Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №10**

По дисциплине «Основы алгоритмизации и программирования»

Тема: «Структуры, перечисления, объединения»

**Выполнил:**

Студент 1 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Гирель Т.Н

Брест 2022

**Цель работы:** Изучить синтаксис и правила работы со структурами. Реализовать программу с применением структур, перечислений и объединений.

**Вариант №1**

**Текст задания**

**Создать тип структуры согласно варианту, организовать поля этой структуры так, чтобы они содержали объединение, перечисление (можно добавить дополнительные поля) и битовое поле.**

**Создать массив структур, содержащий информацию согласно варианту индивидуального задания.**

**Реализовать работу с массивом структур через меню: ввод данных в массив, вывод содержимого массива на экран, сортировка по одному полю, удаления записи по заданному значению поля, выборка записей согласно индивидуальному заданию.**

**Текст задания по варианту**

**В магазине сформирован список постоянных клиентов, который включает ФИО, домашний адрес покупателя и размер предоставляемой скидки. Вывести всех покупателей, имеющих 5 % - ную скидку.**

**Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

struct data {

char name[15];

char surname[15];

char middle\_name[15];

unsigned valid\_discount : 1;

char local\_street[20];

short discount;};

short razmer = 0;

struct data\* people;

void output\_struct() {

printf("%-5s %-15s %-15s %-15s %-20s %-15s\n", "Номер", "Имя", "Фамилия", "Отчество", "Улица проживания", "Размер скидки");

for (short i = 0; i < razmer; i++) {

printf(" %-3d", i + 1);

printf(" %-15s", people[i].name);

printf(" %-15s", people[i].surname);

printf(" %-15s", people[i].middle\_name);

printf(" %-20s", people[i].local\_street);

printf(" %-d%%\n", people[i].discount); }

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

void input\_struct() {

++razmer;

people = (struct data\*)realloc(people, razmer \* sizeof(struct data));

printf("Введите данные:\n");

printf("Имя: ");

scanf("%s", people[razmer - 1].name);

printf("Фамилия: ");

scanf("%s", people[razmer - 1].surname);

printf("Отчество: ");

scanf("%s", people[razmer - 1].middle\_name);

printf("Улица проживания: ");

scanf("%\*c");

scanf("%[^\n]", people[razmer - 1].local\_street);

printf("Размер скидки в процентах: ");

scanf("%hd", &people[razmer - 1].discount);

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

void procent\_5() {

for (short i = 0; i < razmer; i++) {

printf("%-5s %-15s %-15s %-15s %-20s %-15s\n", "Номер", "Имя", "Фамилия", "Отчество", "Улица проживания", "Размер скидки");

if (people[i].discount == 5) {

printf(" %-3d", i + 1);

printf(" %-15s", people[i].name);

printf(" %-15s", people[i].surname);

printf(" %-15s", people[i].middle\_name);

printf(" %-20s", people[i].local\_street);

printf(" %-d%%\n", people[i].discount);}}

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

void delete\_element() {

short id;

output\_struct();

printf("Введите id элемента который нужно удалить: ");

scanf("%hd", &id);

struct data buffer;

for (short i = 0; i < razmer - id; i++) {

buffer = people[i + id];

people[i + id] = people[i + id - 1];

people[i + id - 1] = buffer;}

razmer--;

people = (struct data\*)realloc(people, razmer \* sizeof(struct data));

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

void sort() {

short choise, maxmin, id;

struct data buffer;

printf("Хотите сортировать\nПо возрастанию -> 1\nИли убыванию -> 2 ?\nВаш выбор: ");

scanf("%hd", &choise);

if (choise == 1) {

for (short i = 0; i < razmer; i++) {

maxmin = 100; id = 0;

for (short j = i; j < razmer; j++) {

if (people[j].discount <= maxmin) { maxmin = people[j].discount; id = j; }}

buffer = people[id];

people[id] = people[i];

people[i] = buffer;}}

else {

for (short i = 0; i < razmer; i++) {

maxmin = 0; id = 0;

for (short j = i; j < razmer; j++) {

if (people[j].discount >= maxmin) { maxmin = people[j].discount; id = j; }}

buffer = people[id];

people[id] = people[i];

people[i] = buffer;}}

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

void init() { struct data\* people = (struct data\*)calloc(sizeof(struct data), razmer); }

int main() {

init();

setlocale(LC\_ALL, "Rus");

short Exit = 0;

short choise = 0;

while (Exit == 0) {

printf("Что вы собираетесь сделать?\n1.Добавить новый элемент структуры\n2.Вывести всю таблицу\n3.Вывести всех с скидкой 5%\n4.Сортировка по скидке\n5.Удалить элемент структуры\n6.Выйти из программы\nВаш выбор: ");

scanf("%hd", &choise);

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

if (choise == 1) input\_struct();

if (choise == 2) output\_struct();

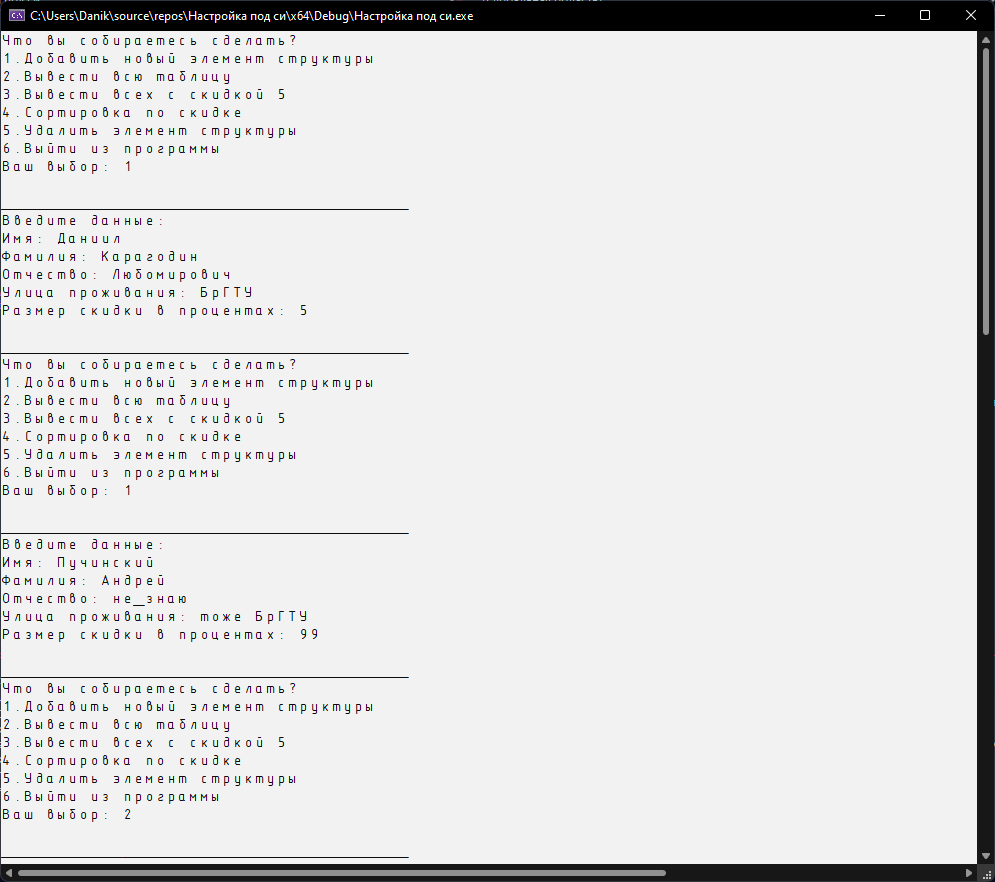
if (choise == 3) procent\_5();

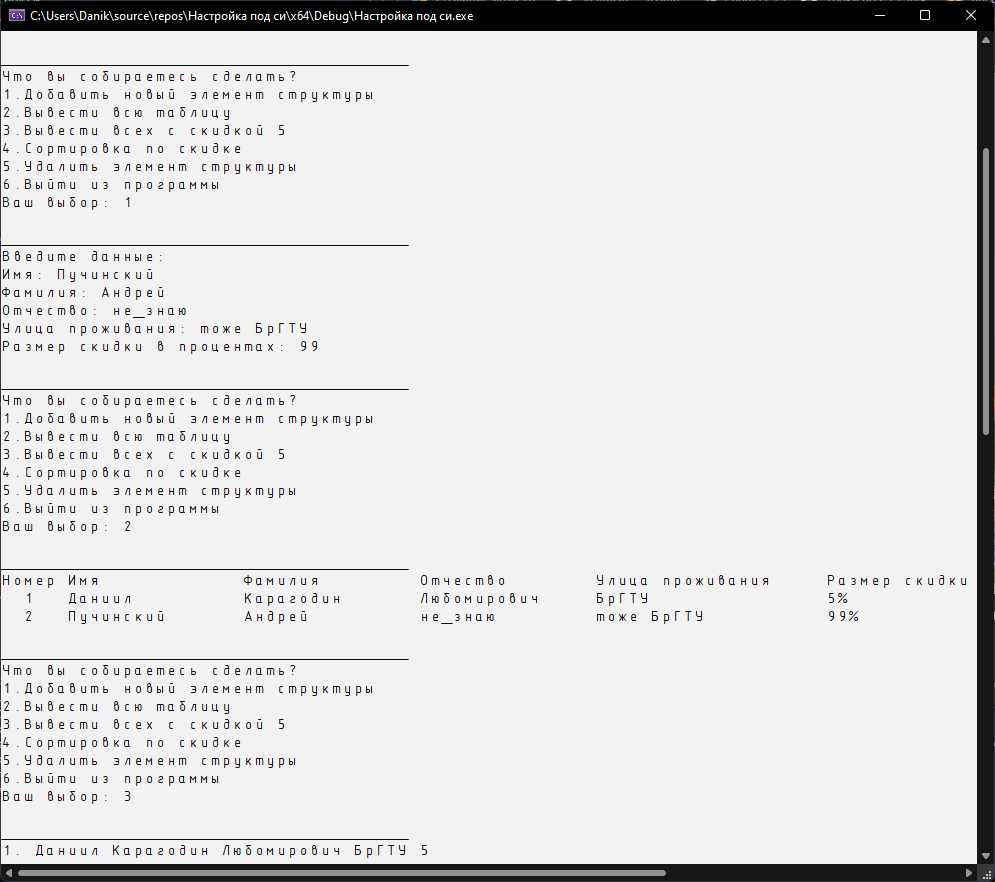
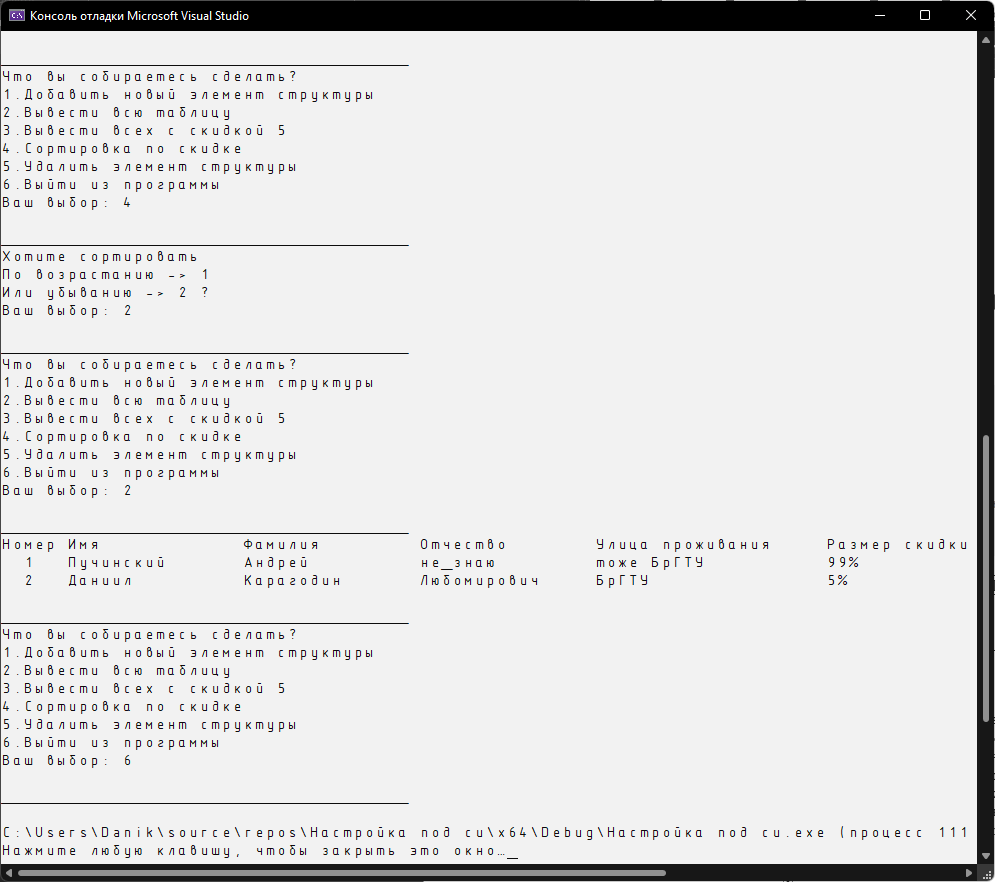
if (choise == 4) sort();

if (choise == 5) delete\_element();

if (choise == 6) Exit = 1;}

return 0;}

**Результат работы**



**Вывод:** Изучил синтаксис и правила работы со структурами. Реализовал программу с применением структур, перечислений и объединений.