



# **Итоговая работа по курсу «Программирование на языке C (базовый уровень)»**

**ФИО Студента: Коноплев Д.А.**

# Оглавление

Постановка задания

Описание программы

- Исходный код программы

- Описание файлов программы

- Описание работы программы

  - Примеры написания командной строки для запуска:

  - Демонстрация работы программы

  - Сборка программы утилитой make

# Постановка задания

## Создание консольного приложения – Статистика температуры

Необходимо реализовать консольное приложение, которое осуществляет считывание текстового файла csv, состоящего из строк следующего формата:

**YEAR;MONTH;DAY;HOUR;MINUTE;TEMPERATURE**

dddd;mm;dd;hh;mm;temperature

dddd - год 4 цифры

mm - месяц 2 цифры

dd - день 2 цифры

hh - часы 2 цифры

mm - минуты 2 цифры

temperature - целое число от -99 до 99

В архиве файле хранится статистика собранная датчиком температуры за 1 календарный год. Предполагается, что датчик собирал информацию не чаще чем 1 раз в минуту и сохранял в заданном формате каждое значение в текстовый файл с новой строки. В какой-то момент времени датчик мог не работать, тогда данные поэтому периоду могут отсутствовать. Пример входного файла:

YEAR	MONTH	DAY	HOUR	MINUTE	TEMPERATURE
2021	1	1	23	1	-5
2021	1	1	23	3	-6
2021	1	1	23	1	-7
2021	1	2	7	5	-10

## Требования к обработке данных

Необходимо вывести статистику по каждому месяцу, с учетом исходных

данных:

- среднемесячная температура
- минимальная температура в текущем месяце
- максимальная температура в текущем месяце

Также необходимо вывести статистику за год:

- среднегодовая температура
- минимальная температура
- максимальная температура

## **Требования к аргументам командной строки**

Приложение должно обрабатывать аргументы командной строки:  
минимальный набор поддерживаемых ключей:

- -h Описание функционала приложения. Список ключей, которые обрабатывает данное приложение и их назначение.
- -f <filename.csv> входной файл csv для обработки.
- -m <номер месяца> если задан данный ключ, то выводится только статистика за указанный месяц.
- если нет параметров, то выдается help

## **Требования к ошибкам в входных данных**

- Приложение должно корректно работать на любых входных данных, если формат csv файла не соответствует заданному, то необходимо указать номер строки файла csv, в которой обнаружена ошибка и не учитывать данную строку.

В архиве с заданием [temperature\\_data\\_examples.zip](#) лежат два файла:

- temperature\_big.csv – файл со статистикой за год
- temperature\_small.csv – укороченный файл с ошибками для тестирования

## **Требования к сборке приложения**

- Приложение должно собираться при помощи утилиты make.
- Все прототипы функций, используемые в приложении, должны быть вынесены в отдельный файл temp\_functions.h
- Тексты функций – в файл temp\_functions.c
- Для реализации приложения рекомендуется использовать массив из структурного типа данных для хранения показаний датчика.

# Описание программы

## Исходный код программы

[https://github.com/dim-DPO-IUS/home\\_c/tree/main/Course](https://github.com/dim-DPO-IUS/home_c/tree/main/Course)

## Описание файлов программы

Программа состоит из следующих файлов:

**report.c** – Главный модуль программы:

- Точка входа (main())
- Обработка аргументов командной строки
- Управление workflow программы

**temp\_function.h** – файл прототипов функций программы;

Реализация бизнес-логики:

- Парсинг CSV
- Работа с двусвязным списком
- Расчет статистики
- Сортировка данных

**temp\_function.c** – Заголовочный файл:

- Объявления структур (sensor, temp\_stats и др.)
- Прототипы всех функций
- Константы и макросы

**Makefile** – файл инструкция для утилиты сборки make;

**report** – собранный исполняемый файл программы.

### Дополнительные файлы (генерируемые)

**load\_errors.log** - Лог-файл с ошибками парсинга CSV. Создается автоматически при обработке файлов

## Описание работы программы

Консольная утилита для обработки CSV-файлов с данными датчиков температуры. Рекомендуется запускать её из командной строки.

Хранение данных:

- Двусвязный список (добавление, удаление, сортировка)

- Оптимизированные структуры с битовыми полями для даты

Парсинг CSV:

- Валидация формата данных (год, месяц, день, температура)

- Обработка високосных лет

- Логирование ошибок в **load\_errors.log**

Поддерживает аргументы командной строки:

- h - справка

- f <file> - обязательный параметр (файл данных)

- m <month> - фильтр по месяцу

- p <N> - вывод N записей

- s <d|t> - сортировка (по дате/температуре)

При запуске без указания ключей, программа кратко выведет информацию о своем назначении и предложении ввести ключ «-h» для получения инструкций.

Примеры написания командной строки для запуска:

# Краткая информация о программе

```
./report
```

# Справка

```
./report -h
```

# Полная статистика

```
./report -f temperature_big.csv
```

# Статистика за апрель

```
./report -f temperature_small.csv -m 4
```

# Сортировка по температуре + вывод 5 записей

```
./report -f data.csv -s t -p 5
```

## Демонстрация работы программы

Рисунок 1. Запуск программы без опций

```
dim0k@dim0k-evm:~$ ./report
-----
Temperature Data Analysis Tool
-----
Features:
- CSV data import
- Temperature statistics analysis
- Custom period filtering
- Data sorting options
- Automatic error logging to 'load_errors.log'

Usage: program -f <filename.csv> [options]

Required option:
-f <file>    Specify input CSV file (required)

Analysis options:
-m <month>   Show stats for specific month (1-12)
-m <month>   Show stats for all month (0)

Data output options:
-p <N>       Print first N records from dataset
-s <criteria> Sort data (d - by date, t - by temperature)

Other options:
-h           Show this help message

Error Handling:
- Errors are logged to 'load_errors.log'
- Invalid values are automatically skipped
- Empty lines and comments are ignored
```

Рисунок 2. Запуск программы с опцией «-h»

```
dim0k@dim0k-evm:~$ ./report -h
-----
Temperature Data Analysis Tool.
-----
Available options:
-h          Show this help message
-f <file>   Specify input CSV file
-m <month>  Show statistics for specific or all month (1-12 or 0)
-p <N>      Print first N records from the dataset
-s <criteria> Sort data (d - by date, t - by temperature)
```

Рисунок 3. Запуск программы с указанием имени файла

```
dim0k@dim0k-evm:~$ ./report -f temperature_big.csv

File Loading Statistics:
-----
Parameter      | Value
-----
Source file     | temperature_big.csv
Total lines     | 518400
Valid records   | 515520
Empty lines     | 0
Invalid values  | 2880
Log file        | load_errors.log

Monthly Temperature Statistics:
-----
Year | Month | Count | Avg Temp | Max Temp | Min Temp
-----
2021 | 1     | 43200 | 15.5     | 30       | 1
2021 | 2     | 40320 | 14.5     | 28       | 1
2021 | 3     | 43200 | 15.5     | 30       | 1
2021 | 4     | 43200 | 15.5     | 30       | 1
2021 | 5     | 43200 | 15.5     | 30       | 1
2021 | 6     | 43200 | 15.5     | 30       | 1
2021 | 7     | 43200 | 15.5     | 30       | 1
2021 | 8     | 43200 | 15.5     | 30       | 1
2021 | 9     | 43200 | 15.5     | 30       | 1
2021 | 10    | 43200 | 15.5     | 30       | 1
2021 | 11    | 43200 | 15.5     | 30       | 1
2021 | 12    | 43200 | 15.5     | 30       | 1

Yearly statistics:
-----
Year | Count | Avg Temp | Max Temp | Min Temp
-----
2021 | 515520 | 15.4     | 30       | 1
```



Рисунок 4. Лог файл load\_errors.log (файл очень большой, поэтому показан только конец файла)

```
load_errors.log X
Line 86385: Invalid data - 2021; 2;30;23;44;30
Line 86386: Invalid data - 2021; 2;30;23;45;30
Line 86387: Invalid data - 2021; 2;30;23;46;30
Line 86388: Invalid data - 2021; 2;30;23;47;30
Line 86389: Invalid data - 2021; 2;30;23;48;30
Line 86390: Invalid data - 2021; 2;30;23;49;30
Line 86391: Invalid data - 2021; 2;30;23;50;30
Line 86392: Invalid data - 2021; 2;30;23;51;30
Line 86393: Invalid data - 2021; 2;30;23;52;30
Line 86394: Invalid data - 2021; 2;30;23;53;30
Line 86395: Invalid data - 2021; 2;30;23;54;30
Line 86396: Invalid data - 2021; 2;30;23;55;30
Line 86397: Invalid data - 2021; 2;30;23;56;30
Line 86398: Invalid data - 2021; 2;30;23;57;30
Line 86399: Invalid data - 2021; 2;30;23;58;30
Line 86400: Invalid data - 2021; 2;30;23;59;30
=== Loading completed ===
Total records: 515520
Errors: 2880
```

Рисунок 5. Запуск программы с указанием файла и опцией выбора месяца №1

```
dim0k@dim0k-evm:~$ ./report -f temperature_small.csv -m 1
Temperature statistics for month No. 1:
-----
Year | Month | Count | Avg Temp | Max Temp | Min Temp
-----|-----|-----|-----|-----|-----
2021 | 1 | 3 | -44.7 | -43 | -47
```

Рисунок 5. Запуск программы с указанием файла и опцией выбора месяца №0

```
dim0k@dim0k-evm:~$ ./report -f temperature_small.csv -m 0
```

Monthly Temperature Statistics:

Year	Month	Count	Avg Temp	Max Temp	Min Temp
2021	1	3	-44.7	-43	-47
2021	2	2	-27.5	-25	-30
2021	3	1	-10.0	-10	-10
2021	4	1	0.0	0	0
2021	5	1	10.0	10	10
2021	6	1	25.0	25	25
2021	7	1	30.0	30	30
2021	8	1	20.0	20	20
2021	9	1	18.0	18	18
2021	10	1	2.0	2	2
2021	11	1	-5.0	-5	-5
2021	12	1	-20.0	-20	-20

Рисунок 6. Запуск программы с указанием имени файла и опцией печати первых 10 строк списка в порядке файла

```
dim0k@dim0k-evm:~$ ./report -f temperature_small.csv -p 10
```

Printing 10 list values:

№	Date	Time	Temp
1	2021-01-16	01:01	-47°C
2	2021-01-16	01:03	-44°C
3	2021-01-16	01:04	-43°C
4	2021-02-16	01:01	-25°C
5	2021-02-17	01:01	-30°C
6	2021-03-16	01:01	-10°C
7	2021-04-16	01:01	0°C
8	2021-05-16	01:01	10°C
9	2021-06-16	01:01	25°C
10	2021-07-16	01:01	30°C

Рисунок 7. Запуск программы с указанием имени файла и опциями сортировки по температуре и печати первых 10 строк списка

```
dim0k@dim0k-evm:~$ ./report -f temperature_small.csv -s t -p 10
```

Printing 10 list values:

#	Date	Time	Temp
1	2021-01-16	01:01	-47°C
2	2021-01-16	01:03	-44°C
3	2021-01-16	01:04	-43°C
4	2021-02-17	01:01	-30°C
5	2021-02-16	01:01	-25°C
6	2021-12-16	01:01	-20°C
7	2021-03-16	01:01	-10°C
8	2021-11-16	01:01	-5°C
9	2021-04-16	01:01	0°C
10	2021-10-16	01:01	2°C

## Сборка программы утилитой make

### Сборка программы утилитой make

```
• dim0k@dim0k-evm:/mnt/DATA-1/0Dev/MC-Dev/0-LESSONS/01/src/work4dak/home_c/Course$ make clean
rm -f *.i *.s *.o report
• dim0k@dim0k-evm:/mnt/DATA-1/0Dev/MC-Dev/0-LESSONS/01/src/work4dak/home_c/Course$ make
gcc -g -std=c99 -Wall -E report.c -o report.i
gcc -g -std=c99 -Wall -S report.i -o report.s
gcc -g -std=c99 -Wall -g3 -c report.c
gcc -g -std=c99 -Wall -E temp_function.c -o temp_function.i
gcc -g -std=c99 -Wall -S temp_function.i -o temp_function.s
gcc -g -std=c99 -Wall -g3 -c temp_function.c
gcc -o report report.o temp_function.o
• dim0k@dim0k-evm:/mnt/DATA-1/0Dev/MC-Dev/0-LESSONS/01/src/work4dak/home_c/Course$
```