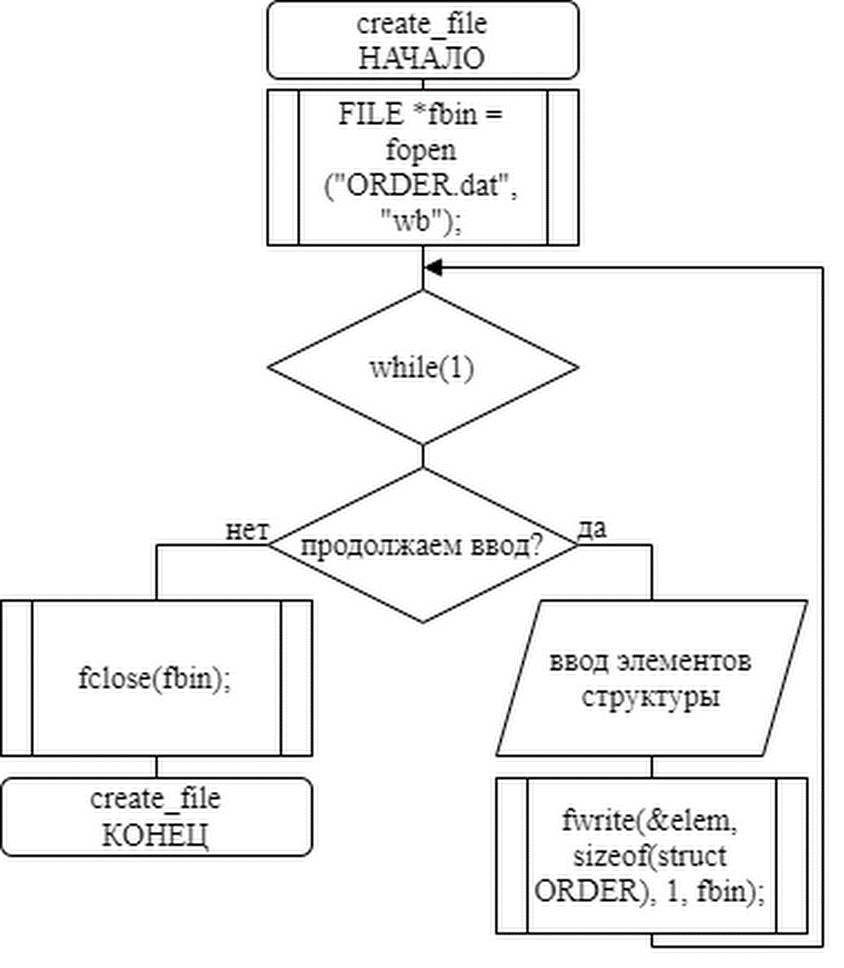
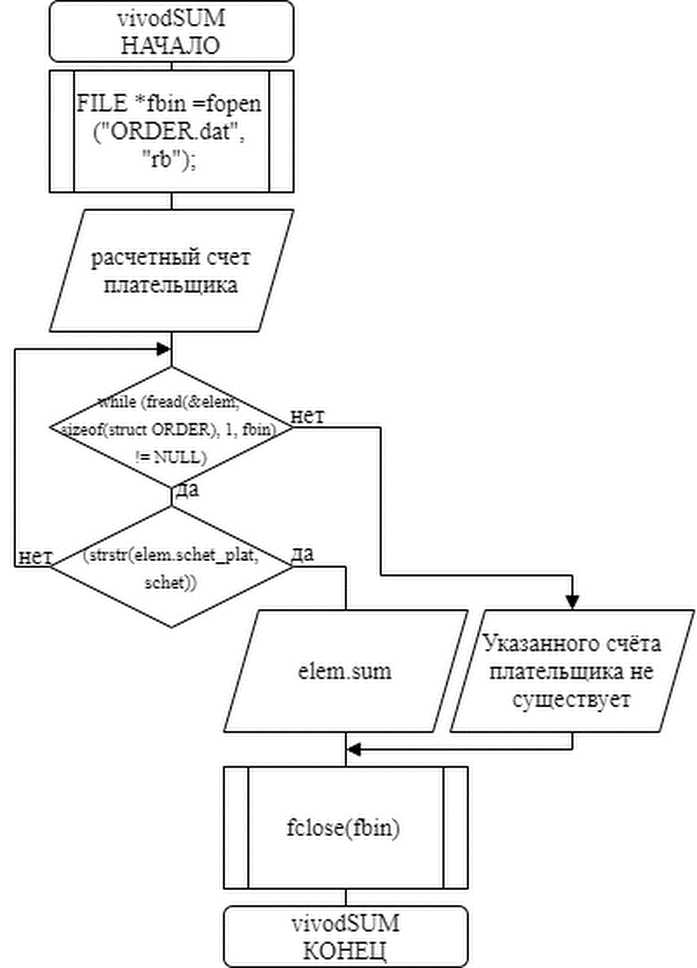


Рисунок 6.1 – Структурная схема функции ввода структур и функции вывода суммы снятой с расчетного счета плательщика, введенного с клавиатуры

6.4.2 Написана программа на Си согласно вышеописанного алгоритма.

#include <stdio.h>

#include <locale.h>

//константы и структуры

struct ORDER

{

char schet\_plat[21]; //(счет плательщика)

char schet\_poluch[21]; //(счет получателя)

int sum; //(сумма перевода (грн.))

};

//прототипы функций

int create\_file();

int vivod\_file();

int sort\_file();

int vivodSUM();

int main()

{

setlocale(LC\_ALL, "Rus");

printf(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \n");

printf("/ \n");

printf(" Программа работает со структурой состоящей из \n");

printf(" счёта плательщика, получателя и суммы перевода. \n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \n\n");

short num;

while (1)

{

printf("Выберете дальнейшее действие: \n");

printf("1 - ввод данных \n");

printf("2 - вывод данных \n");

printf("3 - сумма снятая с расчетного счёта плательщика; \n");

printf("4 - выход из программы \n");

printf(" -> "); scanf("%d",&num); printf("\n");

switch (num)

{

case 1:

{

create\_file();

sort\_file();

break;

}

case 2:

{

vivod\_file();

break;

}

case 3:

{

vivodSUM();

break;

}

case 4:

{

return 0;

}

}

}

}

//запись в файл

int create\_file()

{

FILE \*fbin;

fbin = fopen ("ORDER.dat", "wb");

if (!fbin)

{

printf("\nОшибка открытия файла\n");

return 1;

}

struct ORDER elem;

short i;

printf("Осуществляется ввод даных");

while(1)

{

printf("\nПродолжаем ввод? (1-да, 2-нет): "); scanf("%i", &i);

if (i==2)

{

printf("\n");

fclose(fbin);

return 0;

}

printf("======================================\n");

printf("Введите счёт плательщика (20символов) \n");

scanf("%s", &elem.schet\_plat);

printf("Введите счёт получателя (20символов) \n");

scanf("%s", &elem.schet\_poluch);

printf("Введите перечисляемую сумму (грн.) \n");

scanf("%d", &elem.sum);

fwrite(&elem, sizeof(struct ORDER), 1, fbin);

}

}

int vivod\_file()

{

FILE \*fbin = fopen ("ORDER.dat","rb");

if (!fbin)

{

printf("\nОшибка открытия файла\n");

return 1;

}

struct ORDER elem;

printf("Счёт плательщика Счёт получателя Сумма перевода\n");

while (fread(&elem, sizeof(struct ORDER), 1, fbin) != NULL)

printf("%-25s%-25s%i\n", elem.schet\_plat, elem.schet\_poluch, elem.sum);

printf("\n");

fclose(fbin);

return 0;

}

int sort\_file()

{

FILE \*fbin = fopen("ORDER.dat", "r+b");

struct ORDER elem1, elem2;

//размер файла;

fseek(fbin, 0, SEEK\_END);

int lenFILE = ftell(fbin);

int c = lenFILE / sizeof(struct ORDER);

int i,j;

for (j = (c-1); j>0; j--)

for (i=0; i<j; i++)

{

fseek (fbin, i\*sizeof(struct ORDER), SEEK\_SET);

fread(&elem1, sizeof(struct ORDER), 1, fbin);

fread(&elem2, sizeof(struct ORDER), 1, fbin);

if (strcmp(elem1.schet\_plat, elem2.schet\_plat) > 0)

{

fseek (fbin, i\*sizeof(struct ORDER), SEEK\_SET);

fwrite(&elem2, sizeof(struct ORDER), 1, fbin);

fwrite(&elem1, sizeof(struct ORDER), 1, fbin);

}

}

fclose(fbin);

}

int vivodSUM()

{

FILE \*fbin = fopen ("ORDER.dat","rb");

if (!fbin)

{

printf("\nОшибка открытия файла\n");

return 1;

}

struct ORDER elem;

char schet[21];

printf("Введите расчетный счет плательщика, для вывода суммы его платежа:\n");

scanf("%s",&schet); printf("\n");

while (fread(&elem, sizeof(struct ORDER), 1, fbin) != NULL)

if (strstr(elem.schet\_plat, schet))

{

printf("Сумма = %i\n", elem.sum);

fclose(fbin);

return 0;

}

printf("Такого счёта плательщика не найдено :( \n\n");

fclose(fbin);

return 1;

}

6.4.3 Выполнена отладка программы.

Результаты тестирования отображены на рисунках 6.3- 6.5.

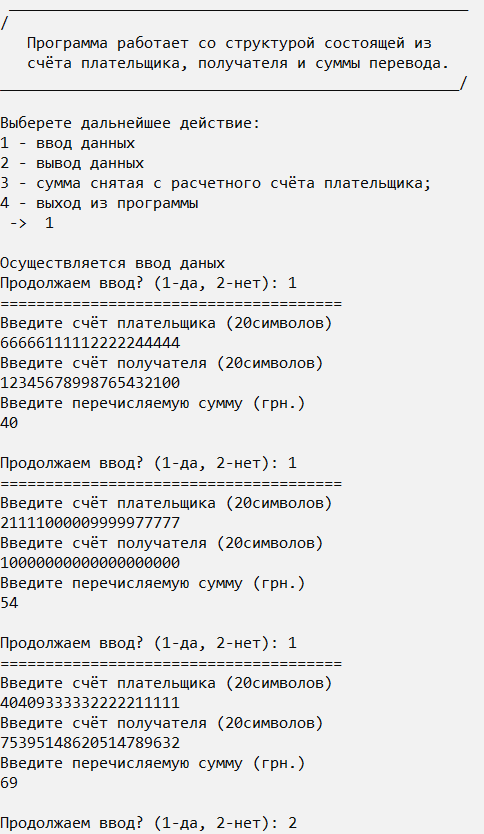


Рисунок 6.3 – Меню программы и ввод структур

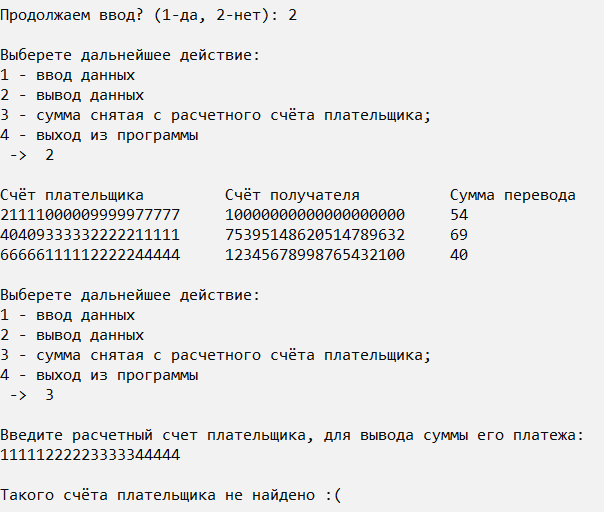


Рисунок 6.4 – Вывод отсортированных структур и поиск несущ. счёта плательщика

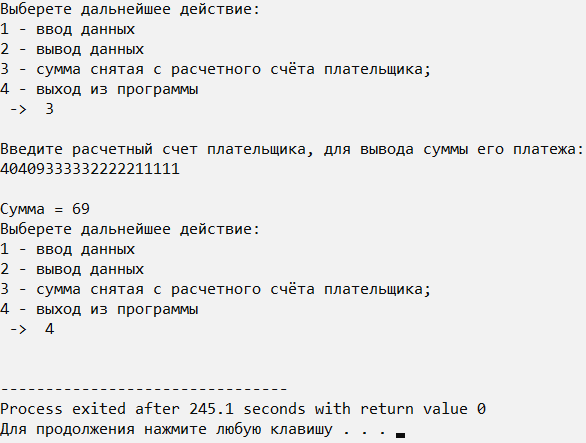


Рисунок 6.5 – Поиск существующего счёта плательщика и выход из программы

Результаты тестирования полностью соответствуют ожиданиям.

**Выводы**

В ходе выполнения данной лабораторной работы были получены навыки программирования операций над структурами и бинарными файлами. Были изучены способы описания структур данных на языке Си. Исследованы особенности обработки бинарных файлов, хранящих структурные типы данных. Полученные во время разработки навыки помогут разрабатывать более сложные программы со структурами данных и бинарными файлами, более эффективные по времени выполнения алгоритмы.