



Отчёт по лабораторной работе № 11 по курсу 1

студента группы M80-108Б-19 Хренниковой Ангелины, № по списку 23

Адреса www, e-mail, jabber, skype: lina.khrennikova@mail.ru

Работа выполнена: “ 3 ” декабря 2019г.

Преподаватель: Поповкин А. В. каф.806

Входной контроль знаний с оценкой

Отчёт сдан “3” декабря 2019 г., итоговая оценка

Подпись преподавателя

1. **Тема:** Обработка последовательности литер входного текстового файла. Простейшие приемы простого лексического анализа. Диаграммы состояний и переходов.

2. **Цель работы:** Составить программу на языке Си, выполняющую анализ и обработку видимого текста в соответствии с вариантом задания.

3. **Задание (вариант №):** Подсчитать количество положительных десятичных чисел в строке, допустимых 16-битными процессорами.

4. **Оборудование (лабораторное):**
ЭВМ PC, процессор Intel® Core™ i7-3770 CPU @ 3.40GHz * 8, имя узла сети alise18 с ОП 15974,4 МБ, НМД 345,5 Гб.
Терминал Gnome адрес 192.168.2.118/24. Принтер
Другие устройства

Оборудование ПЭВМ студента, если использовалось:

Процессор Intel® Core™ i3-7020U CPU @ 2.30GHz * 4, ОП 8192 МБ, НМД 256 Гб. Монитор LCD
Другие устройства

5. **Программное обеспечение (лабораторное):**
Операционная система семейства UNIX, наименование Ubuntu версия 18.04
Интерпретатор команд Bash версия 4.4.20(1)
Система программирования версия
Редактор текстов Nano версия 2.9.3
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 18.04
Интерпретатор команд Bash версия 4.4.19(1)
Система программирования версия
Редактор текстов Emacs версия 25.2.2
Утилиты операционной системы gcc

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

Программа, которая считает числа в строке, содержащей не только цифры:

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>

int main() {
    int k=0, f, t=0, j, m=5, max=32767, g=0, l;
    long int n=0;
    char s[10];
    FILE *fi;
    if((fi = fopen("test.txt", "r"))!=NULL) {
        while (!feof(fi)) {
            for(int i=0; i<10; i++) {
                fscanf(fi,"%s", &s[i]);
                if
(s[i]!='0'&&s[i]!='1'&&s[i]!='2'&&s[i]!='3'&&s[i]!='4'&&s[i]!='5'&&s[i]!='6'&&s[i]!='7'&&s[i]!='8'&&$
if ((k<m)&&(k!=0)) {
                    if (n!=0) {
                        // printf("n1=%ld ", n);
                        t++;
                        k=0;
                        n=0;}
                    else continue;
                }
                if ((k==m)&&(k!=0)) {
                    if (n<max/10) {
                        if (n!=0) {
                            // printf("n2=%ld ", n);
                            t++;
                            k=0;
                            n=0;}
                        else continue;
                    }
                    if (j<=(max%10)) {
                        if (n!=0) {
                            // printf("n3=%ld ", n);
                            t++;
                            k=0;
                            n=0;}
                        else continue;
                    }
                    if (j>(max%10))
                        continue;
                    k=0;
                    n=0;
                }
            }
            if (k==0) {
                // printf("k=0 ");
            }
            if (k>m) {k=0; n=0;}
            g=0;
        }
        if (s[i]=='-') {
            g=1;
        }
    }
}
```

```

        // printf("g=%d ", g);
        continue;
    }
    if (g==1) {
        continue;
    }

    if (g==0) {
        if (s[i]=='0'||s[i]=='1'||s[i]=='2'||s[i]=='3'||s[i]=='4'||s[i]=='5'||s[i]=='6'||s[i]=='7'||s[i]=='8'||$
j=(int)(s[i]-'0');
        k++;
        if (k>m) { g=1; k=0;}
        n=n*10+j;
        // printf("j=%d k=%d n4=%ld ", j, k, n);
    }
}

}
if ((s[9]=='0'||s[9]=='1'||s[9]=='2'||s[9]=='3'||s[9]=='4'||s[9]=='5'||s[9]=='6'||s[9]=='7'||s[9]=='8'||s[9]=='$
printf("%d\n ", t+1);
else
    printf("%d\n ", t);
}
fclose(fi);
}
else
    printf("\nФайл не найден!\n");
}

```

Программа, которая читает строку, содержащую только числа:

```

#include <stdio.h>
#include <stdbool.h>
#include <string.h>

int main()
{
    bool g;
    int t=0;
    int s[6];
    FILE *fi;
    if((fi = fopen("text.txt", "r"))!=NULL) {
        g=false;
        while (!feof(fi)) {
            for(int i=0;i<6;i++) {
                fscanf(fi,"%d", &s[i]);
                if (s[i]>=32768 && s[i]<=0) { continue;}
                else {t++; g=true;}
            }
        }
        if (g==false) {
            printf("В файле не найдено ни одного числа, подходящего под условие.\n");
        }
        else printf("%d", t/2);
        fclose(fi);
    }
    else
        printf("\nФайл не найден!");
}

```

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

№	Функция	Что делает?
1	isalpha	Функция isalpha проверяет аргумент, передаваемый через параметр character, является ли он строчной или прописной буквой алфавита.
2	isdigit	Функция isdigit проверяет аргумент, передаваемый через параметр character, является ли он десятичной цифрой.
3	isspace	Функция isspace проверяет параметр character, является ли он символом пробела. Обратите внимание на то, что символ пробела — это, на самом деле, несколько символов.
4	isalnum	Функция isalnum проверяет параметр character, через который передается целочисленное значение — эквивалент символа, является ли character десятичной цифрой или символом алфавита, прописной или строчной буквой. Если проверяемый параметр character десятичная цифра или буква верхнего или нижнего регистров, то результат выполнения функции — истина. В противном случае результат — ложь.
5	isprint	Функция isprint проверяет значение аргумента, передаваемого через параметр character, представляет ли оно собой печатный символ. Печатными символами являются все символы, кроме управляющих символов.
6	tolower	Функция tolower выполняет преобразование прописных букв в строчные. То есть, преобразует свой параметр в строчный эквивалент, если символ с заглавной буквы. Если, передаваемый символ итак строчный, то преобразование не выполняется и значение остаётся неизменным.
7	toupper	Функция toupper выполняет преобразование строчных букв в прописные. То есть, преобразует свой параметр в прописной эквивалент, если символ строчный. Если, передаваемый символ итак заглавный, то преобразование не выполняется и значение остаётся неизменным.

Если вводится строка, в которой используются не только цифры и знак пробела:

Ввод	Вывод
123gf-846g	1
2045532f0rrg512fd1f6	4
0	0
jkhhguio	0
-8654fghvbj785kjyu	1
1g2g3u	3
f32767fd7	2
d32768ds4	1

Если вводится строка, в которой используются только цифры и знак пробела:

Ввод	Вывод
15448 -954 8451 45 -4561 8451 0 52	5
123 -564 895300 654 654 -5 54 1	5
1 -5 -5 -5 -5 -5 -5	1
32768 0 25 84 -45230 -32 32767 - 1 9855 41 864105 98 8 8 -47	8
0	В файле не найдено ни одного числа, подходящего под условие.
32768 -15	В файле не найдено ни одного числа, подходящего под условие.
7 7 7 7	4

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

Допущен к выполнению работы. Подпись преподавателя _____

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

```

lina_tucha@LAPTOP-44CRFC1U:~$ cd dir
lina_tucha@LAPTOP-44CRFC1U:~/dir$ touch 2.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano 2.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ touch test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
2.c: In function 'main':
2.c:16:29: warning: format '%s' expects argument of type 'char *', but argument 3 has type 'int' [-Wformat=]
      fscanf(fi,"%s", s[i]);
                        ^~~~~
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano 2.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
1
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
4
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
0
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
0
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
1
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
3
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
2
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
1
lina_tucha@LAPTOP-44CRFC1U:~/dir$ touch 3.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano 3.c
lina_tucha@LAPTOP-44CRFC1U:~/dir$ touch test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 3.c -o 123
3.c: In function 'main':
3.c:16:42: error: expected expression before '>' token
      if (s[i]<32768 && s[i]>=0) { continue;}
                                   ^
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 3.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
5
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 3.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
5
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 3.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
1
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 3.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
8
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
В файле не найдено ни одного числа, подходящего под условие.
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
В файле не найдено ни одного числа, подходящего под условие.
lina_tucha@LAPTOP-44CRFC1U:~/dir$ nano test.txt
lina_tucha@LAPTOP-44CRFC1U:~/dir$ gcc 2.c -o 123
lina_tucha@LAPTOP-44CRFC1U:~/dir$ ./123
4

```

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
<u>1</u>	<u>дом</u>	<u>3.12.19</u>	<u>00:17</u>	lina_tucha@LAPTO P-44CRFC1U:~/dir\$ gcc 3.c -o 123 3.c: In function 'main': 3.c:16:42: error: expected expression before '>' token if (s[i]<32768 && s[i]>=0) { continue;} ^	if (s[i]>=32768 && s[i]<=0) { continue;}	<u>Опечатка</u>
<u>2</u>	<u>дом</u>	<u>3.12.19</u>	<u>0:02</u>	lina_tucha@LAPTO P-44CRFC1U:~/dir\$ gcc 2.c -o 123 2.c: In function 'main': 2.c:16:29: warning: format '%s' expects argument of type 'char *', but argument 3 has type 'int' [-Wformat=] fscanf(fi,"%s", s[i]); ~~~~~ ~^	fscanf(fi,"%s", &s[i]);	

10. Замечание автора по существу работы: 16-битные процессоры: диапазон возможных значений: от -32 768 до 32 767

11. Выводы : Я составила программу на языке Си, выполняющую анализ и обработку видимого текста в соответствии с вариантом задания

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента Хренникова А. С.