ЛАБОРАТОРНАЯ РАБОТА № 1

НАЗВАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

ВЫПОЛНИЛ: ст. гр. 4411 Гайфуллин Д. Р.

ВАРИАНТ № 2

*Цель работы* – исследование парольных подсистем аутентификации пользователей. Реализация простейшего генератора паролей, обладающего требуемой стойкостью к взлому.

*P*= 10-5

*V*= 3 паролей/мин

*T*= 10 дней

*S\**= =

В качестве алфавита символов, используемых при генерации пароля, были использованы следующие наборы:

1. Латинские прописные буквы (A-Z)

2. Русские строчные буквы (а-я)

Мощность данного набора *A* =59

При минимальном значении *L*= 6 выполняется условие *S\**≤*S*=*AL*.

Примеры паролей, сгенерированных программой Simple Passwords:

1) ZLиWлэ

2) Bвбсбщ

3) ZDEнйт

4) EдгFDь

5) яаьыLф

6) мHUйWW

7) лWDCфW

8) вXBCHY

9) IBTVгP

10) эAтLKо

11) иыSIET

12) тLJJQб

13) жлйфэI

14) тFRBгй

15) оиODHу

16) щHцHею

17) XBVBмX

18) зJGIаB

19) LIPAъф

20) BNгыбй

ТЕКСТ ПРОГРАММЫ

main.js

import readline from 'readline';

import {calculateA, calculateSL, calculateL} from './calculate.js';

import {generateAlphabet, generatePasswords, printResult} from './utils.js';

*const* rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

*const* P = 0.00001, *// P - вероятность подбора пароля злоумышленником.*

    V = 4320, *// V - скорость перебора пароля паролей злоумышленником (пароль/день).*

    T = 10, *// T - максимальный срок действия пароля.*

    intervalEngUpper = [65, 90], *// Интервал ASCII-кода для латиницы в верхнем регистре.*

    intervalRusLower = [1072, 1103]; *// Интервал ASCII-кода для кириллицы в нижнем регистре.*

function numberInput(L, alphabet) {

    rl.question('Число паролей: ', (answer) => {

        if (!isNaN(answer)) {

            if(answer > 0) {

*const* passwords = generatePasswords(Math.floor(+answer), L, alphabet);

                printResult(L, passwords);

            } else {

                console.log('Неверный ввод.')

            }

        } else {

            console.log('Неверный ввод.')

        }

        workCycle(L, alphabet);

    });

};

function workCycle(L, alphabet) {

    rl.question('0 - сгенерировать пароли;\n1 - выход.\nКоманда: ', (answer) => {

        switch(answer) {

            case '0':

                numberInput(L, alphabet);

                break;

            case '1':

                rl.close();

                break;

            default:

                console.log('Неверный ввод.')

                workCycle(L, alphabet);

                break;

        };

    });

};

function init() {

*const* A = calculateA(intervalEngUpper, intervalRusLower);

*const* SL = calculateSL(P, V, T);

*const* L = calculateL(A, SL);

*const* alphabet = generateAlphabet(intervalEngUpper, intervalRusLower);

*return* {L, alphabet};

};

function main() {

*const* {L, alphabet} = init();

    workCycle(L, alphabet);

};

main();

utils.js

export function getRandInt(min, max) {

*return* Math.floor(Math.random() \* (max - min + 1) + min)

};

export function getSymbols(firstASCII, lastASCII) {

*var* symbols = [];

    for (*var* i = firstASCII; i <= lastASCII; i++) {

        symbols.push(String.fromCharCode(i));

    };

*return* symbols;

}

export function generateAlphabet(intervalEngUpper, intervalRusLower) {

*return*  getSymbols(intervalEngUpper[0], intervalEngUpper[1]).

        concat(getSymbols(intervalRusLower[0], intervalRusLower[1]));

};

export function generatePassword(passwordLength, alphabet) {

*var* password = '';

    for (*var* i = 0; i < passwordLength; i++) {

        password += alphabet[getRandInt(0, alphabet.length - 1)];

    }

*return* password;

};

export function generatePasswords(passwordsQuantity, passwordLength, alphabet) {

*const* passwords = [];

    for (*var* i = 0; i < passwordsQuantity; i++) {

        passwords.push(generatePassword(passwordLength, alphabet));

    }

*return* passwords;

};

export function printResult(L, passwords) {

    console.log(

        `Рассчитанная длина пароля: ${L}\nПароли:`

    );

    passwords.forEach(password => {

        console.log(`${password}`);

    })

};

calculate.js

export function calculateA (intervalEngUpper, intervalRusLower) {

*return* intervalEngUpper[1] - intervalEngUpper[0] + intervalRusLower[1] - intervalRusLower[0] + 2;

};

export function calculateSL(P, V, T) {

*return* Math.ceil(V \* T / P);

};

export function calculateL (A, SL) {

*var* L = 1;

    while (!(A \*\* L >= SL)) {

        L++;

    };

*return* L;

};

Примеры сгенерированных программой паролей:

1. яDKкеи
2. вьвдвV
3. фзJNэS
4. NEлцFK
5. эамчоP
6. уLлдVс
7. KнFEкд
8. ззWFаь
9. жBюKнт
10. уBNмBя
11. эHоинW
12. HGэуаL
13. цSыWзM
14. свмYJо
15. зXвдEы
16. Cшуэду
17. шзHуиQ
18. щLоцгP
19. AлшьзT
20. пгжBUв