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 №2

на тему «**Кластеризация с заданным L**»

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

г. Владивосток

2019

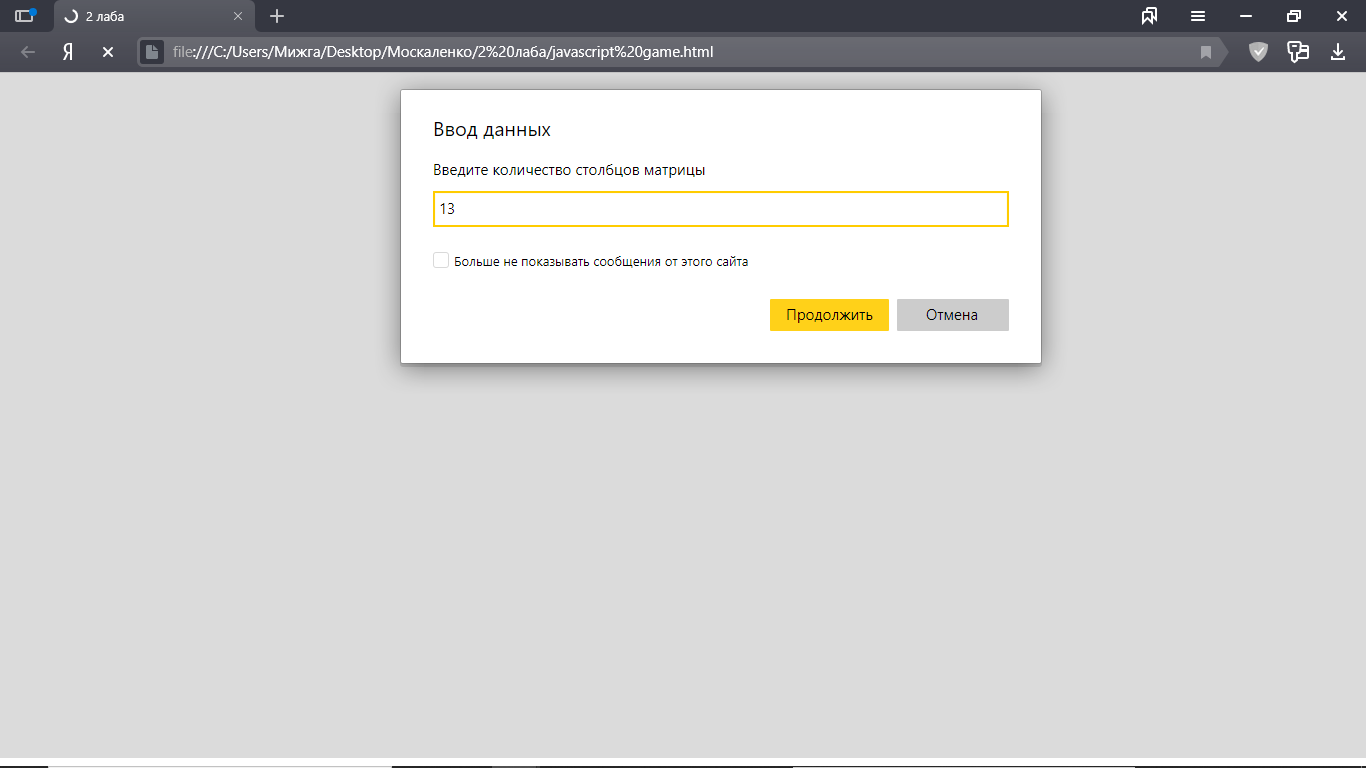
**Постановка задачи и входные данные**

В качестве входных данных выступает число кластеров L, а также матрица заданной размерности, заполненная единицами и нулями.

Задача состоит в том, чтобы построить таблицу несоответствий и на ее основе определить строки с минимальными расхождениями. Полученную цепочку элементов разделить на количество кластеров равное L.

**Реализация**

Изначально пользователю предоставляются поля, в которых можно задать все необходимые входные данные: количество строк матрицы, количество столбцов матрицы, количество связей L, а также предлагается заполнение самой матрицы нулями и единицами.   
Все это было реализовано с помощью метода parseInt(promt(,)), который выводит пользователю диалоговое окно с заданным текстом, а также поле для ввода данных типа integer.



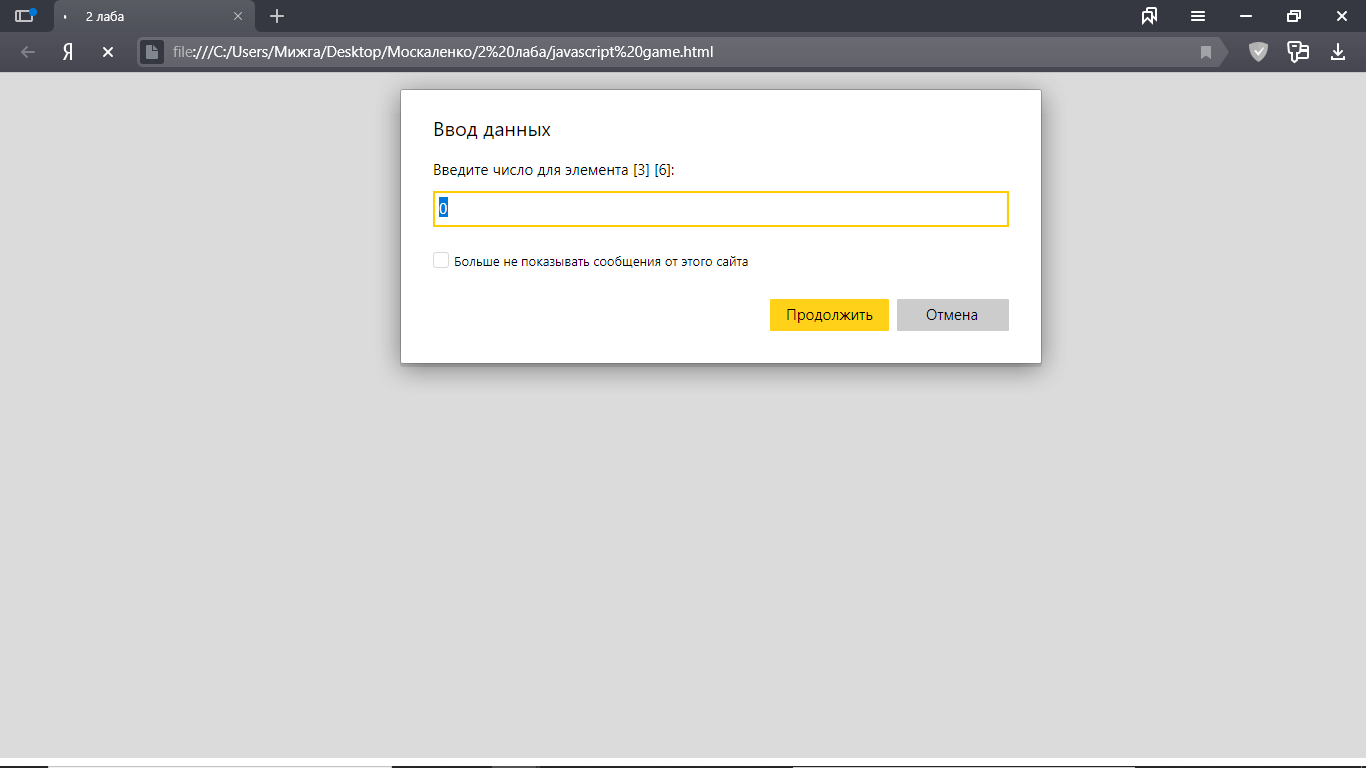


рис. 1,2 – диалоговое окно для ввода данных.

После ввода данных пользователю отображается заданная матрица. Для этого введенные данные были сохранены в двумерном массиве.   
Также создается еще один двумерный массив, каждый элемент которого будет являться блоком div, которому задается нужный визуальный стиль. Именно внутрь блоков div помещены данные из первого массива.  
Если кратко, в первом массиве хранятся введенные цифры, во втором – блоки div, которые являются визуальной оболочкой для цифр.

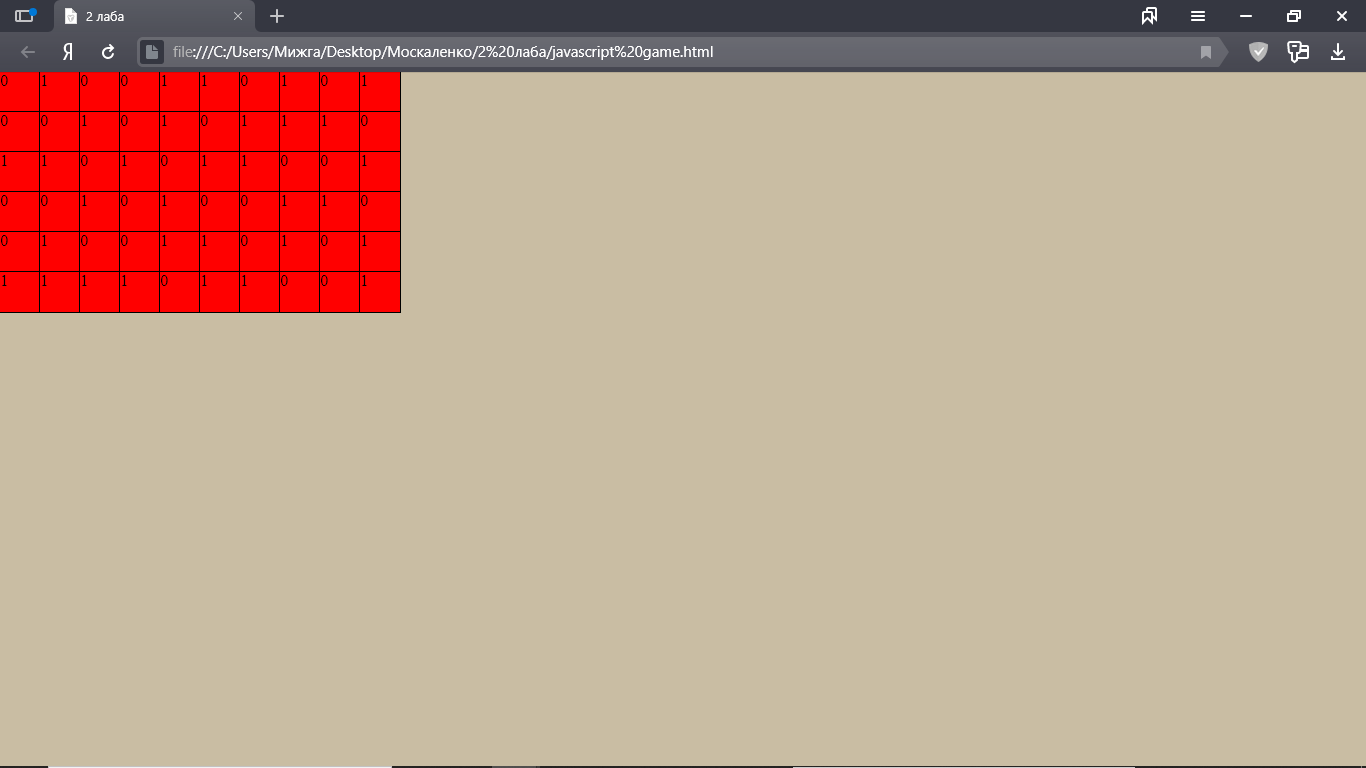


рис. 3 – матрица введенных элементов

Затем нужно выстроить матрицу несоответствий. Для этого используется тройной цикл, внутри которого стоит условие, проверяющее совпадает ли элемент с аналогичным элементом в другой строке. Также вместе с условием запущен счетчик, считающий количество несовпадений. Результат записывается в очередной двумерный массив.  
Для визуального отображения создается еще один двумерный массив, который будет хранить блоки div, с заданным визуальным стилем, как уже это было ранее.

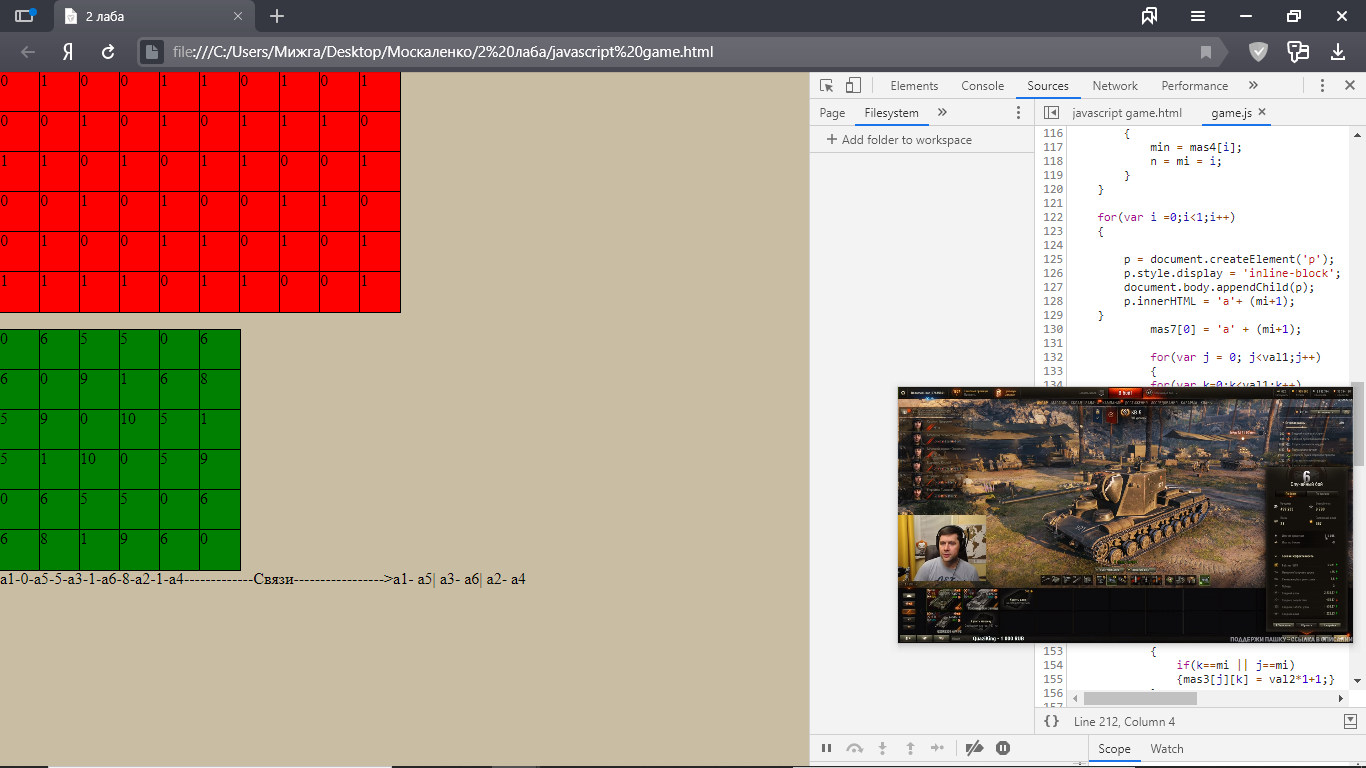


рис.4 – матрица введенных элементов и матрица несоответствий

Цепочка минимальных несоответствий всегда начинается с самого наименьшего несоответствия. Для его нахождения сначала находятся минимальные элементы матрицы во всех строках, а потом уже среди этих элементов ищется самый минимальный. В новый массив записывается индекс, элемента, а также строка, в которой он находится. По индексу элемента идет переход на строку с этим индексом, после чего уже в той строке ищется минимальный элемент не соответствующий элементу, связывающему нынешнюю и прошлую строку.  
Таким образом образовывается два массива: с номерами строк и цифрами, означающими минимальное количество несоответствий между данными строками. Данные массив выводится на экран, вставляя символ « - » после каждого элемента для лучшего визуального восприятия.   
Затем элементы этих массивов объединяются в новом массиве так, как они выводились на экран. В массиве находятся r=L-1 максимальных элементов и на их место в массив записывается символ « | », означающий разделение. Все остальные элементы типа integer(цифры) просто стираются, после чего в массиве остаются лишь элементы a1…an, а также разделители « | ». Остается лишь последовательно вывести на экран элементы массива.

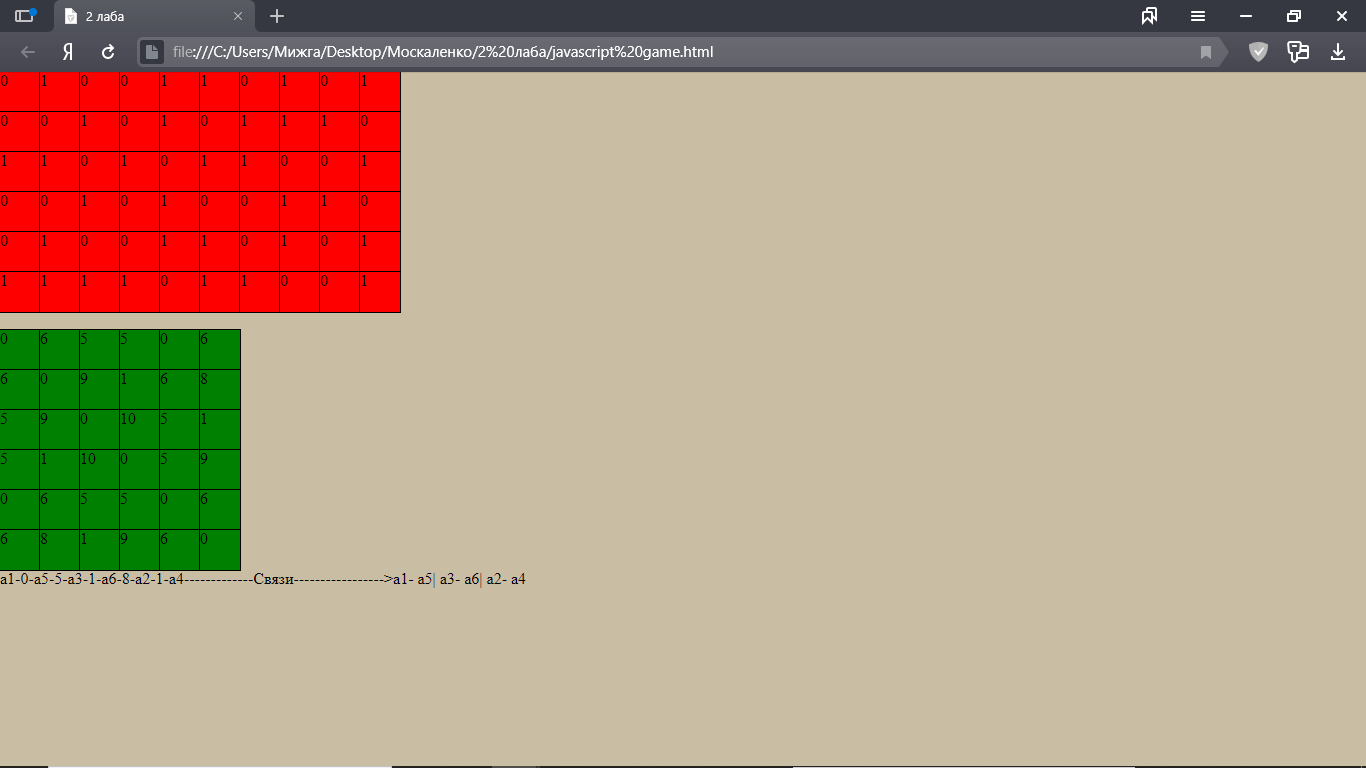


Рис.5 – матрица введенных данных, матрица несоответствий, цепочка с минимальным количеством несоответствий, кластеры.

**Код программы.**

var count=0; count1 = 0;

var command,command1;

var mas = [];var mas1 = []; mas2 = []; mas3 = []; mas4=[];mas5=[];mas6 = [];mas7 = [];

var block,v1,size,size1,but,val1,val2,l,m,mi,n,max,ma;

window.onload = function()

{

block = document.createElement('div');

document.body.appendChild(block);

val1 = parseInt(prompt("Введите количество строк матрицы", 2));

val2 = parseInt(prompt("Введите количество столбцов матрицы", 2));

l = parseInt(prompt("Введите l", 1));

matrix();

}

function matrix()

{

for(var i = 0;i<val1;i++)

{

mas[i] = [];

mas1[i] = [];

for (var j = 0;j<val2;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas[i][j] = div;

mas[i][j].style.width='40px';

mas[i][j].style.height='40px';

mas[i][j].style.background='red';

mas[i][j].style.outline = '1px solid #000';

mas[i][j].style.display = 'inline-block';

mas1[i][j] = parseInt(prompt("Введите число для элемента ["+i+"] ["+j+"]:", 0));

mas[i][j].innerHTML = mas1[i][j];

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

count = 0;

br1 = document.createElement('br');

document.body.appendChild(br1);

for(var i=0; i<val1;i++)

{

mas2[i] = [];

for(var j=0;j<val1;j++)

{

div = document.createElement('div');

document.body.appendChild(div);

mas2[i][j] = div;

mas2[i][j].style.width='40px';

mas2[i][j].style.height='40px';

mas2[i][j].style.background='green';

mas2[i][j].style.outline = '1px solid #000';

mas2[i][j].style.display = 'inline-block';

}

br1 = document.createElement('br');

document.body.appendChild(br1);

}

count = 0;

for(var i = 0;i<val1;i++)

{ mas3[i] = [];

for(var j=0;j<val1;j++)

{

for(var k=0;k<val2;k++)

{

if(mas1[i][k] != mas1[j][k]) count=count+1;

}

mas3[i][j] = count;

mas2[i][j].innerHTML = mas3[i][j];

if(i==j)

mas3[i][j] = val2\*1+1;

count=0;

}

}

count = 0;

for(var i=0;i<val1;i++)

{

mas5[i] = [];

min = mas3[i][0];

for(var j=0;j<val1;j++)

{

if(mas3[i][j]<min)

{

min = mas3[i][j];

m = j;

}

}

mas4[i] = min;

mas5[i] = m;

}

count = 0;

min = mas4[0]; mi = 0;

for(var i = 0;i<val1;i++)

{

if(mas4[i]<min)

{

min = mas4[i];

n = mi = i;

}

}

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

{

p = document.createElement('p');

p.style.display = 'inline-block';

document.body.appendChild(p);

p.innerHTML = 'a'+ (mi+1);

}

mas7[0] = 'a' + (mi+1);

for(var j = 0; j<val1;j++)

{

for(var k=0;k<val1;k++)

{

if(k==mi || j==mi)

{mas3[j][k] = val2\*1+1;}

}

}

for(var i = 0;i<(val1\*1-1);i++)

{

p = document.createElement('p');

p.style.display = 'inline-block';

document.body.appendChild(p);

p.innerHTML = '-' + mas4[mi] +'-'+ "a"+ (mas5[mi]\*1+1);

mas6[i] = mas4[mi]; mas7[i+1] = 'a'+ (mas5[mi]\*1+1);

for(var j = 0; j<val1;j++)

{

for(var k=0;k<val1;k++)

{

if(k==mi || j==mi)

{mas3[j][k] = val2\*1+1;}

}

}

for(var k=0;k<val1;k++)

{

min = mas3[0][0];

for(var j=0;j<val1;j++)

{

if(mas3[k][j]<min)

{

min = mas3[k][j]; mas4[k] = min;

mas5[k] = j;

}

}

}

mi=mas5[mi];

}

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

max = mas6[0];

for(var j=0;j<mas6.length;j++){

if(mas6[j]>max){

max = mas6[j];

ma = j;

}

}

mas6[ma] = ' | ';

}

for(var i=0;i<mas6.length;i++){

if(mas6[i]<(val2+1)) mas6[i] = ' - ';

}

//alert(mas6 +" "+mas7);

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

p = document.createElement('p');

p.style.display = 'inline-block';

document.body.appendChild(p);

p.innerHTML = "-------------Связи----------------->" + mas7[i];

}

for(var i=0;i<mas7.length-1;i++){

p = document.createElement('p');

p.style.display = 'inline-block';

document.body.appendChild(p);

p.innerHTML = ' '+mas6[i]+ ' ' + mas7[i+1];

}

}