

# Отчет по лабораторной работе № 11 по курсу «Фундаментальная информатика»

Студент группы Алапанова Эльза Халилевна, № по списку 3

Контакты e-mail : alapanowa02@yandex.ru

Работа выполнена: «21» января 2021г.

Преподаватель: каф. 806 Найденов Иван Евгеньевич

Отчет сдан «        » \_\_\_\_\_ 20 \_\_\_\_ 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.
3. **Задание (вариант №17):** Выделить все числа, записанные в троичной системе счисления и кратные 3.
4. **Оборудование** (студенческое)

Процессор Intel® Core™ i5-10210 @ 1.60 GHz с ОП 8192 Мб, НМД 512 Уб. Монитор 1920 x 1080

**Программное обеспечение (студенческое):**

Операционная система семейства Windows, наименование Windows 10 домашняя версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_.

Система программирования \_\_\_\_\_ версия \_\_\_\_\_

Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы Sublime Text

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

На стандартный поток ввода подаётся произвольная последовательность строк из ascii-знаков. В программе не должно быть ограничений на количество и длину строк исходного текста, слова могут состоять из любых непробельных ascii-знаков. Все числа, присутствующие во входном потоке, гарантированно являются целыми и не превосходят по модулю  $5 * 10^{18}$ .

Если число кратно 10 и состоит из чисел 0, 1, 2, то выводим его.

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

План работы:

1) Если в начале стоит '-', то  $y = -1$ , тем самым потом можем вывести этот минус.

2) Если в начале стоит '+', то  $plus0 = 1$ , тем самым потом можем вывести этот +/

3) При ведущих нулях  $x$  все равно удовлетворяет условию, однако, cnt увеличивается с каждым разом -> число делим на 10, увеличивая с каждым разом  $i$ . Сравниваем cnt и  $i$ , если  $i$  больше, то выставляем столько ведущих нулей, сколько разница между  $i$  и cnt.

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

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

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

```
#include <math.h>
```

```
typedef enum {  
    S0, S1  
} state;
```

```
int is_digit(char c)  
{  
    if ((c == '0') || (c == '1') || (c == '2')) {  
        return 1;  
    }  
    return 0;  
}
```

```
int main(void)  
{  
    int y = 1, o = 0, cnt = 0, z = 0, plus = 0, i, nul = 0, copy, plus0 = 0;  
    char c;  
    long int x = 0;  
    state S = S0;
```

```
    while ((c = getchar()) != EOF) {  
        if ((c == '+') && (x == 0) && (cnt == 0)) {  
            plus = -2;  
            plus0 = 1;  
            continue;  
        }  
        if (c == '-') {  
            if ((y != 1) || (x != 0)) {  
                S = S1;  
            } else {  
                y = -1;  
                z = -1;  
            }  
            continue;  
        }  
        if ((c == '\n') && (o != 0)) {  
            x = 0; cnt = 0; y = 1; o = 0;  
            printf("\n");  
            continue;  
        }  
        switch (S) {
```

```
            case S0:  
                if (is_digit(c) == 1) {  
                    x = 10 * x + c - '0';  
                    ++cnt;  
                    z = 2;  
                    plus += 1;  
                    S = S0;  
                } else {  
                    if (c == ' ' && cnt != 0) {  
                        if (x % 10 == 0) {  
                            o = 1;  
                            copy = x;  
                            for (i = 0; copy != 0; i++) {  
                                copy /= 10;  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }
```

```

    if (i != cnt && x != 0) {
        i = cnt - i;
        if (y == -1) {
            printf("%c", '-');
        } else if (plus0 == 1) {
            printf("%c", '+');
        }
        while (i != 0) {
            printf("%d", nul);
            i--;
        }
        printf("%ld ", x);
    } else if (z == 2 && x == 0) {
        if (y == -1) {
            printf("%c", '-');
        } else if (plus0 == 1) {
            printf("%c", '+');
        }
        while (cnt != 0) {
            printf("%d", nul);
            cnt--;
        }
        printf("%c", ' ');
    } else if (y == -1 && x != 0) {
        printf("-%ld ", x);
    } else if (plus0 == 1 && x != 0) {
        printf("+%ld ", x);
    } else if (z == 2 && x == 0 && y == -1) {
        printf("-%ld ", x);
    } else if (plus == -1 && x == 0) {
        printf("+%ld ", x);
    } else {
        printf("%ld ", x);
    }
}
cnt = 0; x = 0; y = 1; o = 0; z = 0; plus = 0; nul = 0; plus0 = 0;
S = S0;
} else {
    if (c == '\n' && cnt != 0) {
        if (x % 10 == 0) {
            copy = x;
            for (i = 0; copy != 0; i++) {
                copy /= 10;
            }
            if (i != cnt && x != 0) {
                i = cnt - i;
                if (y == -1) {
                    printf("%c", '-');
                } else if (plus0 == 1) {
                    printf("%c", '+');
                }
                while (i != 0) {
                    printf("%d", nul);
                    i--;
                }
                printf("%ld", x);
            } else if (z == 2 && x == 0) {
                if (y == -1) {
                    printf("%c", '-');

```

```

    } else if (plus0 == 1) {
        printf("%c", '+');
    }
    while (cnt != 0) {
        printf("%d", nul);
        cnt--;
    }
    } else if (y == -1 && x != 0) {
        printf("-%ld", x);
    } else if (plus0 == 1 && x != 0) {
        printf("+%ld", x);
    } else if (z == 2 && x == 0 && y == -1) {
        printf("-%ld", x);
    } else if (plus == -1 && x == 0) {
        printf("+%ld", x);
    } else {
        printf("%ld", x);
    }
}
cnt = 0; x = 0; y = 1; o = 0; z = 0; plus = 0; nul = 0; plus0 = 0;
printf("\n");
S = S0;
} else {
    S = S1;
}
}
}
break;

```

case S1:

```

    if ((c == ' ') || (c == '\n')) {
        cnt = 0; x = 0; y = 1; o = 0; z = 0; plus = 0; nul = 0; plus0 = 0;
        if (c == '\n') {
            printf("\n");
        }
        S = S0;
    } else {
        S = S1;
    }
    break;
}
}
printf("\n");
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 дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1						
2						

10. **Замечания автора** по существу работы : замечаний нет.

11. **Выводы** : очень трудно дается программирование на Си..

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_

Подпись студента \_\_\_\_\_