|  |  |
| --- | --- |
|  | **Отчёт по лабораторной работе** №11 по курсу Основы информатики ­  студента группы М08-107Б-18 Хренова Геннадия Николаевича , № по списку 23  Адреса www, e-mail, jabber, skype [khrenov.gena@yandex.ru](mailto:khrenov.gena@yandex.ru)  Работа выполнена: “23“ ноября 2018 г.  Преподаватель: аспирант каф.806 Ридли А. Н.  Входной контроль знаний с оценкой  Отчёт сдан “ “ 20 г., итоговая оценка  Подпись преподавателя |

1. **Тема**: Обработка последовательности литер входного текстового файла. Простейшие приёмы лексического анализа. Диаграммы состояний переходов.
2. **Цель работы**: Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем вариантом задания.
3. **Задание** (*вариант №* 18 ): Перевести все мерные температуры из шкалы Цельсия в шкалу Фаренгейта.
4. **Оборудование** (*лабораторное*):

ЭВМ Pentium , процессор 2.7 ГГц , имя узла сети с ОП 4096 МБ

НМД 2 ГБ. Терминал адрес . Принтер

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор intel core i3 , ОП 3072 , НМД 50 ГБ. Монитор

Другие устройства

1. **Программное обеспечение** (*лабораторное*):

Операционная система семейства Unix , наименование версия

Интерпретатор команд bash версия

Система программирования версия

Редактор текстов Emacs версия

Утилиты операционной системы

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix , наименование Ubuntu версия 18.04LTS

Интерпретатор команд bash версия

Система программирования версия

Редактор текстов Emacs версия

Утилиты операционной системы

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

1. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)
2. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

Начинаем записывать слово в переменную, если оно состоит из одних цифр. Если встречается другой символ — обнуляем переменную и идём к следующему слову. Чтобы разряды не смешивались, предыдущее значение умножается на 10. Если после цифр идёт знак «С» (по Цельсию),

а далее следует конец слова, то в этом случае умножаем наше число на 9/5 и прибавляем 32, затем выводим его с символом «F»(по Фаренгейту). Во всех остальных случаях обнуляем переменные и переходим к следующему слову.

*Пункты 1-7 отчёта составляются* ***строго до*** *начала лабораторной работы.*

Допущен к выполнению работы. Подпись преподавателя

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

gennadii@lenovo-b560:~/Рабочий стол$ cat 11.c

#include <stdio.h>

typedef enum {NUMBER, SIGN, SPACE, OTHER, ifC} State;

int is\_digit(char n)

{

return (n >= '0' && n <= '9');

}

int is\_sign(char n)

{

return (n == '+' || n == '-');

}

int is\_space(char n)

{

return (n == ' ' || n == '\n' || n == ';');

}

int C(char n)

{

return (n == 'C');

}

long long roundF(long long a)

{

return(((a \* 9) % 5) <= 2 ? a \* 9 / 5 : a \* 9 / 5 + 1);

}

int main(void)

{

char c;

long long l = 0;

long long k = 0;

int j = 1;

State St = SPACE;

do {

c = getchar();

switch (St) {

case SIGN:

if (is\_space(c)) {

St = SPACE;

j = 1;

} else if (is\_digit(c)) {

l = c - '0';

St = NUMBER;

} else {

St = OTHER;

j = 1;

}

break;

case SPACE:

if (is\_digit(c)) {

l = c - '0';

St = NUMBER;

} else if (is\_sign(c)) {

if (c == '-') {

j = -1;

}

if (c == '+') {

j = 1;

}

St = SIGN;

} else {

St = OTHER;

}

break;

case NUMBER:

if (is\_digit(c)) {

l = l \* 10 + (c - '0');

}

if (is\_space(c)) {

St = SPACE;

l = 0;

j = 1;

}

if (C(c)) {

St = ifC;

k = l;

} else if (!is\_digit(c) && !is\_space(c)) {

St = OTHER;

l = 0;

j = 1;

}

break;

case ifC:

if (is\_space(c)) {

k = roundF(k) \* j + 32;

if (c == '\n') {

printf("%lldF\n", k);

} else {

printf("%lldF ", k);

}

j = 1;

St = SPACE;

} else {

St = OTHER;

k = 0;

j = 1;

}

break;

case OTHER:

if (is\_space(c)) {

St = SPACE;

}

break;

}

} while (c != EOF);

printf("\n");

return 0;

}gennadii@lenovo-b560:~/Рабочий стол$ gcc 11.c

gennadii@lenovo-b560:~/Рабочий стол$ gcc -pedantic 11.c

gennadii@lenovo-b560:~/Рабочий стол$ ./a.out

100C 0C qwer 1223 123c

212F 32F

gennadii@lenovo-b560:~/Рабочий стол$ ./a.out

-204C -1222 +0 -0

-335F

gennadii@lenovo-b560:~/Рабочий стол$ ./a.out

+0C -0C

32F 32F

gennadii@lenovo-b560:~/Рабочий стол$

**9.Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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 | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |

1. Замечание автора по существу работы

Вывод: Конечный автомат — это некоторая абстрактная модель, содержащая конечное число состояний чего-либо. Используется для представления и управления потоком выполнения каких-либо команд. Он может быть легко представлен в виде графа, что позволяет увидеть все возможные варианты и легко разобраться в коде.ыы

