МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

|  |
| --- |
|  |

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационной безопасности**

**Интеллектуальные компьютерные системы**

**ОТЧЕТ**по лабораторной работе №4

на тему «**Метод Ван Хао**»

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Выполнил студент гр. Б8116-09.03.01 | | |  | Гусев М.Д. | |  | | | Проверил | | |  | Ю.С. Москаленко | | дата | | |  | | | зачтено/не зачтено | | |

г. Владивосток

2019

**Постановка задачи и входные данные**

В качестве входных данных выступают две строки: «Дано» и «требуется получить»

Задача состоит в том, чтобы определить есть ли решение (когда левая от импликации часть совпадает с правой), либо его нет.

**Реализация**

Изначально пользователю предоставляются поля, в которых можно ввести две строки, которые должны быть слева и справа от импликации.

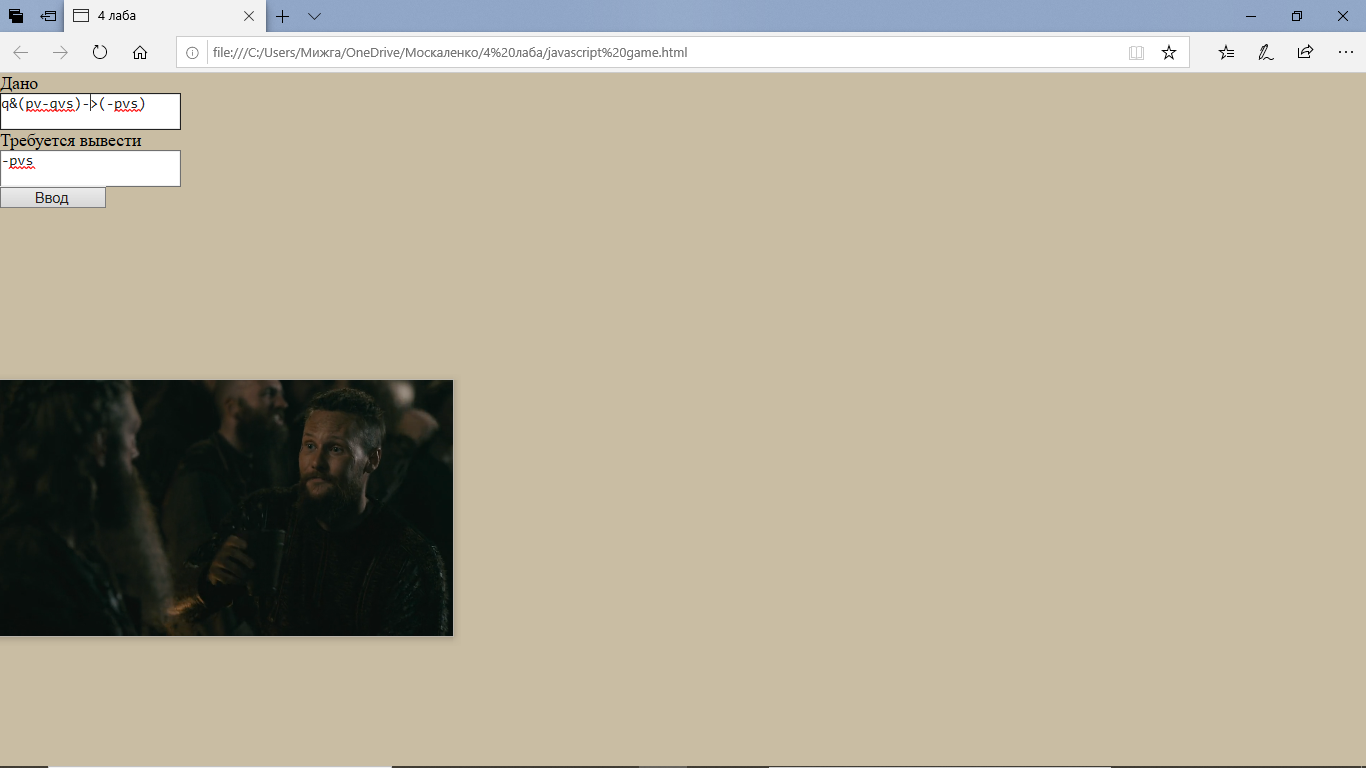
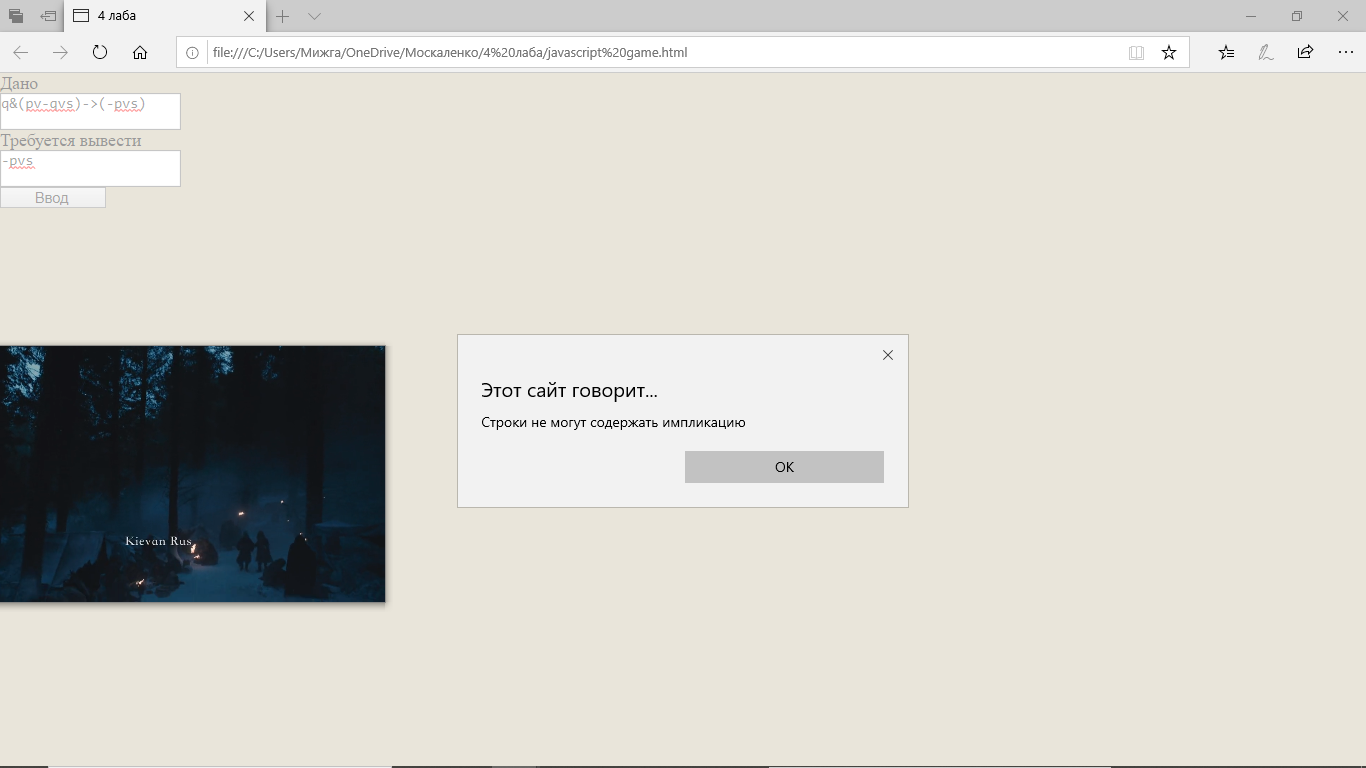
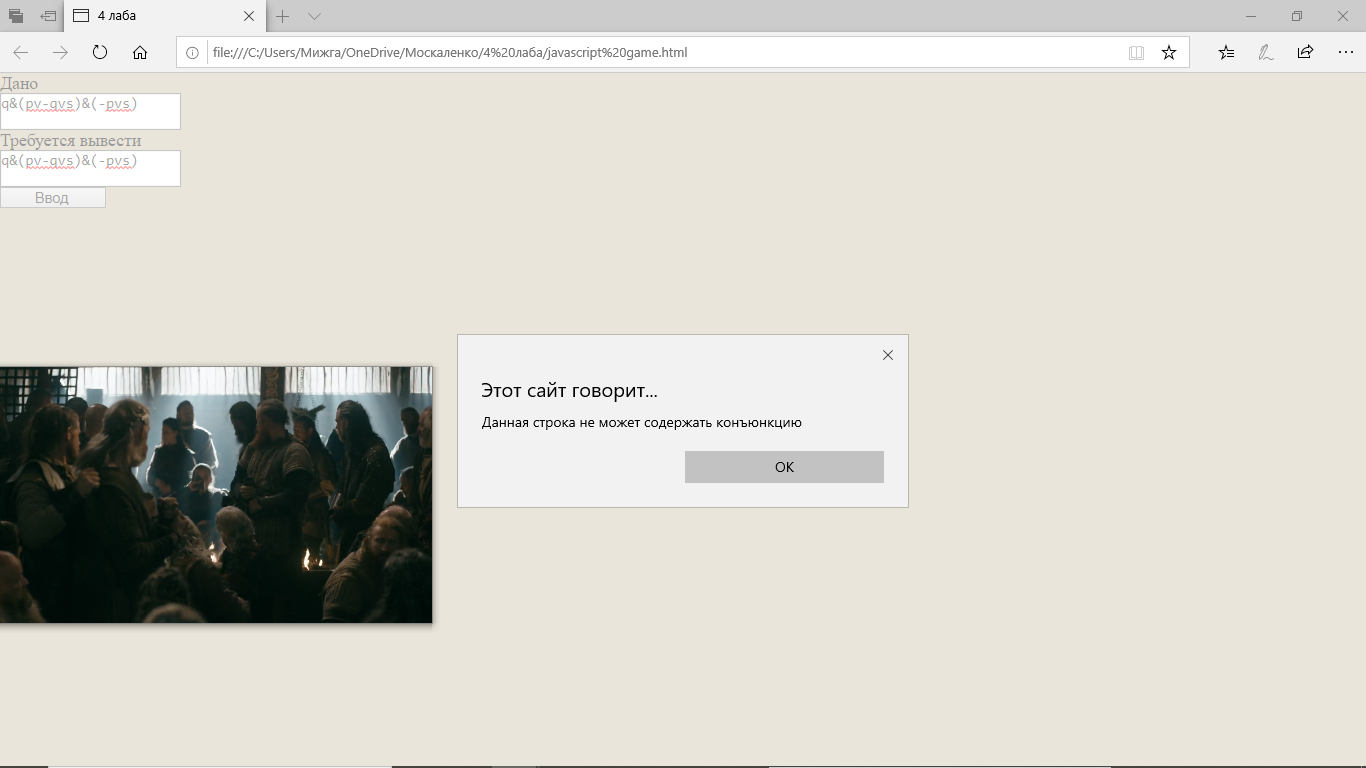


рис. 1 – поля для ввода данных.

Данная реализация метода Ван Хао имеет в себе некоторые ограничения:  
1) Импликация может быть всего одна, поэтому если в какой-то из строк для ввода будет введена импликация, программа выдаст предупреждение, после чего предложит снова ввести данные.  
2) В правой от импликации части строки не может быть конъюнкций, если в ней будет введена конъюнкция, программа выдаст предупреждение, после чего предложит снова ввести данные.

  
Рис.2 – Предупреждение о том, что введена импликация

  
Рис.3 – Предупреждение о том, что в правой части введена конъюнкция

Если же условия соблюдены, то программа просчитывает и выводит бинарное дерево, в котором просчитываются все конъюнкции и дизъюнкции.

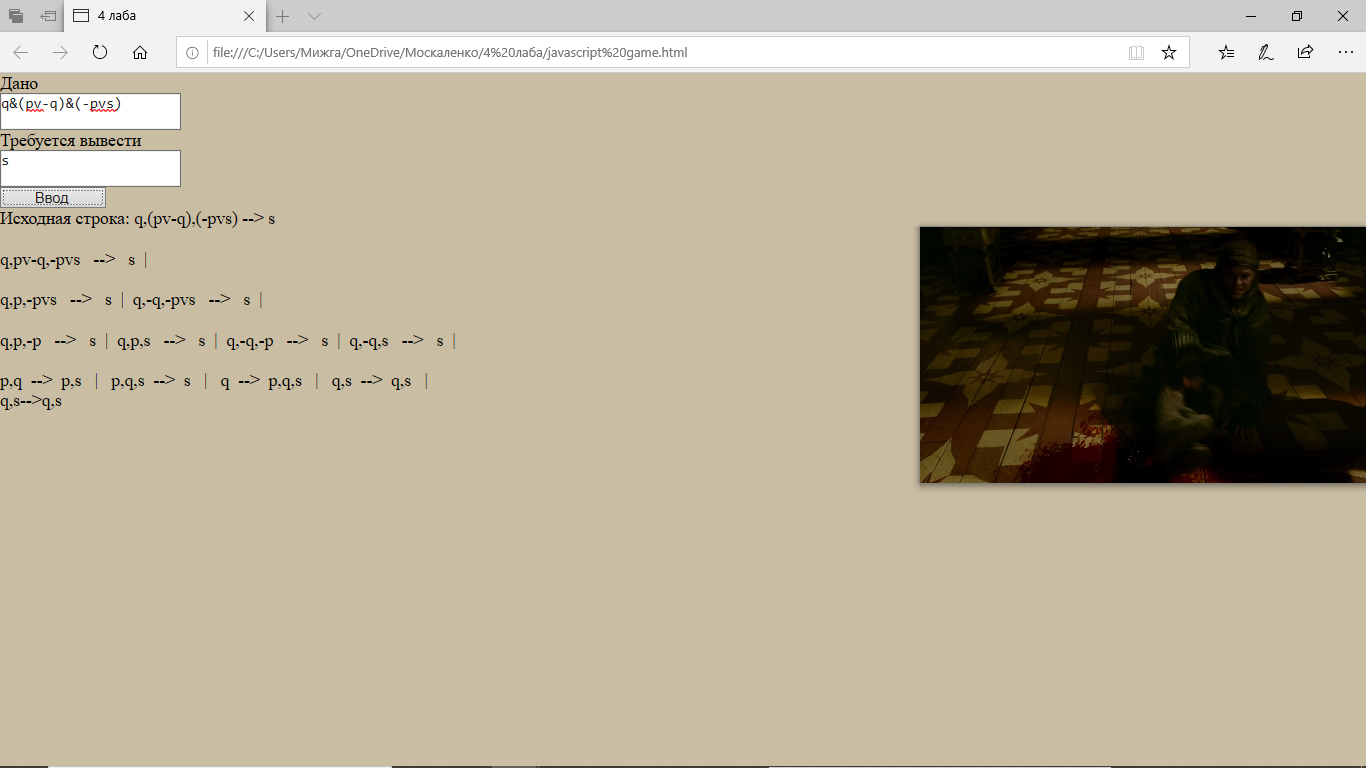


рис. 4 – бинарное дерево с просчетом конъюнкций и дизъюнкций

Затем из дерева выбираются ветки, в которых отсутствуют конъюнкции и дизъюнкции и начинается просчет отрицательных элементов. Отрицательные элементы переносятся в другую сторону относительно импликации, при этом пропадает знак отрицания.

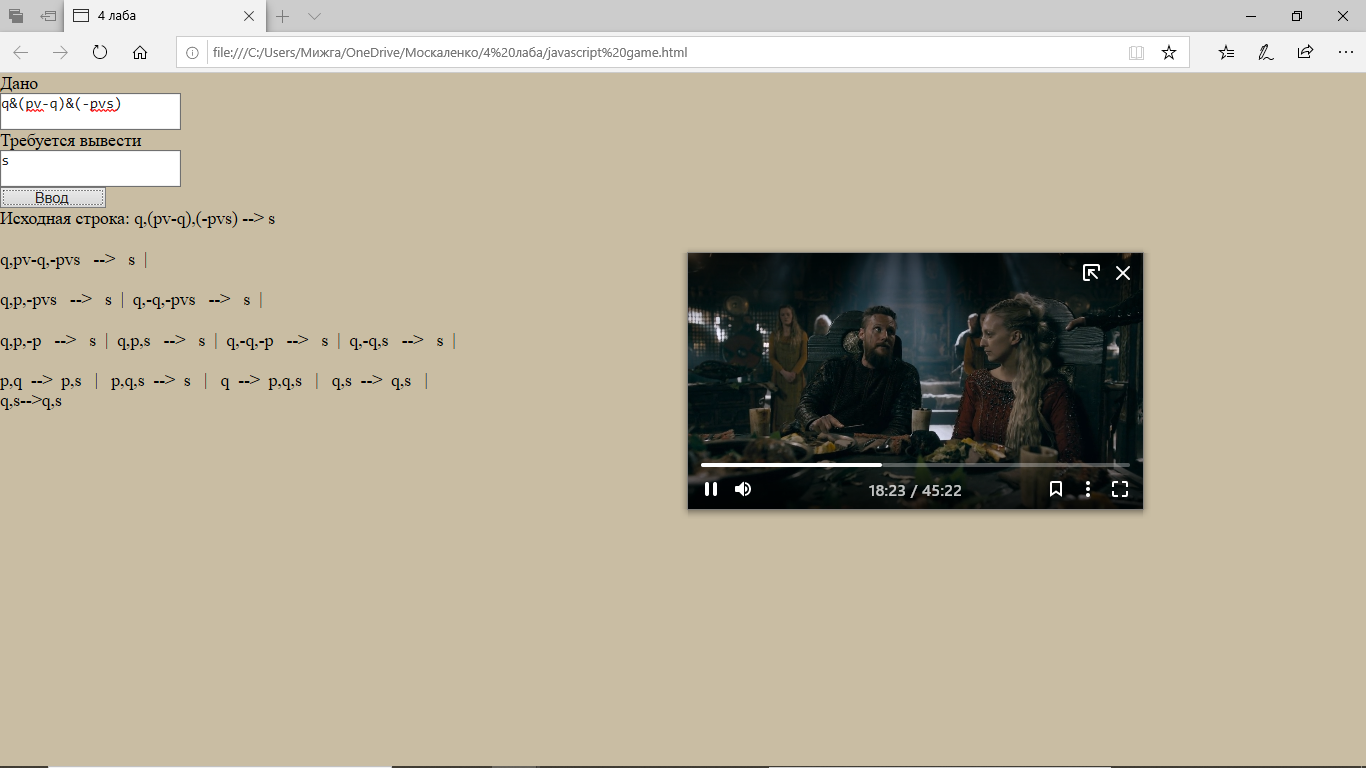


рис.5 – дерево и просчет отрицательных элементов

Если после просчета отрицательных элементов находятся ветки с одинаковой левой и правой частью, то программа выделяет и выводит такие ветки.

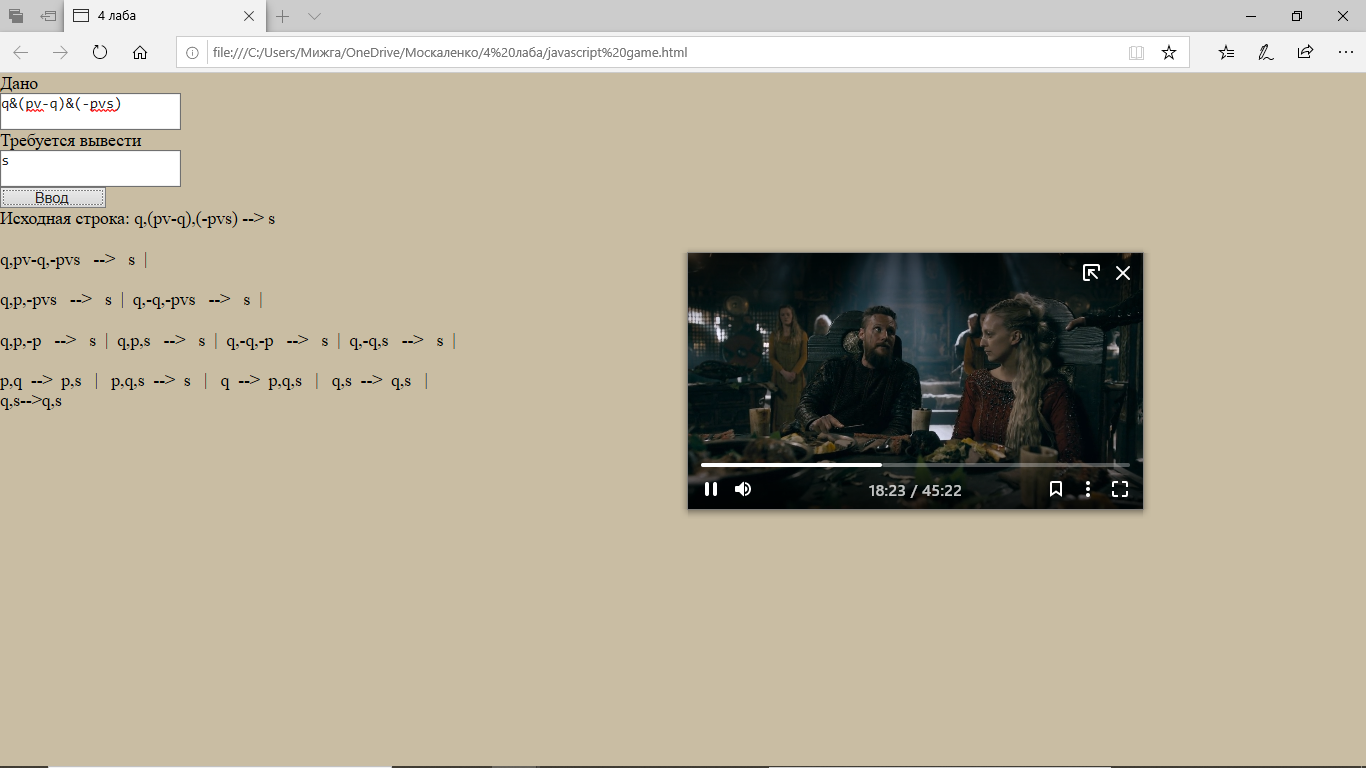


рис. 6 – дерево, просчет отрицательных элементов, вывод ответа

Код программы

var count=0; count1 = 0;

var s;var t=[];var el =[]; floor=[]; right =[]; left =[];answer = [];var ss =[];

var end;

window.onload = function(){

s;t=[];el =[]; floor=[]; right =[]; left =[];answer = [];

var end;

p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML ="Дано";

text1=document.createElement('textarea');

document.body.appendChild(text1);

br=document.createElement('br');

document.body.appendChild(br);

p1 = document.createElement('p');

document.body.appendChild(p1);

p1.innerHTML ="Требуется вывести";

text2=document.createElement('textarea');

document.body.appendChild(text2);

br=document.createElement('br');

document.body.appendChild(br);

for(var i=0;i<1;i++){

input = document.createElement('input');

document.body.appendChild(input);

input.type = 'button';

input.value=" Ввод ";

}

br1 = document.createElement('br');

document.body.appendChild(br1);

input.onclick=function(){

t[0] = text1.value;

ss[0] = text2.value;

base();

}

}

function base(){

for(var i=0;i<ss[0].length;i++){

if(ss[0][i] == ">"){

alert("Строки не могут содержать импликацию");

window.location.reload();

}

if(ss[0][i] == "&"){

alert("Данная строка не может содержать конъюнкцию");

window.location.reload();

}

}

for(var i=0;i<t[0].length;i++){

if(t[0][i] == ">"){

alert("Строки не могут содержать импликацию");

window.location.reload();

}

}

t[1] =[];

for(var i=0;i<t[0].length;i++){

if(t[0][i] != "&" )t[1] += t[0][i];

else if(t[0][i] == "&") t[1] += ",";

}

ss[1]=[];

for(var i=0;i<ss[0].length;i++){

if(ss[0][i] != "v" )ss[1] += ss[0][i];

else if(ss[0][i] == "v") ss[1] += ",";

}

p1 = document.createElement('p');

document.body.appendChild(p1);

p1.innerHTML ="Исходная строка: "+ t[1] +" --> "+ss[1];

br=document.createElement('br');

document.body.appendChild(br);

s = ss[1].split(",");

el[0]=[];

el[0] = t[1].split(",");

for(var i=0;i<el[0].length;i++){

var l="";

for(var j=0;j<el[0][i].length;j++){

if(el[0][i][j] == "("){

for(var k=1;k<el[0][i].length-1;k++){

l +=el[0][i][k];

}

el[0][i] = l;

}

}

}

end =1;

floor.push(end);

work();

}

function work(){

var start = el.length;

count=0;

flag=0;

//alert(start-end+" " + el.length)

for(var i=start-end;i<start;i++){

flag = 0;

for(var j=0;j<el[i].length;j++){

if(flag != 0)break;

for(var k=0;k<el[i][j].length;k++){

if(el[i][j][k] == "v"){

el.push([]);count++;

var l="";

for(var u=0;u<j;u++){

el[el.length-1][u] = el[i][u];

}

for(var u=0;u<k;u++){

l+=el[i][j][u];

}

el[el.length-1][j] = l;

for(var u=j+1;u<el[i].length;u++){

el[el.length-1][u] = el[i][u];

}

el.push([]);count++;

var l="";

for(var u=0;u<j;u++){

el[el.length-1][u] = el[i][u];

}

for(var u=k+1;u<el[i][j].length;u++){

l+=el[i][j][u];

}

el[el.length-1][j] = l;

for(var u=j+1;u<el[i].length;u++){

el[el.length-1][u] = el[i][u];

}

flag=1;

break;

}

}

}

}

end = count;

floor.push(end);

for(var i=el.length-count;i<el.length;i++){

for(var j=0;j<el[i].length;j++){

for(var k=0;k<el[i][j].length;k++){

if(el[i][j][k] == "v"){

work();

return 0;

}

}

}

}

for(var i=0;i<el.length;i++){

if(el[i].length == 1){

if(el[i][0] == s)answer.push(el[i] + "-->"+s[1]+ "&nbsp"+"&nbsp");

}

}

minus();

}

function minus(){

flagi = [];

for(var i = 0;i<el.length;i++){

flag =1;

for(var j=0;j<el[i].length;j++){

for(var k=0;k<el[i][j].length;k++){

if(el[i][j][k] == "v"){

flag = 0;

}

}

}

if(flag == 1){

flagi.push(i);

}

}

count=0;

left=[];

for(var i=0; i<flagi.length;i++){

left.push([]);

for(var j=0;j<el[flagi[i]].length;j++){

left[count][j] = el[flagi[i]][j];

}

count++;

}

for(var i=0;i<flagi.length;i++){

right.push([]);

for(var j=0;j<s.length;j++){

right[i].push(s[j]);

}

}

for(var i=0;i<left.length;i++){

str = left[i].length;

for(var j=0;j<str;j++){

for(var k=0;k<left[i][j].length;k++){

if(left[i][j][k] == "-"){

console.log("зашло"+left[i][j]);

right[i].push(left[i][j][k+1]);

left[i].splice(j,1);

j--;str--;break;

}

}

}

}

flag=0;

for(var i=0;i<right.length;i++){

str = right[i].length;

for(var j=0;j<str;j++){

for(var k=0;k<right[i][j].length;k++){

if(right[i][j][k] == "-"){

left[i].push(right[i][j][k+1]);

right[i].splice(j,1);

j--;str--;break;

console.log(right[i]);

}

}

}

}

for(var i=0;i<right.length;i++){

right[i].sort();

left[i].sort();

}

for(var i=0;i<right.length;i++){

count=0;

if(right[i].length == left[i].length){

for(var j=0;j<right[i].length;j++){

if(right[i][j] == left[i][j]){

count++;

console.log(count);

}

}

if(count == right[i].length){

answer.push(left[i] + "-->"+right[i]+ "&nbsp"+"&nbsp");

}

}

}

if(answer == "[]")answer = "Нет решения";

count=0;

for(var i=0;i<floor.length;i++){

p = document.createElement('p');

document.body.appendChild(p);

var l="";

for(var j=0;j<floor[i];j++){

l += el[count] + "&nbsp"+"&nbsp"+" --> "+"&nbsp"+"&nbsp"+s+"&nbsp"+"&nbsp"+"|"+"&nbsp"+"&nbsp";

count++;

}

p.innerHTML = l;

br=document.createElement('br');

document.body.appendChild(br);

}

var l="";

for(var i=0;i<left.length;i++){

l+=left[i]+"&nbsp"+" --> "+"&nbsp"+right[i]+ "&nbsp"+"&nbsp"+"&nbsp"+"|"+"&nbsp"+"&nbsp"+"&nbsp";

}

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = l;

var p = document.createElement('p');

document.body.appendChild(p);

p.innerHTML = answer;

}