МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №9

по дисциплине: Основы программирования тема: «Использование структур»

Выполнил: ст. группы ПВ-201 Машуров Дмитрий Русланович

Проверил:

Притчин Иван Сергеевич Брусенцева Валентина

Станиславовна

Белгород 2021 г.

**Лабораторная работа № 9**

**Использование структур**

**Цель работы**: получение навыков работы со структурами.

**Задания для подготовки к работе**

1. Изучить:

* назначение структурного типа и правила его описания;
* правила описания и инициализации переменных-структур;
* назначение и описание битовых полей;
* обращение к членам структур с помощью прямого и косвенного селектора.

1. Разработать алгоритм и составить программу для решения задачи соответствующего варианта.
2. Подобрать наборы тестовых данных.

**Задание варианта №16**

Даны адреса n человек (n - const). Вывести фамилии людей, живущих в разных городах по одинаковым адресам или сообщить, что таких людей не существует.

**Выполнение:**

1. **Описание алгоритма и выделение подзадач:**

Исходя из условия задачи, будем вводить данные пользователей по очереди и затем определять совпадение адресов и несовпадение городов

Выделение подзадач:

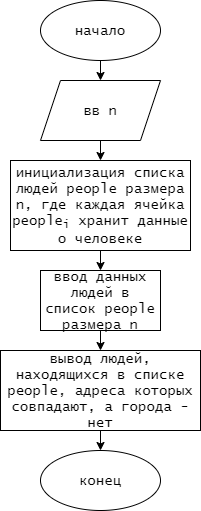
1. Ввод данных людей
2. Сравнение данных людей
3. **Структуры данных**

Для определения данных о человеке:

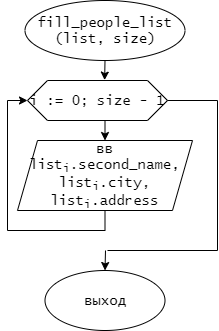
1. typedef struct data {  
    char second\_name[50];  
    char city[50];  
    char address[50];  
   } person;

Где second\_name – фамилия, city – город, address – адрес

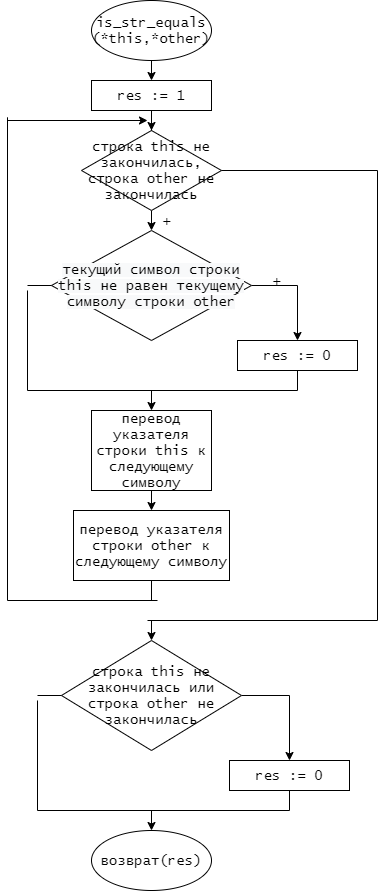
1. **Блок-схема с укрупнёнными блоками**

****

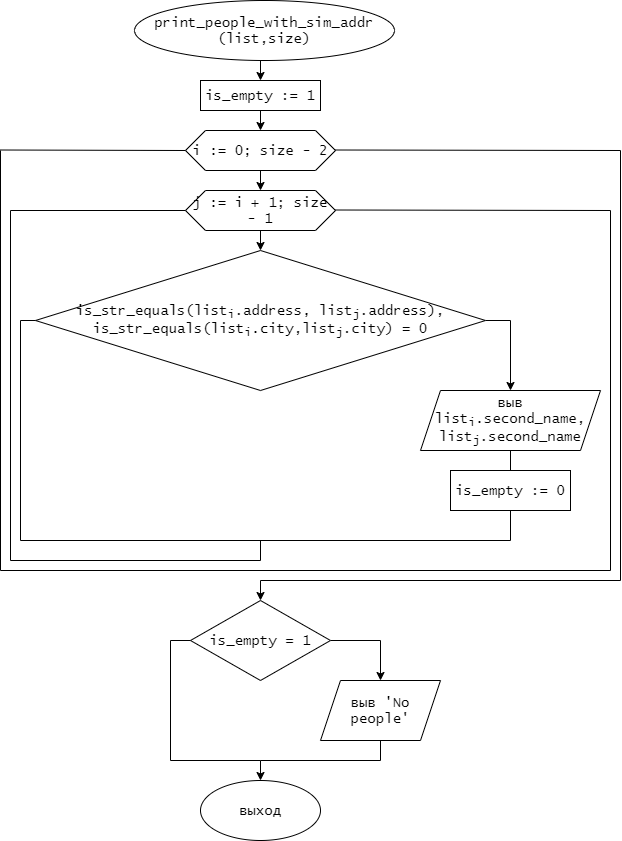
1. **Описание подпрограмм**
2. Ввод данных людей
   1. Заголовок: void fill\_people\_list(person list[], int size)
   2. Назначение: заполняет список людей list размера size
   3. Блок-схема:



1. Определение равенства строк
   1. Заголовок: int is\_str\_equals(const char \*this, const char \*other)
   2. Назначение: возвращает «истину», если строка this равна строке other, иначе – «ложь»
   3. Блок-схема:



1. Сравнение данных людей
   1. Выделение подзадач:
      1. Определение равенства строк
   2. Заголовок: void print\_people\_with\_sim\_addr(person \*list, int size)
   3. Назначение: вывод людей из списка list размера size, адреса которых совпадают, а города – нет
   4. Блок-схема:



1. **Тестовые данные**

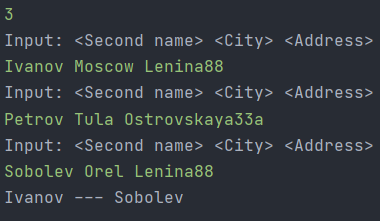
|  |  |  |
| --- | --- | --- |
| **№** | **Ввод** | **Вывод** |
| **1** | “Ivanov Moscow Lenina88”  “Petrov Tula Ostrovskaya33a”  “Sobolev Orel Lenina88” | “Ivanov --- Sobolev” |
| **2** | “Ivanov Belgorod Sumskaya8” “Petrov Belgorod Sumskaya9” | Нет таких людей |

1. **Текст программы**

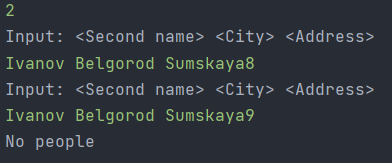
#include <stdio.h>  
  
typedef struct data {  
 char second\_name[50];  
 char city[50];  
 char address[50];  
} person;  
  
/\*возвращает "истину", если строки this и other равны, иначе - "ложь"\*/  
int is\_str\_equals(const char \*this, const char \*other) {  
 while (\*this != '\0' || \*other != '\0') {  
 if (\*this != \*other) {  
 return 0;  
 }  
  
 this++; other++;  
 }  
  
 if (\*other != '\0' || \*this != '\0') {  
 return 0;  
 } else {  
 return 1;  
 }  
}  
  
/\*ввод данных списка людей list размера size\*/  
void fill\_people\_list(person list[], int size) {  
 for (int i = 0; i < size; ++i) {  
 printf("Input: <Second name> <City> <Address>\n");  
 scanf("%s %s %s", list[i].second\_name, list[i].city, list[i].address);  
 }  
}  
  
/\*вывод людей из списка list размера size, адреса которых совпадают, а города - нет\*/  
void print\_people\_with\_sim\_addr(person \*list, int size) {  
 int is\_empty = 1;  
 for (int i = 0; i < size - 1; ++i) {  
 for (int j = i + 1; j < size; ++j) {  
 if ((is\_str\_equals(list[i].address, list[j].address)) && !(is\_str\_equals(list[i].city, list[j].city))) {  
 printf("%s --- %s\n", list[i].second\_name, list[j].second\_name);  
 is\_empty = 0;  
 }  
 }  
 }  
  
 if (is\_empty) {  
 printf("No people\n");  
 }  
}  
  
int main() {  
 int n;  
 scanf("%d", &n);  
  
 person people[n];  
 fill\_people\_list(people, n);  
  
 print\_people\_with\_sim\_addr(people, n);  
}

1. **Результаты:**

*Пример №1:*



*Пример №2:*



1. **Анализ ошибок:**

* Нет ошибок