**Лабораторная работа 4**

**Вариант 5**

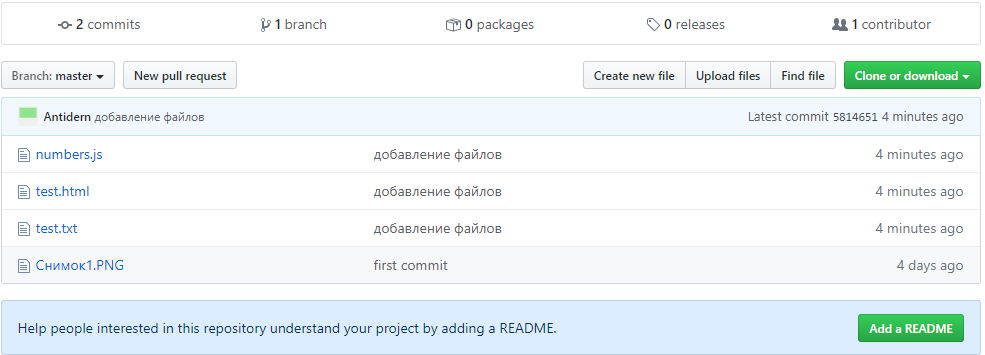
С помощью Visual Studio Code cоздадим необходимые файлы проекта, для того чтобы было что загружать в созданный репозиторий на GitHub.

Теперь в проекте есть 3 файла – это test.txt, содержащий текст, который необходимо видоизменить, html-страница, для отображения полученного результатов и numbers.js для обработки файла.

Чтобы загрузить их в репозиторий, необходимо включить систему контроля версий в директорию проекта. Для этого откроем соответствующую директорию в Git Bash, пропишем команду "git init", а затем добавим все три файла, расположенные в директории, при помощи команды "git add .". Данное действие можно прокомментировать, добавив коммит с необходимым сообщением при помощи команды "git commit -m".

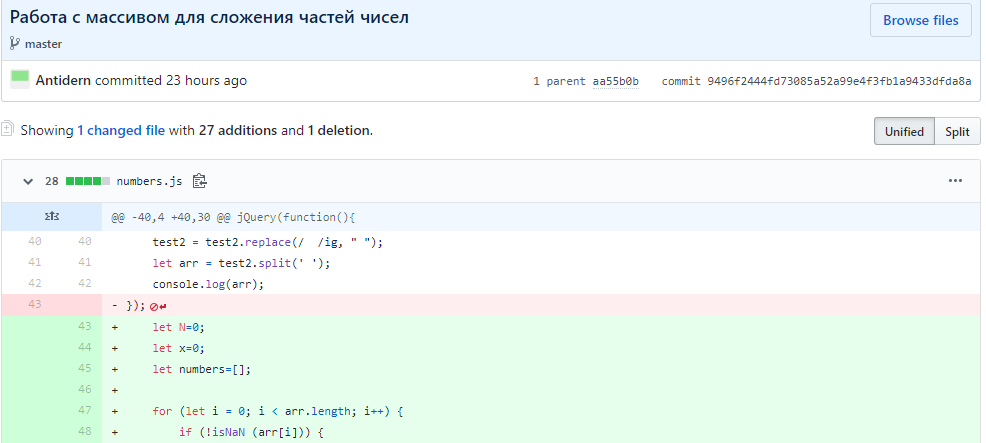
Теперь, для того чтобы загрузить добавленные или измененные файлы в репозиторий GitHub, необходимо установить авторизованное подключение, выполнив команду "git remote add origin" со ссылкой на репозиторий. После авторизации в открывшемся окне, соединение будет установлено. Теперь можно загрузить файлы в репозиторий при помощи команды "git push -u origin master". В консоль будет выведена подробная информация о статусе загрузки файлов.

После загрузки новые (измененные) файлы отобразятся в репозитории:

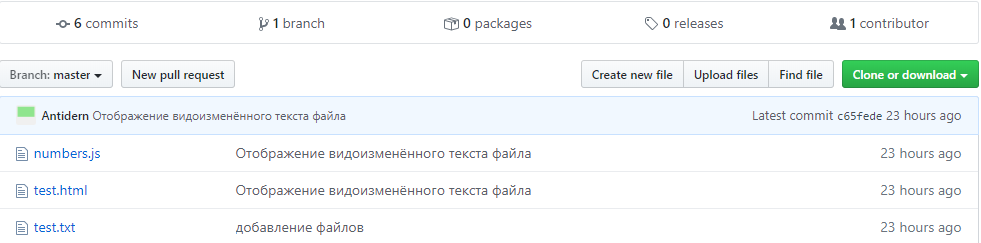


Следующим коммитом мы изменим test.html и numbers.js для того, чтобы отобразить текст из файла на веб-странице. После изменения файлов следует выбрать Source Control на боковой панели Visual Studio Code, сохранить файлы, описать изменения и добавить их в репозиторий с помощью команд Сommit и Push.

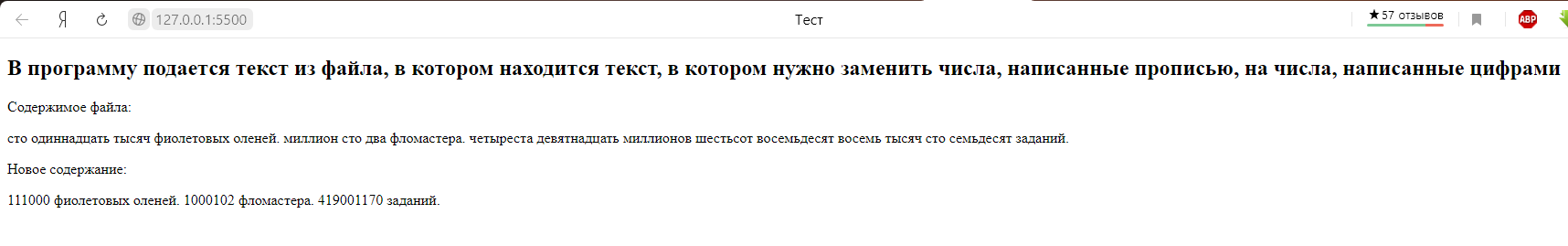
Осталось реализовать непосредственно алгоритм замены прописных чисел на цифровые. С помощью регулярных выражений заменим все прописные части чисел на числа в виде цифр. Далее, преобразуем полученную строку в массив и соединим соседние части чисел в отдельное число. Код алгоритма представлен в листинге в конце лабораторной работы. После обновления файла в репозитории в GitHub снова можно наблюдать изменения. Можно сравнить текущую версию файла с предыдущей. Будет наглядно отображено, какие строки были удалены и какие добавлены:



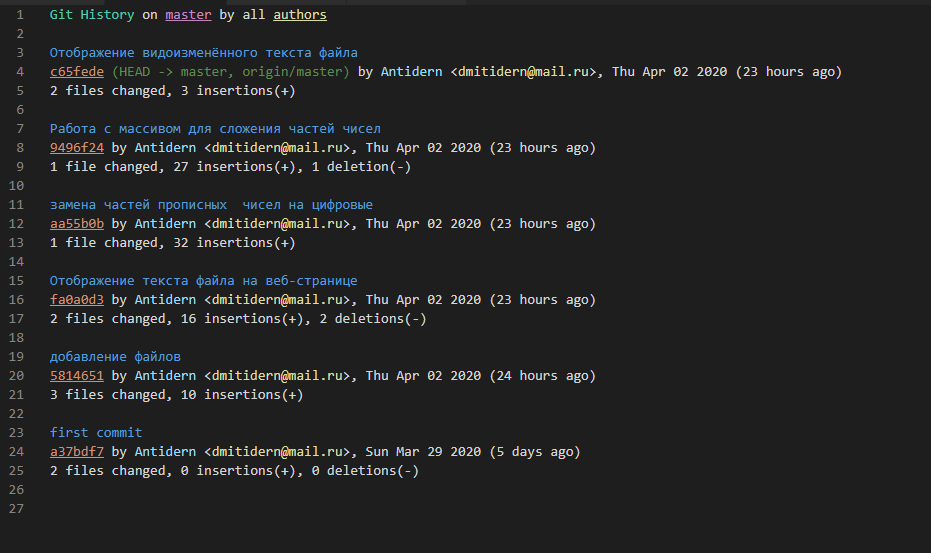
Теперь необходимо преобразовать полученный массив в строку и вывести её на веб-страницу, не забыв добавить изменения в репозиторий GitHub.



Результат работы программы:



Все коммиты отправленные на GitHub:



Листинг программы:

var test;

$.ajax({

    url: "test.txt",

    dataType: "text",

    async: true,

    success: function(msg){

        test = msg;

        // alert('Содержимое файла: '+test);

        document.getElementById("content").innerHTML = (test);

    }

});

jQuery(function(){

    let test1 = new String;

    test1 = document.getElementById("content").innerHTML;

    test2 = test1.replace(/миллион(ов|ом|ах|ами|а|е|у|ы|)/ig , " 1000000");

    test2 = test2.replace(/тысяч(ов|ей |ах|ех|ях|ами|ями|а|о|е|и|у|ы|на|)/ig, " 1000");

    test2 = test2.replace(/девяност(о|а)/ig, " 90");

    test2 = test2.replace(/ст(ам |ах |ами |а |о|и|у|ы|на)/ig, " 100");

    test2 = test2.replace(/сот/ig, " 100");

    test2 = test2.replace(/девятнадцат(ь|ью|и)/ig, " 19");

    test2 = test2.replace(/восемнадцат(ь|ью|и)/ig, " 18");

    test2 = test2.replace(/семнадцат(ь|ью|и)/ig, " 17");

    test2 = test2.replace(/шестнадцат(ь|ью|и)/ig, " 16");

    test2 = test2.replace(/пятнадцат(ь|ью|и)/ig, " 15");

    test2 = test2.replace(/четырнадцат(ь|ью|и)/ig, " 14");

    test2 = test2.replace(/тринадцат(ь|ью|и)/ig, " 13");

    test2 = test2.replace(/двенадцат(ь|ью|и)/ig, " 12");

    test2 = test2.replace(/одиннадцат(ь|ью|и)/ig, " 11");

    test2 = test2.replace(/дцат(ь |ью |и |у )/ig, " 10");

    test2 = test2.replace(/десят(ь |ом |ой |ью |и |)/ig, " 10");

    test2 = test2.replace(/девят(ь|ью|и|)/ig, " 9 ");

    test2 = test2.replace(/вос(емь|емью|ьми )/ig, " 8 ");

    test2 = test2.replace(/сем(ь|и|ью)/ig, " 7 ");

    test2 = test2.replace(/шест(ь|и|на|ью|)/ig, " 6 ");

    test2 = test2.replace(/пят(ь|и|на|ью)/ig, " 5 ");

    test2 = test2.replace(/четыр(ём|е|на|ёх|ех)/ig, " 4 ");

    test2 = test2.replace(/тр(ём|и|ёх|ех)/ig, " 3 ");

    test2 = test2.replace(/дв(а|умя|ух|ум|е)/ig, " 2");

    test2 = test2.replace(/од(ин |ного |ном |ним |на )/ig, " 1 ");

    test2 = test2.replace(/  /ig, " ");

    let arr = test2.split(' ');

    console.log(arr);

    let N=0;

    let x=0;

    let numbers=[];

    for (let i = 0; i < arr.length; i++) {

        if (!isNaN (arr[i])) {

            let n1 = parseInt(arr[i]);

            if (!isNaN(arr[i+1]) ) {

                let n2 = parseInt(arr[i+1]);

                if((n1 - n2) >= 991) {

                    N += n1;

                } else if (n1 > n2) {

                    arr[i+1] = n2 + n1;

                } else if(n1 < n2) {

                    arr[i+1] = n1 \* n2;

                }

            } else {

                numbers[x] = n1 + N;

                x++;

                N=0;

            }

        } else { /\* для записи в новый массив нечисловых значений\*/

            numbers[x] = arr[i];

            x++;

        }

    }

    let str1 = numbers.join(' ');

    document.getElementById("fff").innerHTML = (str1);

});