Лабораторная работа No2

**Исследование способов структурного тестирования программного обеспечения**

**Цель работы**

Исследовать основные подходы к структурному тестированию программного обеспечения. Приобрести практические навыки построения графа потоков управления и определения независимых ветвей программы.

**Вариант - 8**

Варианты заданий соответствуют заданиям по лабораторной работе No1. По варианту задаются требования к программам. Для каждой из них необходимо:

1) Написать программу, выполняющую заданные действия.

2) Построить граф потоков управления.

3) Вычислить цикломатическое число для построенного графа потоков

управления.

4) Определить независимые ветви программы.

**Ход работы**

var fs = require('fs');

// MARK: FIRST TASK

// Rotate by 90 \* k degrees function

function rotate(matrix, k) {

let isSquare = isSquareMatrix(matrix)

if (isSquare == 'neverno') return 'matrix is no square'

var result = [...matrix];

let size = matrix[0].length

var delta = matrix

for (let d = 0; d < k; d++) {

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

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

result[i][j] = delta[size - j - 1][i]

}

}

delta = result

}

return result

}

function isSquareMatrix(matrix) {

// console.log(matrix)

for(let item in matrix){

if(matrix[item].length != matrix[Number(item + 1)].length){

return 'neverno'

}

}

}

// Один тест на количество

let matrix = [

[0,0],

[0,1]

]

let k = 4;

let rotated = rotate(matrix, k)

console.log(`Task 1 `, rotated)

// MARK: SECOND TASK

function is5CharactersE(stroka) {

if(stroka.includes("eeeee")){

return true

}

else{

return false

}

}

// В начале, в конце, в середине, присутствуют не все e, отсутствуют e

let str = 'i tuteeeee'

console.log('Имеются ли в строке 5 идущих подряд букв e', is5CharactersE(str))

function readFile(name){

var data = fs.readFileSync(name, 'utf8');

let count = 0;

for(let stroka of data.split('\n')){

let curStroke = String(stroka);

if(curStroke.length <= 1) count++;

}

return count

}

console.log('Количество пустых строк',readFile('file.txt'))

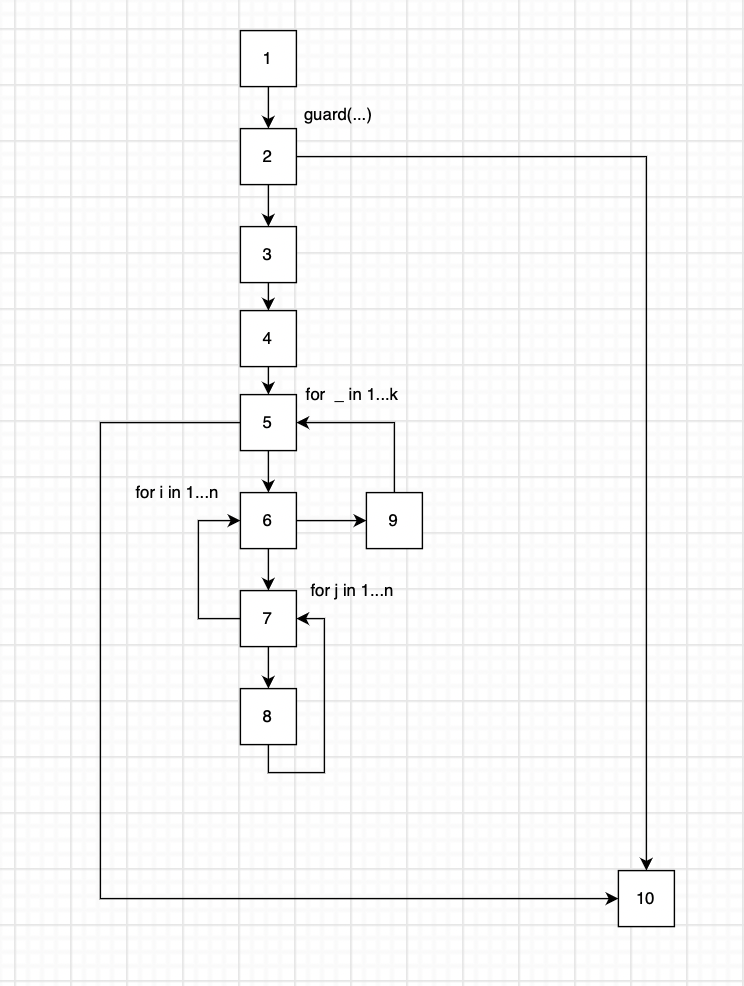


Рисунок – 1

C = 13 – 10 + 2 = 5

1. 1 2 10
2. 1 2 3 4 5 10
3. 1 2 3 4 5 6 9 5 10
4. 1 2 3 4 5 6 7 6 9 5 10
5. 1 2 3 4 5 6 7 8 7 6 9 5 10

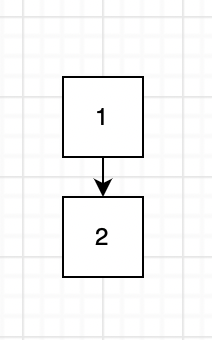


Рисунок – 2

C = 1 – 2 + 2 = 1

1. 1 2

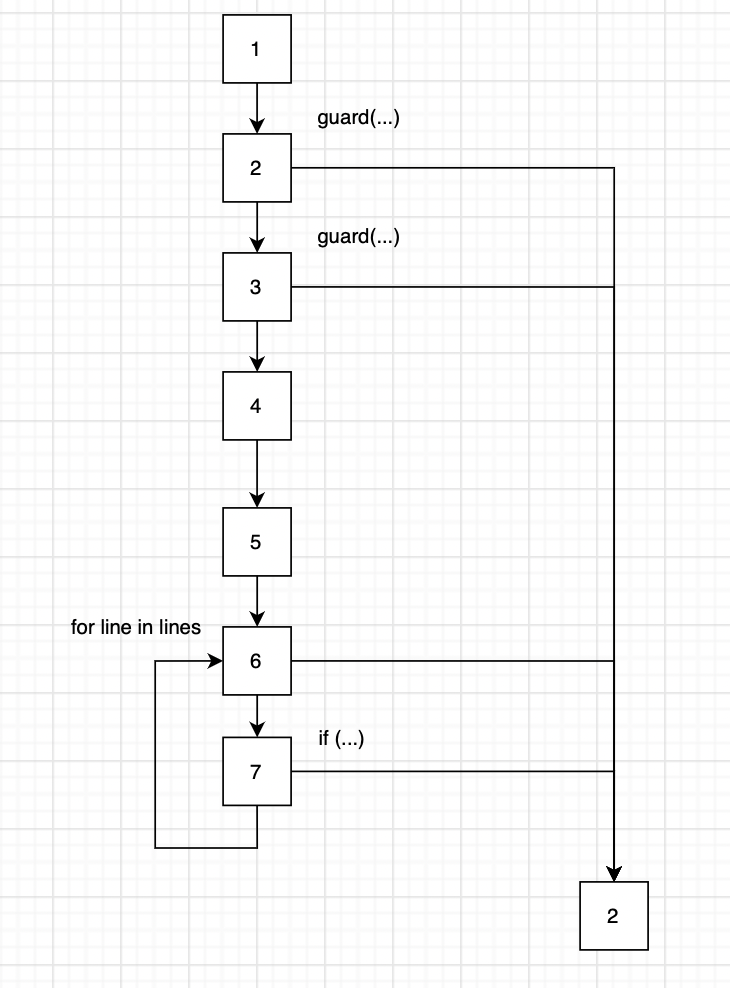


Рисунок – 3

C = 11 – 8 + 2 = 5

1. 1 2 8
2. 1 2 3 8
3. 1 2 3 4 5 6 8
4. 1 2 3 4 5 6 7 8
5. 1 2 3 4 5 6 7 6 8

**Вывод:**

Исследованны основные подходы к структурному тестированию программного обеспечения. Приобретены практические навыки построения графа потоков управления и определения независимых ветвей программы