**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 6**

**по дисциплине «Программирование»**

Тема: Структуры. Массивы структур

|  |  |  |
| --- | --- | --- |
| Студент гр. 3312 |  | Шарапов И. Д. |
| Преподаватель |  | Аббас С. А. |

Санкт-Петербург

2024

**Содержание**

[Цель работы 3](#_Toc159768431)

[Задание (Вариант 6) 3](#_Toc159768432)

[Постановка задачи и описание решения 3](#_Toc159768433)

[Описание переменных 4](#_Toc159768434)

[Структура вызова функций 4](#_Toc159768435)

[Схема алгоритма 5](#_Toc159768436)

[Текст программы 8](#_Toc159768437)

[Контрольные примеры 10](#_Toc159768438)

[Содержимое файлов 10](#_Toc159768439)

[Примеры выполнения программы 11](#_Toc159768440)

[Выводы 12](#_Toc159768441)

# Цель работы

Целью работы является изучение структур и работы с ними в языке Си, а также получение практических навыков в решение задач с большим количеством данных.

# Задание (Вариант 6)

Описать структуру с именем NOTE2, содержащую поля: Name – фамилия и инициалы, TELE – номер телефона, DATE – дата рождения (год, месяц, число). Написать программу, выполняющую:

* ввод с клавиатуры данных в массив BLOCK2, состоящий из 7 элементов типа NOTE2, записи должны быть упорядочены по первым трем цифрам номера телефона;
* вывод на экран информации о человеке, чья фамилия введена с клавиатуры, если такого нет – выдать сообщение.

# Постановка задачи и описание решения

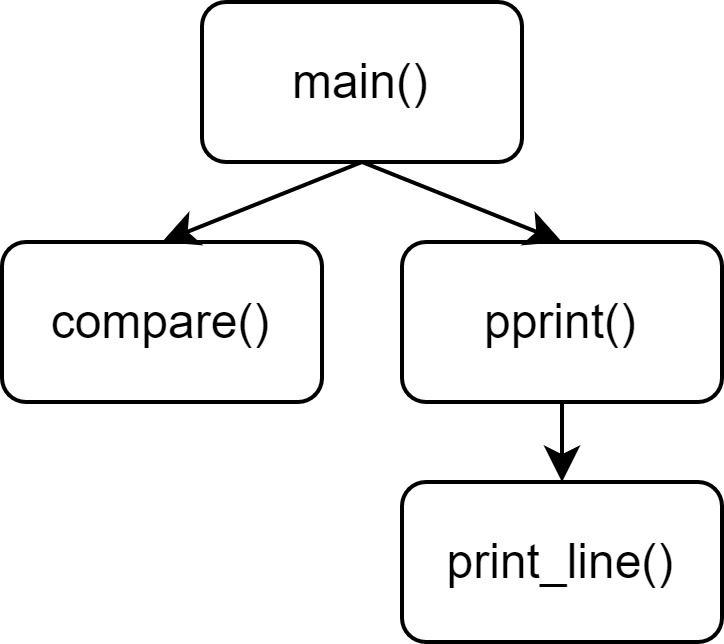
В начале мы спрашиваем пользователя имя файла, из которого мы возьмём данные о пользователях. Спрашиваем до тех пор, пока файл с введённым именем не будет успешно открыт. Далее мы выводим пользователю функционал программы. Затем обрабатываем информацию из файла, построчно считывая и добавляя пользователя в массив. На этом этапе происходит разделение строки через разделители с помощью функции *strtok*.

После заполнения массива BLOCK2 мы сортируем его «пузырьком». После этого начинаем считывать команды пользователя. Если пользователь введёт «!print», то программа выводит таблицу используя функцию *pprint()*. Если пользователь введёт «!end», то программа выведет пользователю сообщение «Goodbye!» и завершит свою работу. В случае ввода любого другого текста программа переберёт всех пользователей и проверит на соответствие поле Name у всех пользователей с помощью функции *strcasecmp.* В случае соответствия программа выведет через tab все оставшиеся поля пользователя.

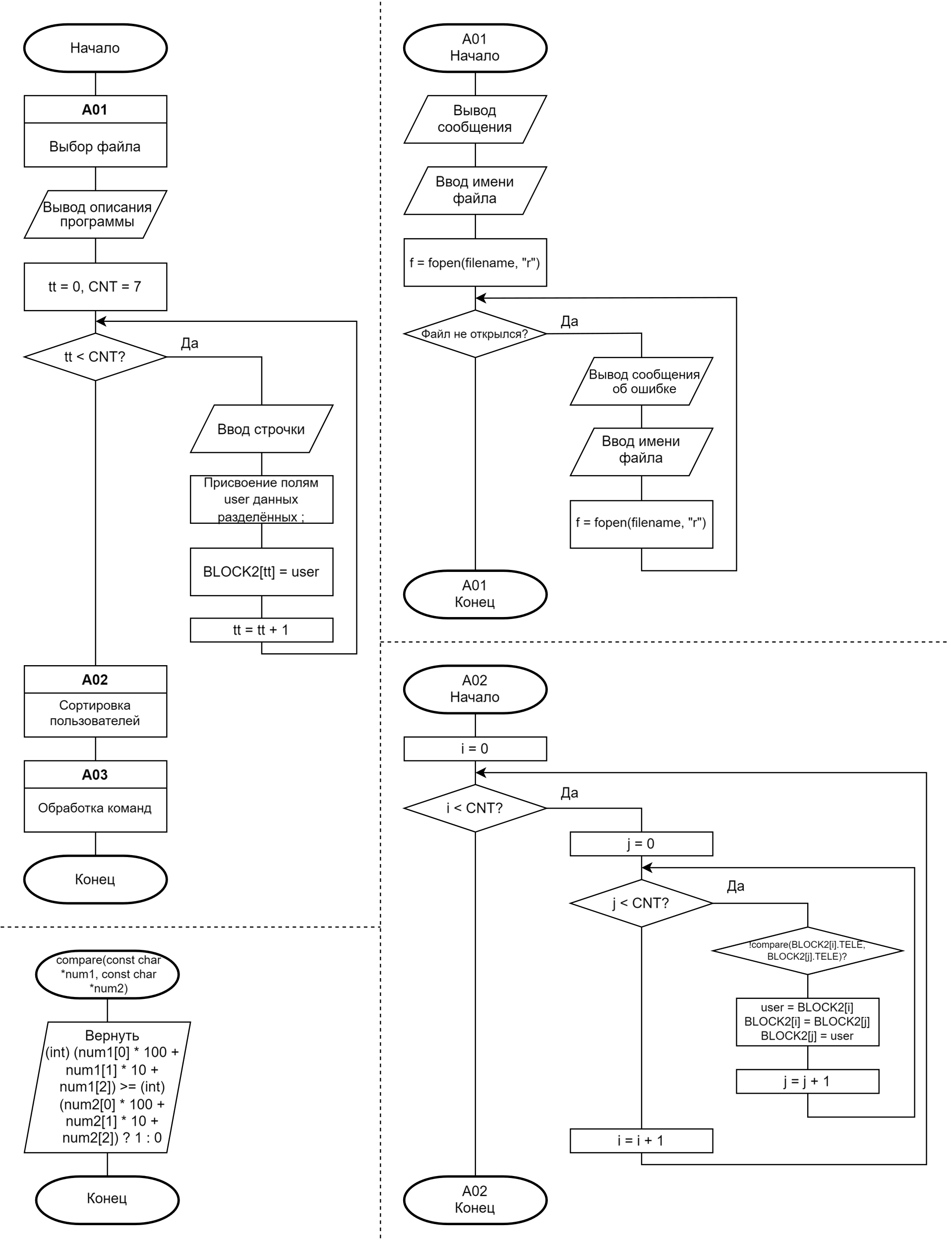
# Описание переменных

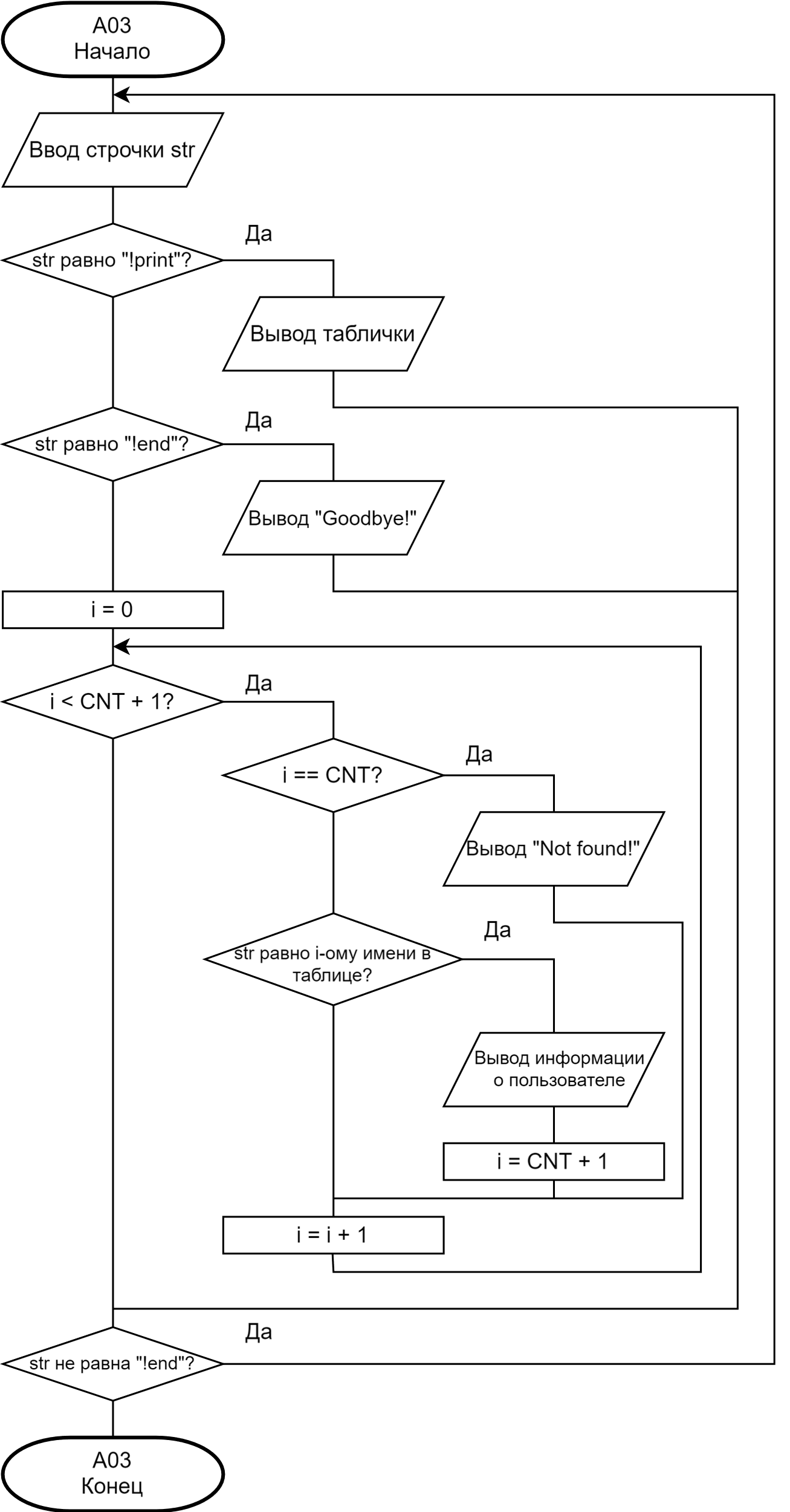
|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Тип | Назначение |
| Функция *int compare(const char \*num1, const char \*num2)* | | | |
| 1 | num1 | char \* | Первый номер телефона для сравнения |
| 2 | num2 | char \* | Второй номер телефона для сравнения |
| Функция *void print\_line(int x)* | | | |
| 1 | x | int | Ширина первого столбца таблицы |
| Функция *void pprint(note \*block)* | | | |
| 1 | max\_name | int | Длина максимального слова в первом столбце |
| 2 | x | int | Длина i-ого слова |
| Функция *int main()* | | | |
| 1 | filename | char[] | Имя файла с данными |
| 2 | f | FILE \* | Указатель на файл |
| 3 | text | char[][] | Информация о пользователях в виде текста |
| 4 | cnt | int | Счётчик количества обработанных данных о конкретном пользователе |
| 5 | user | note | Данные пользователя |
| 6 | BLOCK2 | note[] | Массив пользователей |
| 7 | token | char \* | Указатель на разделитель |
| 8 | str | char \* | Строка запроса |

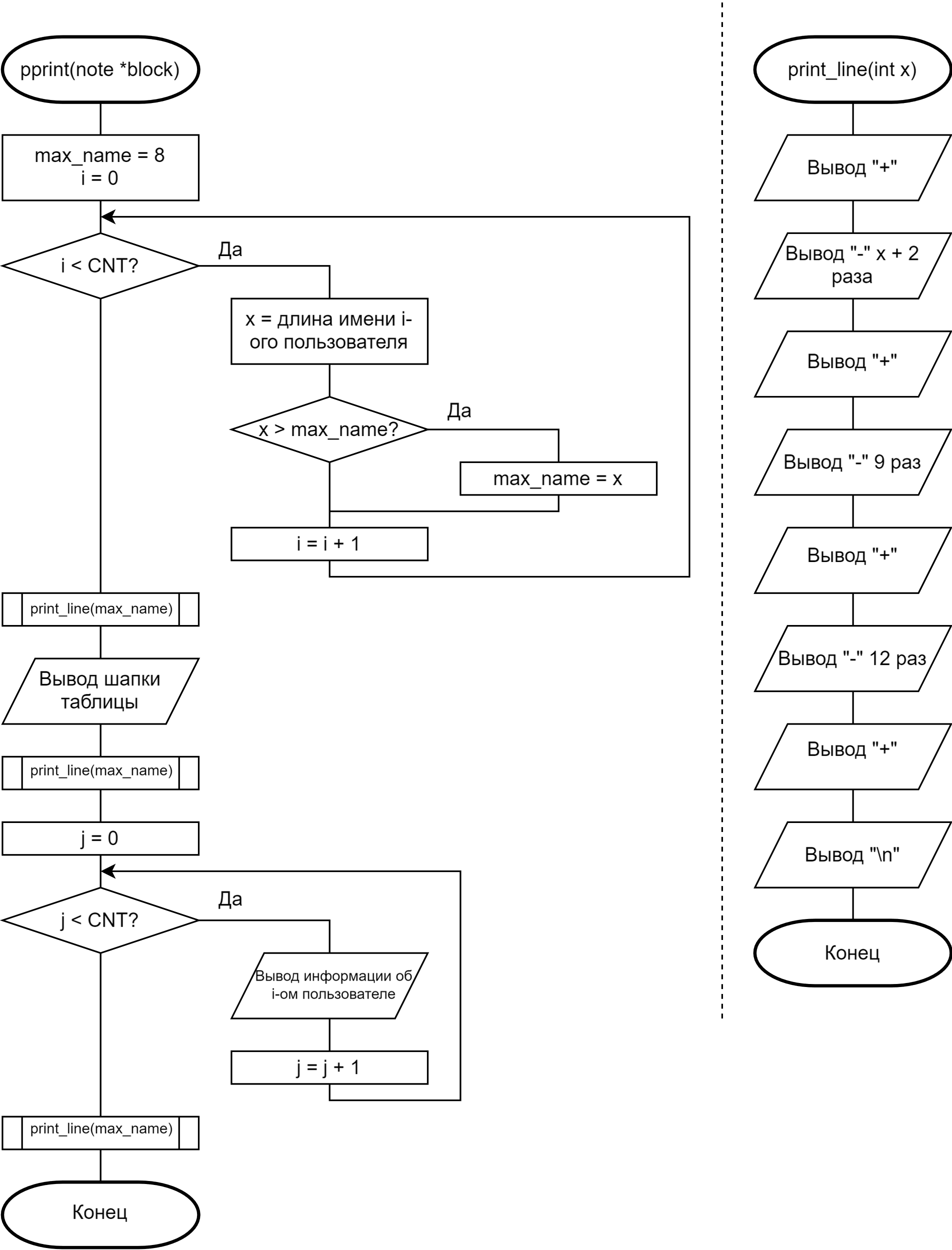
# **Структура вызова функций**



# **Схема алгоритма**







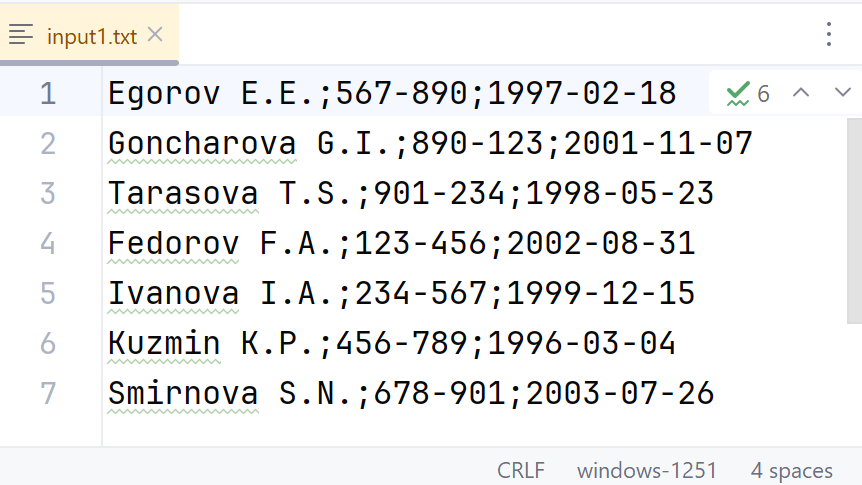
# Текст программы

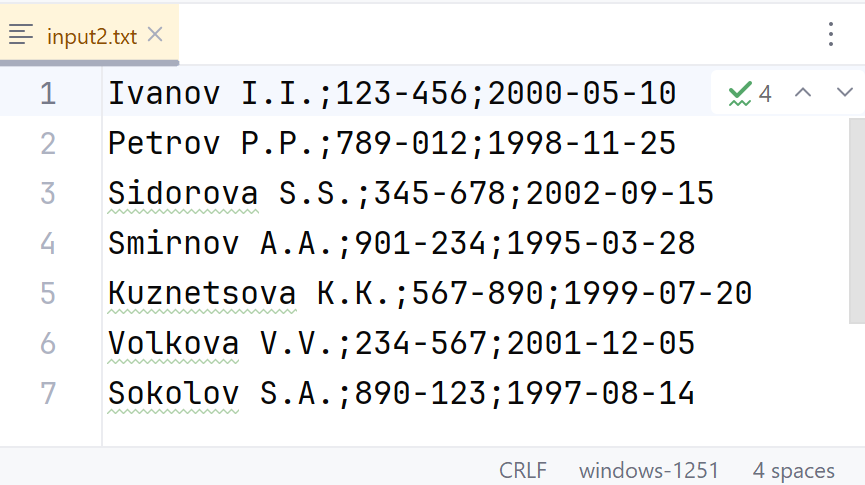
|  |
| --- |
| **#include <stdio.h> #include <string.h>  #define CNT 7  struct NOTE2 {  char \*Name;  char \*TELE;  char \*DATE; };  typedef struct NOTE2 note;  int compare(const char \*num1, const char \*num2) {  return (int) (num1[0] \* 100 + num1[1] \* 10 + num1[2]) >= (int) (num2[0] \* 100 + num2[1] \* 10 + num2[2]) ? 1 : 0; }  void print\_line(int x) {  printf("+");  for (int i = 0; i < x + 2; printf("-"), ++i);  printf("+");  for (int i = 0; i < 9; printf("-"), ++i);  printf("+");  for (int i = 0; i < 12; printf("-"), ++i);  printf("+");  printf("\n"); }  void pprint(note \*block) {  int max\_name = 8, x;  for (int i = 0; i < CNT; ++i) {  x = strlen(block[i].Name);  if (x > max\_name) max\_name = x;  }   print\_line(max\_name);   printf("| %s ", "Fullname");  for (int i = 0; i < max\_name - 8; printf(" "), ++i);  printf("| %s | %s |\n", "Number ", "Birthday ");   print\_line(max\_name);   for (int j = 0; j < CNT; ++j) {  printf("| %s ", block[j].Name);  for (int i = 0; i < max\_name - strlen(block[j].Name); printf(" "), ++i);  printf("| %s | %s |\n", block[j].TELE, block[j].DATE);  }   print\_line(max\_name); }  int main() {  char filename[128];  FILE \*f;  char text[CNT][1024];  int cnt;  note user;  note BLOCK2[CNT];  char \*token;  char \*str;   printf("Please enter the file name:\n");  scanf("%s", filename);  f = fopen(filename, "r");  while (f == NULL) {  printf("Something went wrong! Perhaps such a file does not exist.\nPlease enter the file name again:\n");  scanf("%s", filename);  f = fopen(filename, "r");  }   printf("If you want to display table, enter \'!print\'.\n");  printf("If you want to search for a user, enter the full name.\n");  printf("If you want to close the program, enter \'!end\'.\n");   for (int tt = 0; tt < CNT; ++tt) {  fgets(text[tt], sizeof(text[tt]), f);  token = strtok(text[tt], ";");  cnt = 0;  while (token != NULL) {  if (cnt == 0) user.Name = token;  if (cnt == 1) user.TELE = token;  if (cnt == 2) user.DATE = token;  token = strtok(NULL, ";");  ++cnt;  }  user.DATE[10] = '\0';  BLOCK2[tt] = user;  }   for (int i = 0; i < CNT; ++i) {  for (int j = 0; j < CNT; ++j) {  if (!compare(BLOCK2[i].TELE, BLOCK2[j].TELE)) {  user = BLOCK2[i];  BLOCK2[i] = BLOCK2[j];  BLOCK2[j] = user;  }  }  }   getchar();  do {  fgets(str, 1024 \* sizeof(char), stdin);  str[strlen(str) - 1] = '\0';  if (strcasecmp(str, "!print") == 0) {  pprint(BLOCK2);  } else if (strcasecmp(str, "!end") == 0) {  printf("Goodbye!");  } else {  for (int i = 0; i < CNT + 1; ++i) {  if (i >= CNT) {  printf("Not found!\n");  } else if (strcasecmp(str, BLOCK2[i].Name) == 0) {  printf("%s\t%s\t%s\n", BLOCK2[i].Name, BLOCK2[i].TELE, BLOCK2[i].DATE);  i = CNT + 1;  }  }  }  } while (strcasecmp(str, "!end") != 0);  return 0; }** |

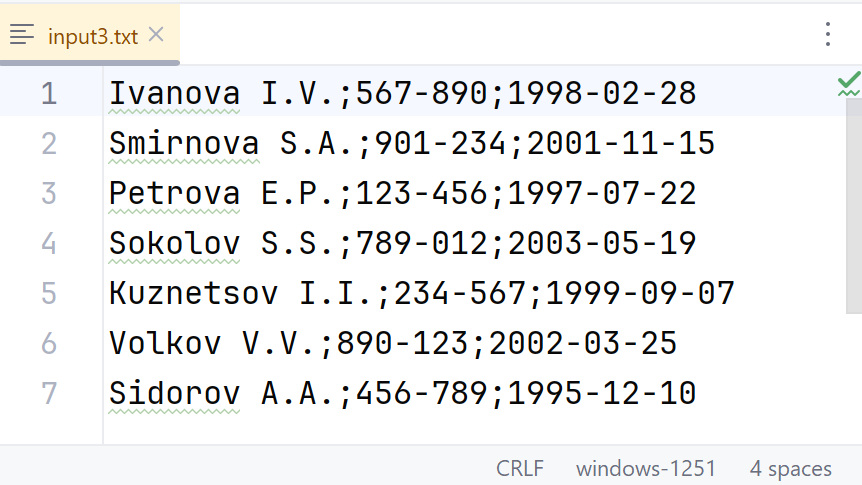
# Контрольные примеры

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | input1.txt  Tarasova T.S.  Petrova E.P.  !end | Tarasova T.S. 901-234 1998-05-23  Not found!  Goodbye! |
| 2 | input2.txt  !print  !end | Goodbye! |
| 3 | input3.txt  Petrova E.P.  Smirnov A.A.  Volkov V.V.  !end | Petrova E.P. 123-456 1997-07-22  Not found!  Volkov V.V. 890-123 2002-03-25  Goodbye! |

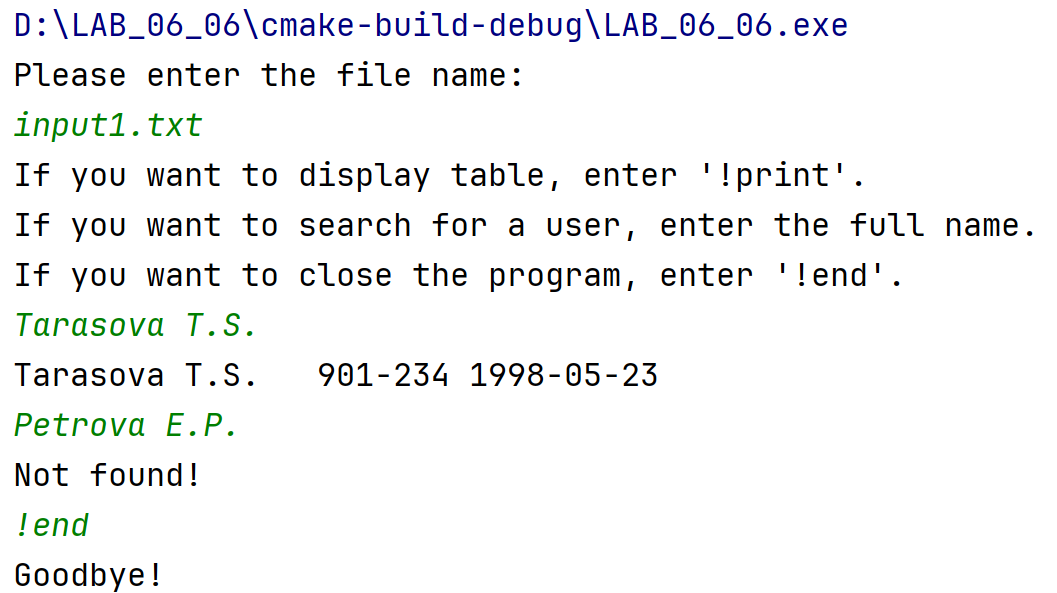
# Содержимое файлов

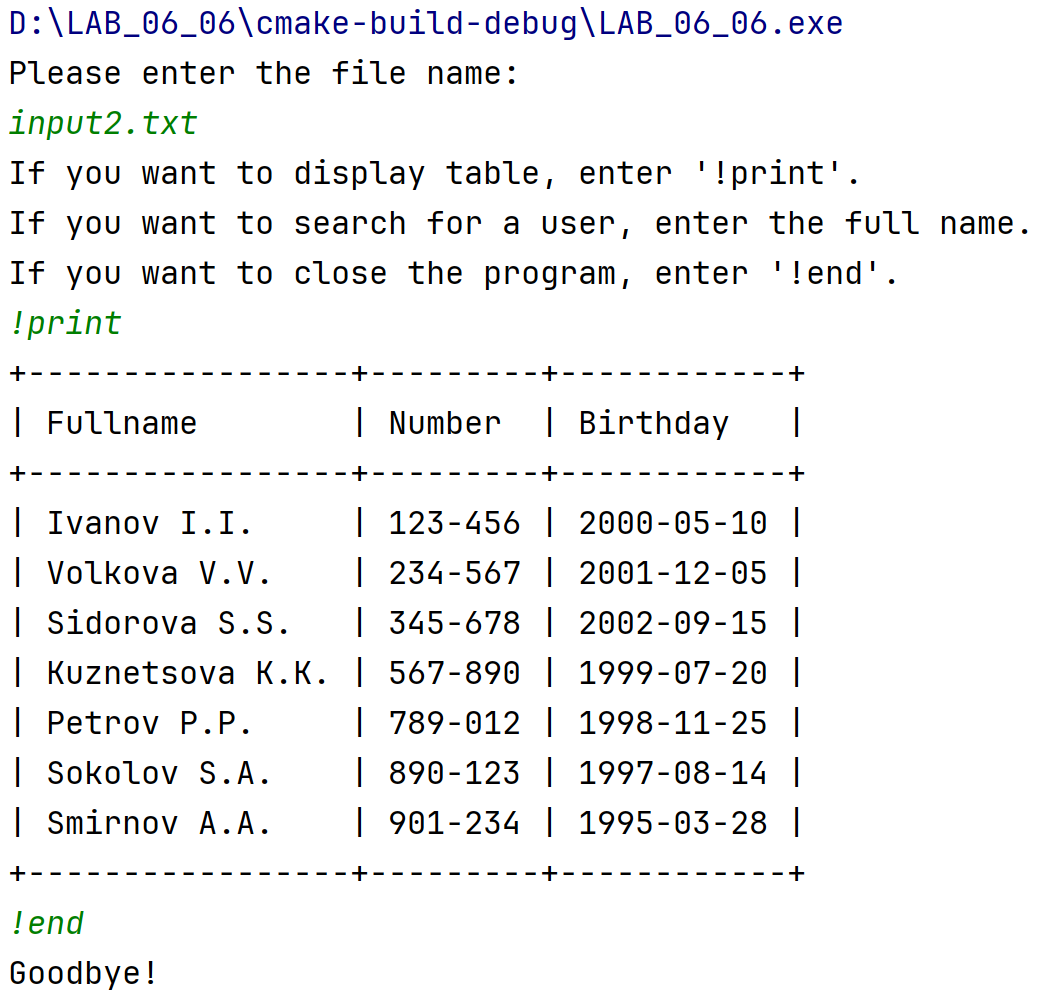


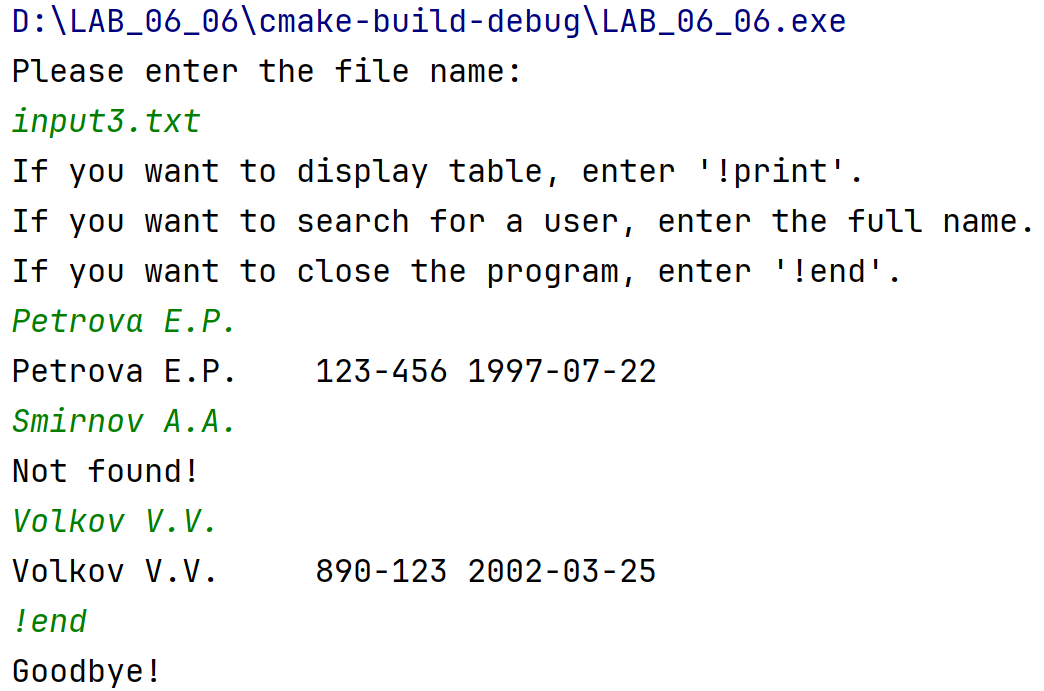




# Примеры выполнения программы







# Выводы

В результате выполнения работы изучены особенности работы с структурами в языке Си. А также получены практические навыки в решении задач на обработку большого количества данных.