

# Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Любарский Иван Владимирович, № по списку 8

Контакты ivanred289@gmail.com, @rMeDGranD

Работа выполнена: «7» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «    » \_\_\_\_\_ 20\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

- Тема:** Обработка последовательности литер входного текстового файла.
- Цель работы:** Составить программу на языке Си для анализа и обработки вводимого текста.
- Задание:** Перевести все мерные температуры из шкалы Цельсия в шкалу Фаренгейта. (Вариант 18)
- Оборудование:**  
Процессор *Intel Core i5-4210U @ 4x 1.7GH* с ОП *15873* Мб, НМД *512* Гб. Монитор *1600x900*
- Программное обеспечение:**  
Операционная система семейства: *linux*, наименование: *ubuntu*, версия *22.04 LTS*  
интерпретатор команд: *bash* версия *4.4.19*.  
Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*  
Утилиты операционной системы --  
Прикладные системы и программы --  
Местонахождение и имена файлов программ и данных на домашнем компьютере --

**6. Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Для выполнения данной лабораторной работы следует воспользоваться моделью конечного автомата (при помощи switch case в Си). При помощи состояний автомата и переходов надо найти и выделить все мерные температуры в отдельных словах и преобразовать их в температуру по Фаренгейту (умножить на 1.8 и прибавить 32).

Для выполнения понадобятся состояния поиска отдельного слова, определения числа и нахождения размерности температуры. Также число может быть дробным, так что нужно добавить состояние считывания дробной части.

После считывания всех данных и преобразований остается только вывести готовый результат.

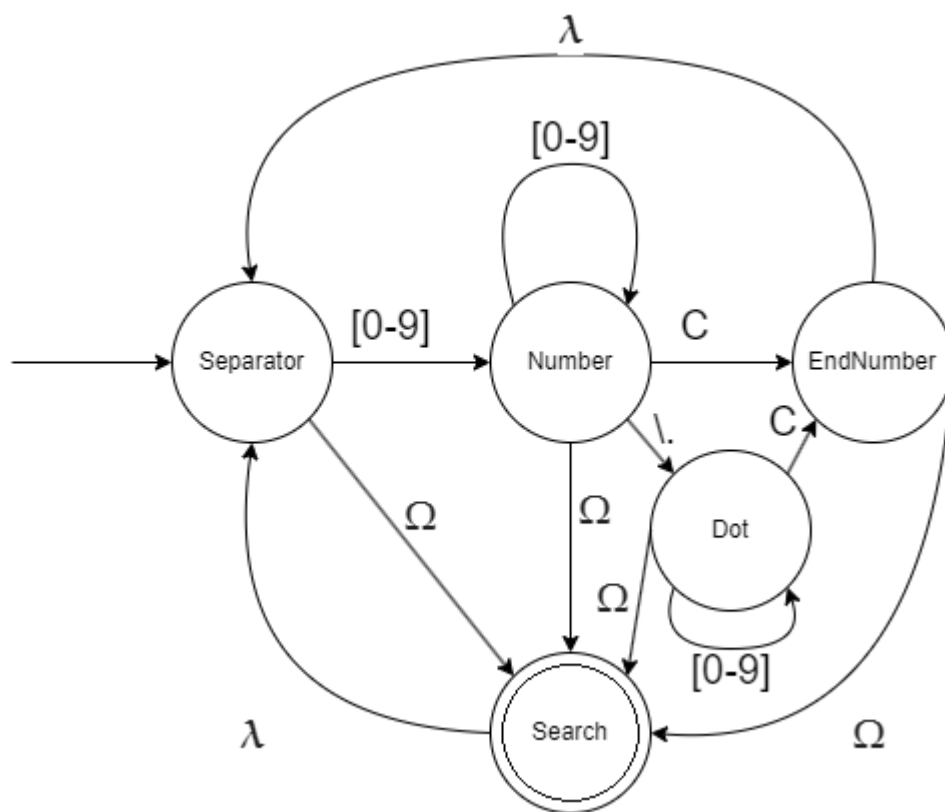
**7. Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

## План работы программы:

- Поиск разделителя
- Считывание числа
- Проверка на дробную часть
- Проверка на правильность слова
- Преобразование числа
- Вывод температуры по Фаренгейту
- Вернуться к 1 пункту

Входные данные	Выходные данные	Описание тестируемого случая
100C	3	Преобразование простейшей температуры
100.1C	-1	Преобразование дробной температуры
100.0C	-2	Преобразование дробной температуры с нулевой дробной частью
.0C	Вывода нет	Попытка преобразования температуры без целой части.

Диаграмма конечного автомата:



**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
#include <stdio.h>
#include <assert.h>

typedef enum {
Search, //Состояние поиска разделителя
Separator, //Состояние проверки следующего за разделителем символа (число или нет?)
Number, //Состояние считывания целой части числа
Dot, //Состояние считывания дробной части числа
EndNumber //Состояние проверки на следующий за C символ (разделитель или нет) и вывод ответа
} State;

int is_number(char c){
if(('0'<=c)&&(c<='9')){ return 1; }
return 0;
}

int is_C(char c){
if(c=='C'){ return 1; }
return 0;
}

int is_separator(char c){
if((c==' ')||(c=='\n')||(c==',')||(c=='\t')){ return 1; }
return 0;
}

int is_dot(char c){
if(c=='.'){ return 1; }
return 0;
}

void test(){

assert(is_number('0'));
assert(is_C('C'));
assert(is_separator(' '));
assert(is_dot('.'));

assert(!is_number('C'));
assert(!is_C('1'));
assert(!is_separator('/'));
assert(!is_dot('g'));

}

int main(){

test();
```

```
double temp, dot, z; //Температура (целая часть), дробная часть, кол-во разрядов в dot
dot = temp = z = 0;
char c;

State state = Separator;

while((c=getchar())!=EOF){

switch (state){

case Search:
if(is_separator(c)){ state = Separator; } //Поиск разделителя
break;

case Separator:
if(is_number(c)){ //Проверка на число
temp = (c - '0');
state = Number;
}
else { state = Search; }
break;

case Number:
if(is_number(c)){ //Считывание числа
temp *= 10;
temp += (c - '0');
}
else if(is_C(c)){ //Окончание числа
temp = temp*18/10 + 32;
state = EndNumber;
}
else if(is_dot(c)){ //Проверка на наличие дробной части
temp = temp*18/10 + 32;
z = 10; dot = 0;
state = Dot;
}
else { state = Search; }
break;

case EndNumber:
if(is_separator(c)){ //Проверка на полноценность слова
printf("%g%c%c", temp, 'F', ' ');
temp = 0;
dot = 0;
state = Separator;
}
else { state = Search; }
break;
```

```
case Dot:
if(is_number(c)){ //Считывание дробной части
z*=10;
dot *= 10;
dot += (c - '0');
}
else if(is_C(c)){ //Проверка на окончание слова
dot = dot*18;
temp += dot/z;
if(z==10){ state = Search; } //Проверка на наличие дробной части
else { state = EndNumber; }
}
else { state = Search; }
break;
}
}
return 0;
}
```

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Вре м я	Событие	Действие по исправлению	Примечание

#### **10. Замечания автора** по существу работы

#### **11. Выводы**

На основе данной лабораторной работы, а точнее на основе конечного автомата, можно понять и в будущем использовать алгоритм анализа и обработки текста, выделения слов, проверки на правильность составленного текста.

Полученный навык может понадобиться для работы с большим количеством текста, где намного легче будет использовать автоматизированную проверку и исправление.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_