

# Отчет по лабораторной работе №11 по курсу фундаментальная информатика

Студент группы М8О-101Б-20 Ядров Артем Леонидович, № по списку 28

Контакты www, e-mail, icq, skype temayadrow@gmail.com

Работа выполнена: «    » \_\_\_\_\_ 202\_\_ г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приемы лексического анализа. Диаграммы состояний и 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 № 20):** Перевести все мерные расстояния из миль (mi) в километры (km).

4. **Оборудование (лабораторное):**  
ЭВМ Intel Pentium G2140, процессор 3.30 GHz, имя узла сети Cameron с ОП 8096 Мб,  
НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel core i5 2.60 GHz с ОП 8096 Мб, НМД 131072 Мб. Монитор dell  
Другие устройства \_\_\_\_\_

5. **Программное обеспечение (лабораторное):**  
Операционная система семейства Unix, наименование Ubuntu версия 18.15.0  
интерпретатор команд bash версия 4.4.20  
Система программирования GNU версия 5.8.13  
Редактор текстов emacs версия 25.2.2  
Утилиты операционной системы gcc, cat, gdb

Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных stud/208104

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix, наименование Fedora версия 32  
интерпретатор команд bash версия 5.0.17  
Система программирования Clion версия 2020.2.1  
Редактор текстов emacs версия 25.2.2  
Утилиты операционной системы gcc, cat, gdb

Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных на домашнем компьютере home/Temich

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

1. Будем обрабатывать только слова, начинающиеся с цифр. Введем 3 состояния: «OUT» (находимся вне обрабатываемого слова), IN\_DIGIT (находимся в обрабатываемом слове) и «CONTINUE» (находимся в слове, которое необходимо пропустить)
2. В состоянии «OUT» мы проверяем введенный символ на то, является ли он цифрой или знаком „-“. Если является цифрой или знаком „-“, то переходим в состояние «IN\_DIGIT» и меняем соответствующую переменную «otr».
3. В состоянии «IN\_DIGIT» мы преобразуем число из символьного типа к типу int. Также проверяем на то, будет ли выводимый результат и наше число находиться в типе int. Если мы прочли не цифру, то проверяем дальнейшие два символа на равенство «mi». Далее вновь считываем символ, он должен быть разделителем. Если считанный символ разделитель, то выводим результат. Если же нет — переходим в состояние «CONTINUE». Так же мы переходим в это состояние если после цифр не следует «mi» или число выходит за границы типа int.
4. В состоянии «CONTINUE» мы проверяем считанный символ на то, является ли он разделителем. Если является разделителем, то переходим в состояние «OUT» и выводим пробел и переводим строку (в случае конца строки).

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

Тесты:

Input: -1mi 1mi 1334669761mi 999999999mi 1mimi sadfads 0mi 213123321mi  
999999999999mi 123.40000mi, dasfsadf

Output: -1km 1km 2147483645km 1608999998km 1mimmi sadfads 0km 342915423km  
999999999999mi 123.40000mi, dasfsadf

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя \_\_\_\_\_

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

```
[Temich@localhost laabs]$ cat lab11.c
```

```
/* Лабораторная работа №11. Вариант 20.
```

```
* Перевести все мерные расстояния из миль (mi) в километры (km).
```

```
* Студент группы 08-101 А.Ядров*/
```

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

```
#define CONTINUE 2
```

```
#define IN_DIGIT 1
```

```
#define OUT 0
```

```
#define MAX_INT 2147483647
```

```
int to_km(int mi) { //функция перевода из миль в километры
    return (int) (mi * 1.609);
}
```

```
int is_digit(int c) { //проверка введенного символа на то, является ли он цифрой или нет
    if (c >= '0' && c <= '9') {
        return 1;
    } else {
        return 0;
    }
}
```

```
int main() {
    int c, n = 0, otr = 0; //c - считываемый символ, n - значение считываемого числа
    int state = OUT; //состояние
    while ((c = getchar()) != EOF) {
        if (state == OUT) { //для состояния "OUT"
            if (is_digit(c)) { //переход в состояние "IN_DIGIT"
                state = IN_DIGIT;
            } else {
                if (c == '-') {
                    otr = 1;
                    state = IN_DIGIT;
                    continue;
                }
                if (c != ' ' && c != '\t' && c != '\n') {
                    state = CONTINUE; //переход в состояние "CONTINUE"
                }
            }
        }
        if (state == IN_DIGIT) { //для состояния "IN_DIGIT"
            if (is_digit(c)) {
                if (n < MAX_INT / 10 - (c - '0')) { //проверка на переполнения типа int
                    n = n * 10 + (c - '0'); //добавление цифры в конец числа
                    if (otr) {
                        if (n > 0) {
                            n *= -1;
                        }
                    }
                } else {
                    state = CONTINUE; //переход в состояние "CONTINUE" в случае переполнения
                    printf("%d%c", n, c);
                    n = 0;
                    otr = 0;
                    continue;
                }
            }
            if (n > (int) MAX_INT / 1.609) { //проверка на переполнения типа при переводе
                state = CONTINUE;
                printf("%d", n);
                n = 0;
                otr = 0;
                continue;
            }
        } else if (c != 'm') { //проверка следующих символов после числа
            state = CONTINUE;
        }
    }
}
```

```

    printf("%d", n);
    n = 0;
} else {
    if ((c = getchar()) !=
        'i') { //после символа 'm' должен следовать символ 'i'. Иначе просто выводим все слово
        state = CONTINUE;
        if (c == '\n' || c == '\t' || c == ' ') {
            state = OUT;
            printf("%d%c ", n, 'm');
            if (c == '\n') {
                printf("%c", '\n');
            }
        } else {
            printf("%d%c%c", n, 'm', c);
        }
        n = 0;
        otr = 0;
    } else {
        c = getchar();
        if (c == '\n' || c == ' ' || c == '\t') {
            state = OUT;
            printf("%dkm ", to_km(n)); //перевод числа из миль в километры
            n = 0;
            otr = 0;
            if (c == '\n') {
                printf("%c", '\n');
            }
        } else {
            state = CONTINUE;
            printf("%d%s%c", n, "mi", c); //случай, если после "mi" следуют какие-то символы
            n = 0;
            otr = 0;
        }
    }
}
}
}
if (state == CONTINUE) {
    if (c == '\n' || c == ' ' || c == '\t') {
        state = OUT;
        printf("%c", ' ');
        if (c == '\n') {
            printf("%c", '\n');
        }
        continue;
    }
    printf("%c", c);
}
}
return 0;
}[Temich@localhost laabs]$ gcc lab11.c
[Temich@localhost laabs]$ ./a.out
-1mi 1mi 1334669761mi 999999999mi 1mimi sadfads 0mi 213123321mi
-1km 1km 2147483645km 1608999998km 1mimmi sadfads 0km 342915423km
999999999999mi 123.40000mi, dasfsadf
999999999999mi 123.40000mi, dasfsadf
^C
[Temich@localhost laabs]$

```

**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	Время	Событие	Действие по исправлению	Примечание

**10. Замечания автора** по существу работы \_\_\_\_\_

---



---



---

**11. Выводы**

Я научился составлять программы, написанные на Си, для анализа и обработки вводимого текста.

---



---



---



---



---

Недочёты при выполнении задания могут быть устранены следующим образом:

---



---



---

Подпись студента \_\_\_\_\_