**1.Завдання.**

*Завдання:*

**Вхідні дані:** кількість робочих комбінацій Np (Np = 8, 16, 32), кратність помилок ti ( ti=2), які

повинен виправляти код

**Вихідні дані:** надрукувати значення k, r, dmin, Gn,k (утворююча матриця)

1. Для заданих вхідних параметрів визначити кількість інформаційних символів k, кількість

перевірочних символів r, значення мінімальної кодової відстані dmin.

1. Для отриманого значення довжини коду n=k+r шляхом перебору всіх можливих двійкових комбінацій довжини n визначити Np перших комбінацій, мінімальна кодова відстань між якими рівна dmin
2. Отримати утворюючу матрицю

**2.Код програми**:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link href="style.css" rel="stylesheet">

<title>JS</title>

</head>

<body>

<div id="task">

<label>Enter the count of combinations: <input id="combs" type="text" value=""></label>

<button id="calc">Calcualte</button><br>

<label>Enter the rate of mistake: <input id="mistake" type="text" value=""></label>

<div>Rows(k): <span id="rows"></span></div>

<div>d<sub>min</sub>: <span id="dmin"></span></div>

<div>Cols(r): <span id="cols"></span></div>

<div id="checkingMatrix"></div>

<div id="creatingMatrix"></div>

<div id="codeDist"></div>

<div>Code combinations with two rate mistakes: <u><span id="propElems"></span></u></div>

</div>

<script>

'use strict';

let calc = document.getElementById('calc');

let rows = document.getElementById('rows');

let cols = document.getElementById('cols');

let codeCombins = [];

function showSolutions() {

calcRowsCount();

calcDmin();

calcColsCount();

generateTable(checkingMatrix, rows.innerHTML, cols.innerHTML);

fillCheckingMatrix();

generateTable(creatingMatrix, rows.innerHTML, +cols.innerHTML + +rows.innerHTML);

fillCreatingMatrixWithInform();

fillCreatingMatrixWithCheck();

defineCodeCombins();

generateTable(codeDist, combs.value, combs.value);

fillCodeDistTable();

findMistakes();

}

calc.onclick = showSolutions;

function calcRowsCount() {

let combs = document.getElementById('combs');

if (combs.value != 16) {

alert('Incorrect number of combinations');

return;

}

rows.innerHTML = log(2, combs.value);

}

function calcDmin() {

let mistake = document.getElementById('mistake');

if (mistake.value != 2) {

alert('Incorrect mistake rate');

return;

}

document.getElementById('dmin').innerHTML = 2 \* mistake.value + 1;

}

function calcColsCount() {

if (!document.getElementById('dmin').innerHTML) return;

let r = 0;

do {

r++;

} while ( r != log(2, 4 + r + 1) )

cols.innerHTML = r;

}

function generateTable(elem, rowsCount, colsCount) {

let table = '<table>';

for (let i = 0; i < rowsCount; i++) {

table += '<tr>';

for (let j = 0; j < colsCount; j++) {

table += '<td></td>';

}

table += '</tr>';

}

table += '</table>';

elem.innerHTML = table;

}

function fillCheckingMatrix() {

let matrix = document.querySelector('#checkingMatrix table');

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

for (let j = 0; j < matrix.rows[i].cells.length; j++) {

matrix.rows[i].cells[j].innerHTML = 1;

if (i == j) {

matrix.rows[i].cells[j].innerHTML = 0;

}

}

}

}

function fillCreatingMatrixWithInform() {

let matrix = document.querySelector('#creatingMatrix table');

let i = 0;

let rowLength = matrix.rows[i].cells.length - cols.innerHTML;

while (i < matrix.rows.length) {

for (let j = 0; j < rowLength; j++) {

matrix.rows[i].cells[j].innerHTML = 0;

if (i == j) {

matrix.rows[i].cells[j].innerHTML = 1;

}

}

i++;

}

}

function fillCreatingMatrixWithCheck() {

let matrix = document.querySelector('#creatingMatrix table');

let matrixCols = matrix.rows[0].cells.length - rows.innerHTML + 1;

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

for (let j = matrixCols; j < matrix.rows[i].cells.length; j++) {

matrix.rows[i].cells[j].innerHTML = 1;

if (j - i == 4) {

matrix.rows[i].cells[j].innerHTML = 0;

}

}

}

}

function defineCodeCombins() {

let matrix = document.querySelector('#creatingMatrix table');

let creatingMatrix = [];

codeCombins.push('0000000');

for (let row of matrix.rows) {

codeCombins.push( Array.from(row.cells).map(td => td.innerHTML).join('') );

creatingMatrix.push( Array.from(row.cells).map(td => td.innerHTML).join('') );

}

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

for (let j = i + 1; j < creatingMatrix.length; j++) {

let tmp = '';

for (let k = 0; k < creatingMatrix[j].length; k++) {

let XoR = creatingMatrix[i][k] ^ creatingMatrix[j][k];

tmp += XoR;

}

codeCombins.push(tmp);

tmp = '';

}

}

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

for (let j = i + 1; j < creatingMatrix.length; j++) {

for (let r = j + 1; r < creatingMatrix.length; r++) {

let tmp = '';

for (let k = 0; k < creatingMatrix[i].length; k++) {

let XoR = creatingMatrix[i][k] ^ creatingMatrix[j][k] ^ creatingMatrix[r][k];

// alert(i+': '+creatingMatrix[i] + '!!!' +j+': '+creatingMatrix[j] + '!!!' +r+': '+creatingMatrix[r]);

tmp += XoR;

}

codeCombins.push(tmp);

tmp = '';

}

}

}

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

for (let j = 1; j < 2; j++) {

for (let r = 2; r < 3; r++) {

for (let m = 3; m < 4; m++) {

let tmp = '';

for (let k = 0; k < creatingMatrix[i].length; k++) {

let XoR = creatingMatrix[i][k] ^ creatingMatrix[j][k] ^ creatingMatrix[r][k] ^ creatingMatrix[m][k];

tmp += XoR;

}

codeCombins.push(tmp);

tmp = '';

}

}

}

}

// codeCombins.forEach((comb, i) => task.insertAdjacentHTML('beforeend', `<p>${i + 1}: ${comb}</p>`));

}

function fillCodeDistTable() {

let matrix = document.querySelector('#codeDist table');

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

for (let j = 0; j < codeCombins.length; j++) {

let tmp = '';

let count = 0;

for (let k = 0; k < codeCombins[j].length; k++) {

let XoR = codeCombins[i][k] ^ codeCombins[j][k];

tmp += XoR;

}

for (let c = 0; c < tmp.length; c++) {

if (tmp[c] != 0) count++;

}

matrix.rows[i].cells[j].innerHTML = count;

tmp = '';

}

}

}

function findMistakes() {

let matrix = document.querySelector('#codeDist table');

let dMin = +mistake.value + 1;

let propElemsPos = Array.from(matrix.rows[0].cells).map((td, i) => {

if (+td.innerHTML == dMin) return `V${i + 1} `;

}).join('');

document.getElementById('propElems').innerHTML = propElemsPos;

}

function log(x, y) {

return Math.log(y) / Math.log(x);

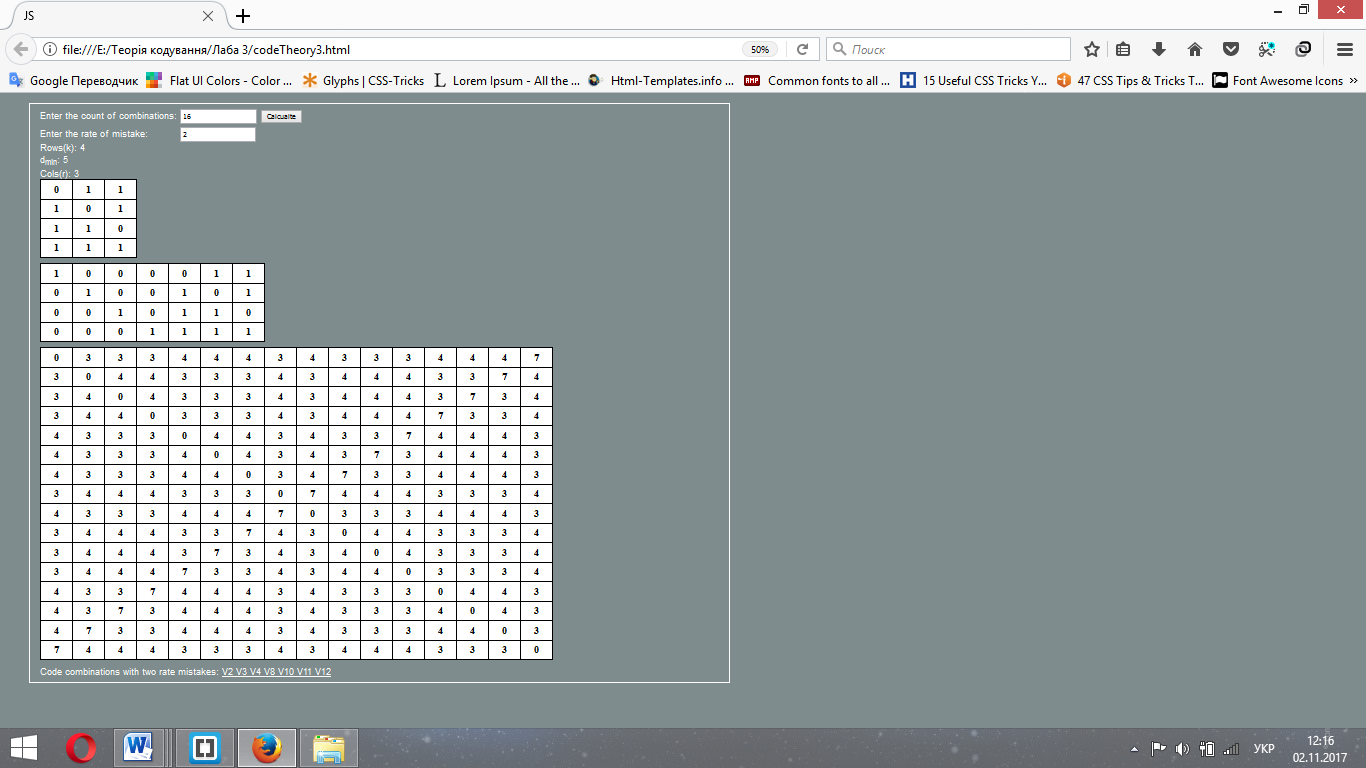
}

</script>

</body>

</html>

**3.Вікно інтерфейсу**



**4.Мова програмування – JavaScript.**