

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
ТЕМА: ОБРАБОТКА ТЕКСТОВЫХ ФАЙЛОВ

Студент гр. 6303

Зубов К.А.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент Зубов К.А

Группа 6304

Тема работы: Обработка текстовых файлов

Содержание пояснительной записки:

- Содержание
- Введение
- Описание функций, необходимых для работы программы
- Примеры работы программы
- Заключение
- Список использованных источников
- Приложение

Дата выдачи задания:

Дата сдачи:

Дата защиты:

Студент		Зубов К.А
Преподаватель		Берленко Т.А.

АННОТАЦИЯ

В данной работе была создана программа на языке программирования C, которая производит обработку csv-файлов с помощью набора функций. Были созданы и описаны необходимые функции, позволяющие считывание строк, содержащихся в файлах, проводить поиск определенных строк в файлах, проводить поиск определенных значений из строк, содержащихся в файлах, сохранять данные из файлов. Приведены примеры работы программы, а также полное описание исходного кода.

Оглавление

1	
2	
1.1	2
1.2	2
1.3	2
1.4	Функция вывода баллов студента.3
1.5	Функция main4
	Примеры работы программы6
	Заключение7
	Список использованных источников8
	Приложение А9
	Исходный код программы9

Введение

Требуется написать программу, на вход которой подаются две таблицы, хранящиеся в csv файлах. Первая таблица содержит информацию о ФИО студента и его Github аккаунте, email'е и номере группы. Вторая таблица содержит фамилию, имя, количество баллов за экзамен. Программа должна находить количество студентов, которые получили максимальный балл и выводить результат на консоль.

Программа получает параметры из входного потока. Параметры:

- input_file_1 - csv файл
- input_file_2 - csv файл

В случае, если программа получила некорректные параметры, то:

- не создается выходного файла
- выводится сообщение об ошибке “Fail with <имя параметра>”.

Функции для работы программы

1.1 Объявление структур

```
typedef struct Students_file1
{
    char last_name[N];
    char first_name[N];
} Students_file1;
```

```
typedef struct Students_file2
{
    char last_name[N];
    char first_name[N];
    int Score;
} Students_file2;
```

Объявление структур для считывания данных из 1 и 2 входных файлов. Поля last_name - фамилия, first_name имя, Score - балл студента.

1.2 Функция записи студента

```
int getstudfile_1(char *string, Students_file1 *Studfrom1, int num)
{
    char *last_name = strtok(string, ";");
    char *first_name = strtok(NULL, ";");
    strcpy(Studfrom1[num].last_name, last_name);
    strcpy(Studfrom1[num].first_name, first_name);
    return 0;
}
```

Функция получает указатели на строку, на массив студентов и номер студента. После этого записывает фамилию, имя и балл студента в массив.

1.3 Функция нахождения количества студентов в таблицах

```
int Numofstud(FILE *file_1, FILE *file_2, Students_file1 *Studfrom1,
Students_file2 *Studfrom2) {
    int n = 0, m = 0;
    char *string = (char *)malloc(200);
    while (!feof(file_1))
    {
        fscanf(file_1, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_1(string, Studfrom1, n) == 0)
                n++;
            string[0] = '\0';
        }
    }
    while (!feof(file_2))
    {
        fscanf(file_2, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_2(string, Studfrom2, m) == 0)
                m++;
            string[0] = '\0';
        }
    }
    return n + m;
}
```

```

    }
    }
    while (!feof(file_2))
    {
        fscanf(file_2, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_2(string, Studfrom2, m) == 0)
                m++;
            string[0] = '\0';
        }
    }
    free(string);
    if (n == m){
        return n;}
    else{
        printf("Fail with number of students");
    }

}

```

Данной функции передают четыре параметра: два указателя на входные файлы, указатель на массив студентов, указатель на кол-во студентов в массиве. Функция возвращает:

Количество студентов в файлах, если оно равно.

Fail with number of students – если количество студентов в таблицах не равное.

1.4 Функция вывода баллов студента.

```

int getscore(Students_file2 Studfrom2, int NumOfStud)
{
    int i;
    for (i = 0; i < NumOfStud; i++)
    {
        return Studfrom2.Score;
    }
}

```

Функция получает количество студентов и элемент структуры, содержащий данные о студенте. Циклически возвращает баллы каждого студента.

1.5 Функция main

```
int main() {
    setlocale(LC_ALL, "");

    FILE* file_1 = fopen("cw/input_file_1.csv", "r");
    FILE* file_2 = fopen("cw/input_file_2.csv", "r");
    Students_file1 *Studfrom1 = (Students_file1 *)calloc(100,
sizeof(Students_file1));
    Students_file2 *Studfrom2 = (Students_file2 *)calloc(100,
sizeof(Students_file2));
    if (file_1 == NULL) {
        printf("Fail with input_file_1\n");
        return 0;
    }
    else if (file_2 == NULL) {
        printf("Fail with input_file_2\n");
        return 0;
    }
    int NumOfStud = Numofstud( file_1, file_2, Studfrom1, Studfrom2);

    int max = 0, count = 0,i = 0;
    for ( i; i<NumOfStud; i++)
    {
        int studsscore = getscore(Studfrom2[i], NumOfStud);

        if (studsscore > max)
        {
            max = studsscore;
            count = 1;
        }
        else if (studsscore == max)
            count++;
    }

    printf("Кол-во студентов с высшим баллом = %d\n", count);
    fclose(file_1);
}
```



```
    fclose(file_2);  
    free(Studfrom2);  
    free(Studfrom1);  
  
    return 0;  
}
```

В функции происходит открытие файлов для чтения. Выделение памяти для массивов, содержащих информацию о студентах. Файлы проверяются на корректное открытие. Происходит нахождение максимального балла и подсчет студентов, имеющих высший балл. Количество студентов с максимальным баллом выводится на консоль. После происходит закрытие файлов и освобождение памяти.

Примеры работы программы

- Пример работы программы при корректных входных данных

• input_file_1	input_file_2	Результат
Зубов;Константин;Александрович;git;mail;6304 Коряков;Максим;Станиславович;git;mail;6304 Шалашов;Максим;Денисович;git;mail;6304 Иванов;Владимир;Алексеевич;git;mail;6304 Григорьев;Иван;Алексадрович;git;mail;6304 Кренделев;Артур;Иванович;git;mail;6304 Чумаков;Андрей;Генадьевич;git;mail;6304 Кострыкин;Данил;Александрович;git;mail;6304 Смирнов;Владислав;Сергеевич;git;mail;6304 Беседин;Алексей;Петрович;git;mail;6304	Зубов;Константин;250 Коряков;Максим;190 Шалашов;Максим;150 Иванов;Владимир;250 Григорьев;Иван;250 Кренделев;Артур;150 Чумаков;Андрей;210 Кострыкин;Данил;240 Смирнов;Владислав;250 Беседин;Алексей;100	4
Григорьев;Иван;Алексадрович;git;mail;6304 Кренделев;Артур;Иванович;git;mail;6304 Чумаков;Андрей;Генадьевич;git;mail;6304 Кострыкин;Данил;Александрович;git;mail;6304 Смирнов;Владислав;Сергеевич;git;mail;6304 Беседин;Алексей;Петрович;git;mail;6304	Григорьев;Иван;250 Кренделев;Артур;150 Чумаков;Андрей;210 Кострыкин;Данил;240 Смирнов;Владислав;250 Беседин;Алексей;100	2

- Пример работы программы при отсутствии одного из входных файлов

input_file_1	input_file_2	Результат
Отсутствует	Зубов;Константин;250 Коряков;Максим;190 Шалашов;Максим;150 Иванов;Владимир;250 Григорьев;Иван;250	Fail with input_file_1
Зубов;Константин;Александрович;git;mail;6304 Коряков;Максим;Станиславович;git;mail;6304 Шалашов;Максим;Денисович;git;mail;6304 Иванов;Владимир;Алексеевич;git;mail;6304 Григорьев;Иван;Алексадрович;git;mail;6304	Отсутствует	Fail with input_file_2

Заключение

В ходе работы была создана программа обработки двух csv-файлов, выполняющая необходимые действия. Также для работы программы были созданы и описаны все необходимые функции и структуры.

Список использованных источников

1. Язык программирования СИ / Керниган Б., Ритчи Д. СПб.: Издательство "Невский Диалект", 2001. 352 с.
2. Полный справочник по С / Шилдт Г.: Вильямс, 2004. - 752 с.

Приложение А

Исходный код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>

#define N 15

typedef struct Students_file1
{
    char last_name[N];
    char first_name[N];
} Students_file1;

typedef struct Students_file2
{
    char last_name[N];
    char first_name[N];
    int Score;
} Students_file2;

int getstudfile_1(char *string, Students_file1 *Studfrom1, int num)
{
    char *last_name = strtok(string, ";");
    char *first_name = strtok(NULL, ";");
    strcpy(Studfrom1[num].last_name, last_name);
    strcpy(Studfrom1[num].first_name, first_name);
    return 0;
}

int getstudfile_2(char *string, Students_file2 *Studfrom2, int num)
{
    char *last_name = strtok(string, ";");
    char *first_name = strtok(NULL, ";");
    char *Score = strtok(NULL, ";");
    strcpy(Studfrom2[num].last_name, last_name);
    strcpy(Studfrom2[num].first_name, first_name);
    Studfrom2[num].Score = atoi(Score);
    return 0;
}

int getscore(Students_file2 Studfrom2, int NumOfStud)
{
    int i;
    for (i = 0; i < NumOfStud; i++)
    {
        return Studfrom2.Score;
    }
}

int Numofstud(FILE *file_1, FILE *file_2, Students_file1 *Studfrom1, Students_file2
*Studfrom2) {
    int n = 0, m = 0;
    char *string = (char *)malloc(200);
    while (!feof(file_1))
    {
        fscanf(file_1, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_1(string, Studfrom1, n) == 0)
                n++;
            string[0] = '\0';
        }
    }
    while (!feof(file_2))
    {
        fscanf(file_2, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_2(string, Studfrom2, m) == 0)
                m++;
            string[0] = '\0';
        }
    }
    return n;
}
```

```

    }
    }
    while (!feof(file_2))
    {
        fscanf(file_2, "%s", string);
        if (string[0] != '\0')
        {
            if (getstudfile_2(string, Studfrom2, m) == 0)
                m++;
            string[0] = '\0';
        }
    }
    free(string);
    if (n == m){
        return n;
    }
    else{
        printf("Fail with number of students");
    }
}

int main() {
    setlocale(LC_ALL, "");

    FILE* file_1 = fopen("cw/input_file_1.csv", "r");
    FILE* file_2 = fopen("cw/input_file_2.csv", "r");
    Students_file1 *Studfrom1 = (Students_file1 *)calloc(100, sizeof(Students_file1));
    Students_file2 *Studfrom2 = (Students_file2 *)calloc(100, sizeof(Students_file2));
    if (file_1 == NULL) {
        printf("Fail with input_file_1\n");
        return 0;
    }
    else if (file_2 == NULL) {
        printf("Fail with input_file_2\n");
        return 0;
    }
    int NumOfStud = Numofstud( file_1, file_2, Studfrom1, Studfrom2);

    int max = 0, count = 0, i = 0;
    for ( i; i<NumOfStud; i++)
    {
        int studsscore = getscore(Studfrom2[i], NumOfStud);

        if (studsscore > max)
        {
            max = studsscore;
            count = 1;
        }
        else if (studsscore == max)
            count++;
    }

    printf("Кол-во студентов с высшим баллом = %d\n", count);
    fclose(file_1);
    fclose(file_2);
    free(Studfrom2);
    free(Studfrom1);

    return 0;
}

```

}