

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

Студент группы М80-109Б-22 Степанов Алексей Николаевич, № по списку 18

Контакты aleksey.stepanov2004@mail.ru, telegram @Alex1stepa

Работа выполнена: «31» октября 2022г.

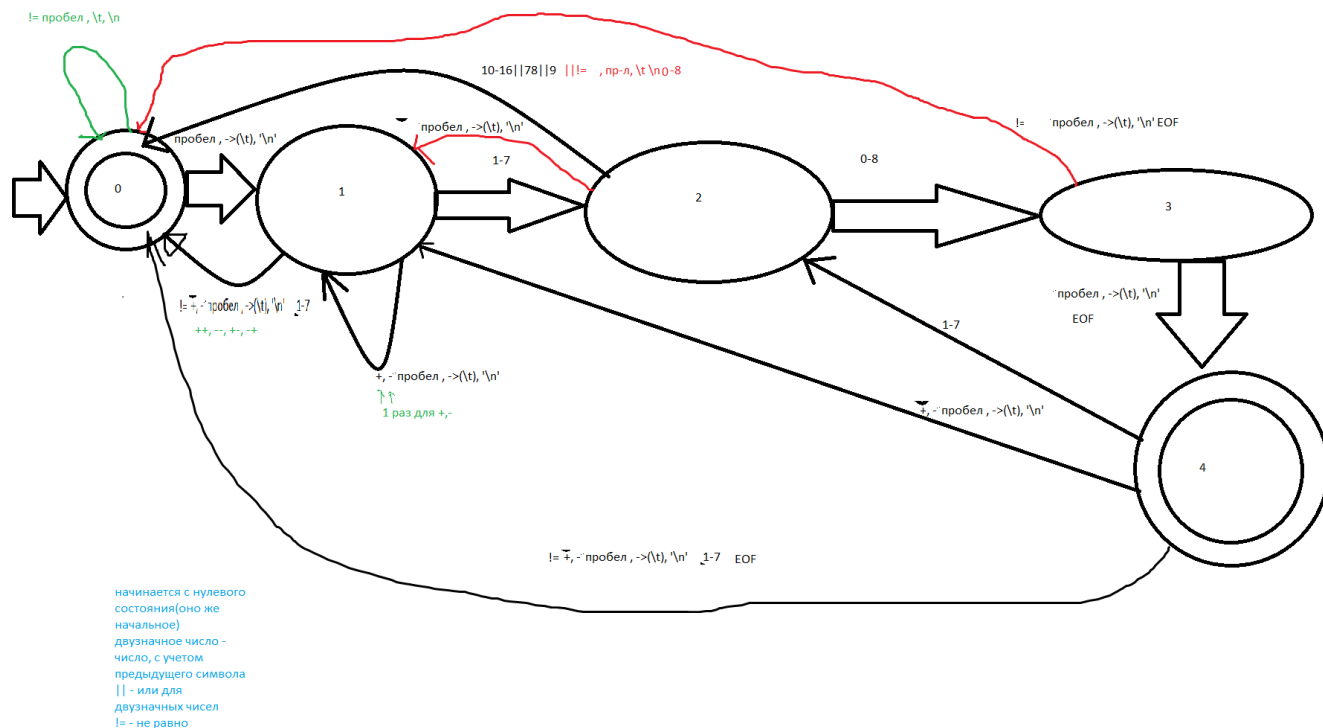
Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Обработка последовательности литер входного текстового файла. Диаграммы состояний и переходов.
2. **Цель работы:** Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем варианта задания.
3. **Задание (вариант 37):** Выделить все девятиричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-итальянски.
4. **Оборудование** (студента):
Процессор *Intel Core i5-8265U @ 8x 3.9GH* с ОП 7851 Мб, НМД 1024 Гб. Монитор 1920x1080
5. **Программное обеспечение** (студента):
Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*
интерпретатор команд: *bash* версия *4.4.19*.
Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*
Утилиты операционной системы --
Прикладные системы и программы: *VTM(QT)*
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):
Используя функцию `getchar`, находящуюся в библиотеке `stdio.h`, мы можем получать литеры из стандартного потока ввода(`stdin`), в котором мы и будем вводить сообщение. Для упрощения работы программиста и общей красоты конечного аппарата, введем дополнительное требование к вводу, чтобы перед сообщением стоял пробел, что обозначает начало файла. В дальнейшем мы будем считывать с помощью идеи конечного аппарата по 4(5) символов и, выявляя числа(конструкции типа : разделитель, возможно + или -, 1-7, 0-8,разделитель), подходящие для 9ой системы счисления и принадлежащие отрезку [17;77], после чего сразу будем выводить их названия в итальянском, разбивая их название на 2 смысловые части: название десятка без окончания и название единиц с предшествующим остатком числительного, соответствующего итальянской грамматики.(это сделано ради корректного вывода таких чисел, ка 21 или 28(*uno* и *otto*, не требующих буквы а для разграничения 2 частей слова).

Схема конечного автомата:



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

"Проводилось unit-тестирование, исходный код тестов приложен в пункте №8"

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

Код программы:

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
typedef enum{
    status_zero, //0
    status_first, // 1
    status_second, // 2
    status_third, // 3
    status_fourth // 4

```

```
}status;
```

//для нормального функционирования нашего конечного аппарата нам потребуются:

// проверки на всевозможные разделители

```
int is_it_lda(char c){
    if ( c == ' ')
        return 1;
    else
        return 0;
}

```

```
void test_lda(){
    char c=' '; //1 test - correct +
    assert(is_it_lda(c)==1);
    c='r'; // 2 test - incorrect +
    assert(is_it_lda(c)==0);
    c=EOF; // 3 test - EOF +
    assert(is_it_lda(c)==0);
}

```

```
}
```

```

int is_it_cmma(char c){
    if ( c == ',')
        return 1;
    else
        return 0;

}

void test_cmma(){
    char c=','; //1 test - correct +
    assert(is_it_cmma(c)==1);
    c='r'; // 2 test - incorrect +
    assert(is_it_cmma(c)==0);
    c=EOF; // 3 test - EOF +
    assert(is_it_cmma(c)==0);

}

int is_it_tab(char c){
    if ( c == '\t')
        return 1;
    else
        return 0;

}

void test_tab(){
    char c='\t'; //1 test - correct +
    assert(is_it_tab(c)==1);
    c='r'; // 2 test - incorrect +
    assert(is_it_tab(c)==0);
    c=EOF; // 3 test - EOF +
    assert(is_it_tab(c)==0);

}

int is_it_n(char c){
    if ( c == '\n')
        return 1;
    else
        return 0;

}

void test_n(){
    char c='\n'; //1 test - correct +
    assert(is_it_n(c)==1);
    c='\t'; // 2 test - incorrect +
    assert(is_it_n(c)==0);
    c=EOF; // 3 test - EOF +
    assert(is_it_n(c)==0);

}

int is_it_num_9(char c){
    if(c >= '0'&&c <= '8')
        return 1;
    else
        return 0;

}

```

```

int is_it_plus_or_minus(char c){
    if( c == '+' || c == '-')
        return 1;
    else
        return 0;
}

void test_is_it_num_9(){
    char c='0'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='1'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='2'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='3'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='4'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='5'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='6'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='7'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='8'; //1 test - correct +
    assert(is_it_num_9(c)==1);
    c='9'; // 2 test - incorrect +
    assert(is_it_num_9(c)==0);
    c='\t'; // 3 test - incorrect +
    assert(is_it_num_9(c)==0);
    c=EOF; // 4 test - EOF +
    assert(is_it_num_9(c)==0);
}

void test_is_it_pm(){
    char c='+';
    assert(is_it_plus_or_minus(c)==1);//1 test -correct +;
    c='-';
    assert(is_it_plus_or_minus(c)==1);//1 test -correct +;
    c='r';
    assert(is_it_plus_or_minus(c)==0);//2 test -incorrect +;
    c=EOF;
    assert(is_it_plus_or_minus(c)==0);//3 test -EOF +;
}

int numeri (char a, char b){
    if(is_it_num_9(a)&&is_it_num_9(b)){

switch(a){
    case '1':
        printf("dici");
        break;
    case '2':
        printf("vent");
        break;
    case '3':
        printf("trent");
        break;
    case '4':
        printf("quarant");
        break;

```

```

    case '5':
        printf("cinquant");
    break;
    case '6':
        printf("sessant");
    break;
    case '7':
        printf("settant");
    break;
    default:
        return 0;
    break;

}
switch(b){
    case '0':
        printf("a\n");
    break;
    case '1':
        printf("uno\n");
    break;
    case '2':
        printf("adue\n");
    break;
    case '3':
        printf("atre\n");
    break;
    case '4':
        printf("aquattro\n");
    break;
    case '5':
        printf("acinque\n");
    break;
    case '6':
        printf("asei\n");
    break;
    case '7':
        printf("asettant\n");
    break;
    case '8':
        printf("otto\n");
    break;
    default:
        return 0;
    break;
}
return 1;
}
else{
    return 0;
}
}
void test_numeri(){
    char a='4',b='0';
    assert(numeri(a,b)==1);
    a='r',b='8';
    assert(numeri(a,b)==0);
    a='7',b='3';
    assert(numeri(a,b)==1);
}

```

```

int main()
{
    status stateus=status_zero;
    char c=' ',a=' ',b=' ';
    int flag=0;//счетчик + и -
    test_lda();
    test_cmma();
    test_tab();
    test_is_it_num_9();
    test_n();
    test_is_it_pm();
    // test_numeri(); //комментарий, ибо мешает корректной работе - выводит результат тестирования на
    стандартный поток

```

```

while((c=getchar())!=EOF){ // условие завершения работы алгоритма
    switch(stateus){

```

```

        //_____
    case status_zero:
        if (is_it_lda(c)||is_it_n(c)||is_it_tab(c)||is_it_cmma(c)){
            stateus =status_first;
        }
        break;
        //_____

```

```

    case status_first:
        if(is_it_num_9(c)&& c!='0'&&c!='8'){
            a=c;
            stateus=status_second;
            flag=0;

        }
        else if (is_it_num_9(c)&&(c=='0'||c=='8')){
            stateus = status_zero;
            flag=0;
        }
        else if(is_it_plus_or_minus(c)&&flag==0){
            flag=1;
        }
        else if(is_it_plus_or_minus(c)&&flag==1){
            stateus =status_zero;
            flag=0;
        }
        else if (is_it_lda(c)||is_it_n(c)||is_it_tab(c)||is_it_cmma(c)){
            flag=0;
        }
        else {
            stateus = status_zero;
            flag=0;
        }
        break;
        //_____

```

```

    case status_second:
        if(is_it_num_9(c)&& !(a=='1'&&c<'7') && !(a=='7'&&c>'7')){
            b=c;
            stateus=status_third;
        }
        else
            stateus=status_zero;
        break;
        //_____

```

```

    case status_third:
        if (is_it_lda(c)||is_it_n(c)||is_it_tab(c)||is_it_cmma(c)){
            stateus =status_fourth;
        }

```

```

        numeri(a,b);
        a='';
        b='';
    }
    else
        stateus=status_zero;
    break;
// _____
case status_fourth:
    if(is_it_num_9(c)&&c!='0'&&c!='8'){
        a=c;
        stateus=status_second;
        flag=0;

    }
    else if (is_it_num_9(c)&&(c=='0'||c=='8')){
        stateus = status_zero;
        flag=0;
    }
    else if(is_it_plus_or_minus(c)&&flag==0){
        flag=1;
        stateus =status_first;
    }
    else if(is_it_plus_or_minus(c)&&flag==1){
        stateus =status_zero;
        flag=0;
    }
    else if (is_it_lda(c)||is_it_n(c)||is_it_tab(c)||is_it_cmml(c)){
        flag=0;
        stateus =status_first;
    }
    else{
        flag=0;
        stateus =status_zero;
    }
    break;

}
}
if(stateus==status_third){
    numeri(a,b);
    stateus=status_fourth;
}
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	Дата	Время	Событие	Действие по исправлению	Примечание
---	-------------	------	-------	---------	----------------------------	------------

	дом.					

10. Замечания автора по существу работы

Хотелось бы пояснения к формулировке задачи, чтобы не возникало вопросов вида:

Печатать ли значения этих чисел как десятичных либо переводить их в десятичную и уже тогда распечатывать их значения?

11. Выводы

От лабораторной работы получил исключительно положительные эмоции и впечатления. По моему мнению, знания, приобретенные мною на данной лабораторной работе, помогли мне лучше осознать принципы работы операционных систем, метода ввода и вывода ЯП СИ, отладчика СИ, более пристально изучить язык программирования Си, научиться отлаживать свой код и находить ошибки разного типа, использовать конечные автоматы для работы со строками(массивами литер), что несомненно поможет мне при решении практических задач.

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

Подпись студента _____