1	
по курсу: <u>1 фундаментальная информатика</u>	
студент группы : <u>М8О-105Б-21 Титеев Рамиль Маратович</u> , № по списку:	23
Адреса www, e-mail, jabber, skype <u>derol.gym@gmail.com</u>	
Работа выполнена: "26 ноября 2021г"	
Преподаватель: каф. 806 В. К. Титов	
Входной контроль знаний с оценкой	
Отчёт сдан "" 20г., итоговая оценка	
Подпись преподавателя	
1 Тема : Системы программирования на языке Си. Отладка простейших программ.	
2 Цель работы : изучение конкретной системы программированию на Си и получение навыко текстов и отладки программ. Составление и отладка простейшей программы на языки С итерат характера с целочисленными рекуррентными соотношениями, задающие некоторое регулярное точки в целочисленной системе координат(i,j) с дискретным временем К и динамическим параг движения l.	гивного е движение
3 Задание (вариант № 23): Подсчитать количество положительных десятичных чисел в строке, допустимых 16-битными п	процессорами.
4 Оборудование (лабораторное): ЭВМ, процессор, имя узла сети с ОПГБ НМД ГБ. Терминал адрес Принтер Другие устройства	
Оборудование ПЭВМ студента, если использовалось: Процессор <u>Ryzen4600 @ 6x 3.0GHz</u> , ОП <u>16384</u> МБ, НМД <u>Г</u> ГБ. Монитор: <u>встроенду и маказата в между и </u>	нный
5 Программное обеспечение (лабораторное): Операционная система семейства UNIX, наименование:	
Интерпретатор команд: версия версия версия версия	
Редактор текстов: версия Утилиты операционной системы:	
Прикладные системы и программы:	
Программное обеспечение ЭВМ студента, если использовалось: Операционная система семейства <u>UNIX</u> , наименование <u>Ubuntu</u> версия <u>20.0</u> . Интерпретатор команд: <u>bash</u> версия <u></u>	
Система программирования: С версия версия	
Редактор текстов: <u>Emacs</u> версия Утилиты операционной системы:	
Прикладные системы и программы: Местонахождения и имена файлов программ и данных:/usr/bin , а также /bin	

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

Выполнение программы происходит с посимвольным считыванием каждый буквы в исходном тексте. При этом в условие цикла while находится условие, которое проверяет на окончание файла. Внутри цикла происходит:

Считываем символ. Если встретился минус, то заменяем флаг, отвечающий за знак числа, на отрицательный. Если полученный символ является цифрой то добавляем его в конец нашего результата. Если встретился разделитель, то проверяем наше число, является ли оно больше чем 65536 (2¹⁶) или меньше нуля. Если все прошло, то проверяем флаги, если по мере считывания не встречался минус и любой другой символ кроме цифры, то выводим полученный результат, а затем обнуляем его. Если нам встретился любой другой символ, то меняем значение флага на 1.

После того, как мы дошли до конца файла, выводим итоговое количество чисел и прекращаем работу.

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

```
#include <stdio.h>
#define check 65536
int main(){
char s:
int n = 0, c = 0; // n - количество чисел, c - само число, которое будет составляться из
входных цифр
int f = 1, FLAG = 0; // Инициализируем два флага, f будет отвечать за знак числа, FLAG
будет принимать значение в зависимости от того, встретилась ли нам не цифра.
do{
s = getchar(); // Считываем символ
int a = s - '0'; // Изменяем тип на int
if(s == '-'){} // Если встретился минус, то меняем значение f на отрицательное значение
}
// Проверка, является ли символ цифрой
else if (a \ge 0 \&\& a \le 9)
c = c*10 + a;
// Если встретился разделитель
else if(s == ' ' || s == ',' || s == '\n'){
// Проверяем допускается ли это число 16-битными процессами и является ли оно
положительным
if (c*f < check && c*f > 0 && FLAG == 0){
n++;
printf("%d-е число: %d\n", n, c);
c = 0; f = 1; FLAG = 0; // обнуляем c, переменные f и FLAG принимают значение по
умолчанию
}
// Если же встретился любой другой символ, то обнуляем с и FLAG принимает значение 1
else{
FLAG = 1;
```

```
c = 0;
}
}while (s != EOF);
// Вывод
printf("Всего %d чисел\п", n);
return 0;
}
```

Допущен к выполнению работы. Подпись преподавателя

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

```
(base) ramil@ramil:~/labs/lab 11$ cat header.txt
            Лабараторная работа №11
* Обработка последовательности литер входного текстового файла. *
       Простейшие приёмы лексического анализа.
         Диаграммы состояний и переходов.
          Выполнил студент гр. М8О-105-Б
            Титеев Рамиль Маратович
**********************
(base) ramil@ramil:~/labs/lab 11$ cat > lab11_v23.c
#include <stdio.h>
#define check 65536
int main(){
  int n = 0, c = 0; // n - количество чисел, c - само число, которое будет составляться из входных цифр
  int f = 1, FLAG = 0; // Инициализируем два флага, f будет отвечать за знак числа, FLAG будет принимать значение в
зависимости от того, встретилась ли нам не цифра.
    s = getchar(); // Считываем символ
    int a = s - '0'; // Изменяем тип на int
    if(s == '-') \{ // Если встретился минус, то меняем значение <math>f на отрицательное значение
    // Проверка, является ли символ цифрой
    else if (a \ge 0 \&\& a \le 9){
      c = c*10 + a;
    // Если встретился разделитель
    else if(s == ' ' || s == ',' || s == ' n'){
      // Проверяем допускается ли это число 16-битными процессами и является ли оно положительным
      if (c*f < check && c*f > 0 && FLAG == 0){
        n++:
         printf("%d-е число: %d\n", n, c);
      c = 0; f = 1; FLAG = 0; // обнуляем с, переменные f и FLAG принимают значение по умолчанию
    // Если же встретился любой другой символ, то обнуляем с и FLAG принимает значение 1
      FLAG = 1;
      c = 0;
  }while (s != EOF);
  // Вывод
  printf("Всего %d чисел\n", n);
```

```
return 0;
(base) ramil@ramil:~/labs/lab 11$ cat > test
34123 dsajdjdf 132 395, 3748, 03012 439119 2312jsa 3434das
(base) ramil@ramil:~/labs/lab 11$ g++ lab11_v23.c
(base) ramil@ramil:~/labs/lab 11$ ./a.out < test
1-е число: 34123
2-е число: 132
3-е число: 395
4-е число: 3748
5-е число: 3012
Всего 5 чисел
(base) ramil@ramil:~/labs/lab 11$ cat > test
2938 re7ei38i e939.34
938011
eio9o00 00 39031,88,1209,eie39
(base) ramil@ramil:~/labs/lab 11$ ./a.out < test
1-е число: 2938
2-е число: 39031
3-е число: 88
4-е число: 1209
Всего 4 чисел
(base) ramil@ramil:~/labs/lab 11$ cat > test
ir900,343,65537 0991, 33dkk39, 939, 10031dkkds
f991 837100jc 810fkk89 d77uu
991100102030102301203, 23oo3e991,13991
.432.123.123 9391/13 8381,341
(base) ramil@ramil:~/labs/lab 11$ ./a.out < test
1-е число: 343
2-е число: 991
3-е число: 939
4-е число: 13991
5-е число: 8381
6-е число: 341
Всего 6 чисел
```

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.						
No	Лаб.	Дата	Время	Событие	Действие по исправлению	Примечание	
	или		-		-	-	
	дом.						
	- / \						
10	Замечание автора по существу работы <u>Я научился писать программы на Си с чтением данных из файла, а также работать с символьным типом переменных.</u>						
	а также	работать с с	имвольным ти	пом переменных.			
							
11	Выволь	л Яосво	ил основные і	навыки программировани	я на Си научился писать программы	Отлаживать их	
11	ВыводыЯ освоил основные навыки программирования на Си, научился писать программы, отлаживать их, выполнять ручной просчет.						
	Недоче	ты, допушені	ные при выпол	тнении задания, могут бы	гь устранены следующим образом <u>І</u>	Больше практики и	
			ного материал				
					Подпись студента		
	подпись студени						