УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.1

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

Крутько А.А.

Проверила:

Данилова Г.В.

Группа 251004

Минск 2022

Задание:  


Код программы на **Delphi**:

Program LabFirstBlockThird;

uses

System.SysUtils, Windows;

Function ChooseOption(): String;

Var

Input: String;

IsRight: Boolean;

Begin

Writeln('Введите console, если хотите использовать консоль, file, если файл');

Repeat

IsRight := True;

Readln(Input);

Input := LowerCase(Input);

If (Input <> 'console') and (Input <> 'file') then

Begin

IsRight := False;

Writeln('Повторите ввод!');

End;

Until IsRight;

ChooseOption := Input;

End;

Function ChooseConsoleWayToFill(): String;

Var

Input: String;

IsRight: Boolean;

Begin

Writeln('Введите console, если хотите сами ввести данные, random, чтобы   
 сгенерировать случайные строки');

Repeat

IsRight := True;

Readln(Input);

Input := LowerCase(Input);

If (Input <> 'console') and (Input <> 'random') then

Begin

IsRight := False;

Writeln('Повторите ввод!');

End;

Until IsRight;

ChooseConsoleWayToFill := Input;

End;

Function InputStrConsole(): String;

Var

Str: String;

IsIncorrect: Boolean;

Begin

Repeat

IsIncorrect := True;

Readln(Str);

If (Str[1] = ' ') or (Length(Str) = 0) then

Begin

IsIncorrect := False;

Writeln('Повторите ввод!');

End;

Until IsIncorrect;

InputStrConsole := Str;

End;

Function InputAmountOfOccurrences(): Integer;

Const

MIN\_NUM = 0;

MAX\_NUM = 255;

Var

K: Integer;

IsIncorrect: Boolean;

Begin

Write('Введите число, количество вхождений строки 1 в строку 2, k = ');

Repeat

Try

IsIncorrect := True;

Readln(K);

Except

IsIncorrect := False;

Writeln('Повторите ввод');

End;

If IsIncorrect and ((K < MIN\_NUM) or (K > MAX\_NUM)) Then

Begin

IsIncorrect := False;

Writeln('Повторите ввод');

End;

Until IsIncorrect;

InputAmountOfOccurrences := K;

End;

Function IsCorrectStrFromFile(Var F: TextFile): Boolean;

Var

Str1, Str2: String;

IsIncorrect: Boolean;

Begin

IsIncorrect := True;

If Not Eof(F) Then

Begin

Readln(F, Str1);

If (Str1[1] = ' ') or (Length(Str1) = 0) Then

Begin

IsIncorrect := False;

Writeln('Ошибка, проверьте вашу 1 строку на соответствие   
 данных');

End

End

Else

Begin

IsIncorrect := False;

Writeln('В файле нет данных для первой строки');

End;

If Not Eof(F) Then

Begin

Readln(F, Str2);

If (Str2[1] = ' ') or (Length(Str2) = 0) Then

Begin

IsIncorrect := False;

Writeln('Ошибка, проверьте вашу 2 строку на соответствие   
 данных');

End

End

Else

Begin

IsIncorrect := False;

Writeln('В файле нет данных для второй строки');

End;

IsCorrectStrFromFile := IsIncorrect;

End;

Function IsCorrectAmountFromFile(Var F: TextFile): Boolean;

Const

MIN\_NUM = 0;

MAX\_NUM = 255;

Var

K: Integer;

IsIncorrect: Boolean;

Begin

IsIncorrect := True;

If Not Eof(F) Then

Begin

Try

Readln(F, K);

Except

IsIncorrect := False;

Writeln('Повторите ввод');

End;

End

Else

Begin

Writeln('Нет количества вхождений строк');

IsIncorrect := False;

End;

If IsIncorrect and ((K < MIN\_NUM) or (K > MAX\_NUM)) Then

IsIncorrect := False;

IsCorrectAmountFromFile := IsIncorrect;

End;

Function WayToFile(Flag: String): String;

Var

Way, Str1, Str2: String;

F: TextFile;

IsIncorrect: Boolean;

Begin

Repeat

IsIncorrect := True;

Writeln('Введите путь к файлу');

Readln(Way);

AssignFile(F, Way);

If FileExists(Way) Then

Begin

If Flag = 'In' Then

Begin

Reset(F);

IsIncorrect := IsCorrectStrFromFile(F);

If (IsIncorrect) Then

IsIncorrect := IsCorrectAmountFromFile(F);

If (IsIncorrect and Not Eof(F)) Then

Begin

IsIncorrect := False;

Writeln('уберите лишние данные');

End;

Close(F);

End;

End

Else

Begin

Writeln('Файла в заданном пути нет');

IsIncorrect := False;

End;

Until IsIncorrect;

WayToFile := Way;

End;

Function CalculateAnswer(Str1: String; Str2: String; K: Integer): Integer;

Var

CountN, N, I: Integer;

Begin

CountN := 0;

N := 0;

I := 1;

For I := 1 to Length(Str2) Do

Begin

If Copy(Str2,I,Length(Str1)) = Str1 Then

Begin

Inc(CountN);

If CountN = K Then

N := I;

End;

End;

CalculateAnswer := N;

End;

Function TakeStrFromFile(Var F: TextFile): String;

Var

Str: String;

Begin

Readln(F, Str);

TakeStrFromFile := Str;

End;

Function InputAmountOfOccurrencesFromFile(Var F: TextFile): Integer;

Var

K: Integer;

IsIncorrect: Boolean;

Begin

Readln(F, K);

InputAmountOfOccurrencesFromFile := K;

End;

Function RandomStrConsole(): String;

Var

I, N: Integer;

Str: String;

Begin

Randomize;

N := random(255) + 1;

For I := 1 to N Do

Str := Str + Chr(Random((Ord('z') - Ord('A'))) + Ord('A'));

RandomStrConsole := Str;

End;

Function ConsoleInput(InputChoice: String): Integer;

Var

IsLengthRight: Boolean;

Str1, Str2: String;

K, N: Integer;

Begin

If InputChoice = 'console' Then

Begin

Repeat

IsLengthRight := True;

Writeln('Введите строку 1(первый символ не должен быть пробелом)');

Str1 := InputStrConsole();

Writeln('Введите строку 2(первый символ не должен быть пробелом)');

Str2 := InputStrConsole();

If Length(Str1) > Length(Str2) Then

Begin

Writeln('Повторите ввод строк, чтобы размер вашей первой строки   
 был МЕНЬШЕ размера второй строки');

IsLengthRight := False;

End;

Until IsLengthRight;

End

Else

Begin

Repeat

IsLengthRight := True;

Str1 := RandomStrConsole();

Str2 := RandomStrConsole();

If Length(Str1) > Length(Str2) Then

IsLengthRight := False;

Until IsLengthRight;

Writeln('Строки сгенерированы.');

Writeln(Str1);

Writeln(Str2);

End;

K := InputAmountOfOccurrences();

N := CalculateAnswer(Str1, Str2, K);

ConsoleInput := N;

End;

Function FileInput(): Integer;

Var

Way, Str1, Str2: String;

F: TextFile;

N, K: Integer;

Begin

Way := WayToFile('In');

AssignFile(F, Way);

Reset(F);

Str1 := TakeStrFromFile(F);

Str2 := TakeStrFromFile(F);

K := InputAmountOfOccurrencesFromFile(F);

Close(F);

N := CalculateAnswer(Str1, Str2, K);

FileInput := N;

End;

Function TakeFinalInformation(InputChoice: String): Integer;

Var

Way: String;

N: Integer;

Begin

If (InputChoice = 'console') Then

Begin

InputChoice := ChooseConsoleWayToFill();

N := ConsoleInput(InputChoice);

End

Else

Begin

N := FileInput;

Writeln('Данные считаны успешно');

End;

TakeFinalInformation := N;

End;

Procedure WriteAnswerConsole(N: Integer);

Begin

Writeln('Программа завершилась успешно.',#10,'номер позиции k-го вхождения: ',   
 N);

End;

Procedure WriteAnswerFile(N: Integer);

Var

Way: String;

F: TextFile;

Begin

Way := WayToFile('Out');

AssignFile(F, Way);

ReWrite(F);

Writeln(F, 'номер позиции k-го вхождения: ', N);

Close(F);

End;

Procedure OutputAnswer(OutputChoice: String; N: Integer);

Begin

If (OutputChoice = 'console') Then

WriteAnswerConsole(N)

Else

WriteAnswerFile(N);

End;

Var

InputChoice, OutputChoice: String;

N: Integer;

Begin

Writeln('Данная программа находит количество вхождений(задаваемого пользователем)   
 первой строки во вторую');

InputChoice := ChooseOption();

N := TakeFinalInformation(InputChoice);

OutputChoice := ChooseOption();

OutputAnswer(OutputChoice, N);

Readln;

End.

Код программы на **C++**:

#include <iostream>

#include <iomanip>

#include <string>

#include <fstream>

#include <algorithm>

using namespace std;

int inputAmountOfOccurences() {

int n;

bool isIncorrect;

const int MIN\_NUM = 1;

cout << "Введите число, количество вхождений строки 1 в строку 2, k = " << endl;

do {

isIncorrect = false;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cout << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && n < MIN\_NUM) {

cout << "Число > 0 " << endl;

isIncorrect = true;

}

} while (isIncorrect);

return n;

}

string chooseOption()

{

bool isIncorrect;

string input;

cout << "Введите console, если хотите использовать консоль, file, если файл" << endl;

do

{

isIncorrect = false;

cin >> input;

transform(input.begin(), input.end(), input.begin(), ::tolower);

if (input != "console" && input != "file")

{

cerr << "Повторите ввод" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return input;

}

string chooseConsoleWayToFill()

{

bool isIncorrect;

string input;

cout << "Введите console, если хотите ввести данные сами, random, если хотите сгенерировать рандомные строки" << endl;

do

{

isIncorrect = false;

cin >> input;

transform(input.begin(), input.end(), input.begin(), ::tolower);

if (input != "console" && input != "random")

{

cerr << "Повторите ввод" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return input;

}

string inputStrConsole()

{

string str;

bool isIncorrect;

do

{

isIncorrect = false;

cin >> str;

if (str.empty() || str[0] == ' ')

{

isIncorrect = true;

cerr << "Повторите ввод" << endl;

}

} while (isIncorrect);

return str;

}

int calculateAnswer(string str1, string str2, int k)

{

int countN = 0, n = 0;

for (int i = 0; i <= str2.length() - str1.length(); i++)

{

string str = str2.substr(i, str1.length());

if (str == str1) // (int j = str2.find(str1, i) == i)

{

++countN;

if (countN == k)

{

n = i + 1;

}

}

}

return n;

}

string randomStrConsole()

{

int n;

string str = "";

string upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

string lower = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

transform(lower.begin(), lower.end(), lower.begin(), ::tolower);

string numbers = "0123456789";

string allCurrent = upper + lower + numbers;

n = rand() % 256;

for (int i = 0; i < n; i++)

{

str += allCurrent[rand() % 62];

}

return str;

}

int consoleInput(string input)

{

bool isLengthRight;

int k, n;

string str1, str2;

if (input == "console")

{

do {

cerr << "Введите строку 1(первый символ не должен быть пробелом)" << endl;

str1 = inputStrConsole();

cout << "Введите строку 2(первый символ не должен быть пробелом)" << endl;

str2 = inputStrConsole();

isLengthRight = (str1.length() > str2.length()) ? true : false;

if (isLengthRight)

{

cerr << "Повторите ввод строк, чтобы размер вашей первой строки был   
 МЕНЬШЕ размера второй строки" << endl;

}

} while (isLengthRight);

}

else

{

do {

str1 = randomStrConsole();

str2 = randomStrConsole();

isLengthRight = (str1.length() > str2.length()) ? true : false;

} while (isLengthRight);

cout << "Строки сгенерированы" << endl;

cout << str1 << endl;

cout << str2 << endl;

}

k = inputAmountOfOccurences();

n = calculateAnswer(str1, str2, k);

return n;

}

bool checkExtension(string path)

{

bool isInCorrect = true;

if ((path[path.length() - 1] == 't') && (path[path.length() - 2] == 'x') &&   
 (path[path.length() - 3] == 't') && (path[path.length() - 4] == '.'))

isInCorrect = false;

return isInCorrect;

}

bool isFileCorrect(string way)

{

int k;

bool isIncorrect = false;

string str1, str2;

ifstream fin(way);

if (!fin.eof())

{

getline(fin, str1);

if (str1.empty() || str1[0] == ' ')

{

isIncorrect = true;

cerr << "Проверьте правильность первой строки" << endl;

}

}

else

{

isIncorrect = true;

cerr << "Нет данных для первой строки" << endl;

}

if (!fin.eof())

{

if (!isIncorrect)

{

getline(fin, str2);

if (str2.empty() || str2[0] == ' ')

{

isIncorrect = true;

cerr << "Проверьте правильность второй строки" << endl;

}

}

}

else

{

cerr << "Нет данных для второй строки" << endl;

isIncorrect = true;

}

if (!fin.eof())

{

if (!isIncorrect)

{

fin >> k;

if (fin.fail())

{

isIncorrect = true;

cerr << "Число вхождений не является числом" << endl;

}

if (!isIncorrect && k < 1)

{

isIncorrect = true;

cerr << "Число вхождений > 1" << endl;

}

}

}

if (!fin.eof() && !isIncorrect)

{

isIncorrect = true;

cerr << "Уберите лишние данные" << endl;

}

fin.close();

return isIncorrect;

}

string takeFileWay(string temp)

{

string way;

bool isIncorrect;

do

{

isIncorrect = false;

cout << "Введите путь к файлу" << endl;

cin >> way;

if (temp == "In")

{

ifstream fin(way);

if (!fin.is\_open() && checkExtension)

{

cout << "Проверьте правильность и нахождение вашего файла по заданному   
 пути" << endl;

isIncorrect = true;

}

else

{

isIncorrect = isFileCorrect(way);

}

fin.close();

}

else

{

ifstream fout(way);

if (!fout.is\_open() && checkExtension)

{

cout << "Проверьте правильность и нахождение вашего файла по заданному   
 пути" << endl;

isIncorrect = true;

}

fout.close();

}

} while (isIncorrect);

return way;

}

int fileInput(string way)

{

int n, k;

string str1, str2;

ifstream fin(way, ios::beg);

getline(fin, str1);

getline(fin, str2);

fin >> k;

fin.close();

n = calculateAnswer(str1, str2, k);

return n;

}

int takeFinalInformation(string input)

{

string way, str1, str2;

int n;

if (input == "console")

{

input = chooseConsoleWayToFill();

n = consoleInput(input);

}

else

{

way = takeFileWay("In");

n = fileInput(way);

}

return n;

}

void outputFinalInformation(int n, string outputChoice)

{

string way;

if (outputChoice == "console")

{

cout << "Программа завершилась успешно\n" << "номер позиции k-го вхождения: " <<   
 n;

}

else

{

way = takeFileWay("Out");

ofstream fout(way);

fout << "Программа завершилась успешно\n" << "номер позиции k-го вхождения: " <<   
 n;

fout.close();

}

}

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

int n;

string inputChoice, outputChoice;

cout << "Данная программа находит количество вхождений k 1-ой строки во 2-ую строку"   
 << endl;

inputChoice = chooseOption();

n = takeFinalInformation(inputChoice);

outputChoice = chooseOption();

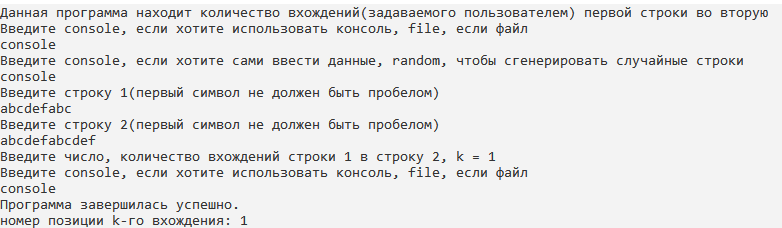
outputFinalInformation(n, outputChoice);

}

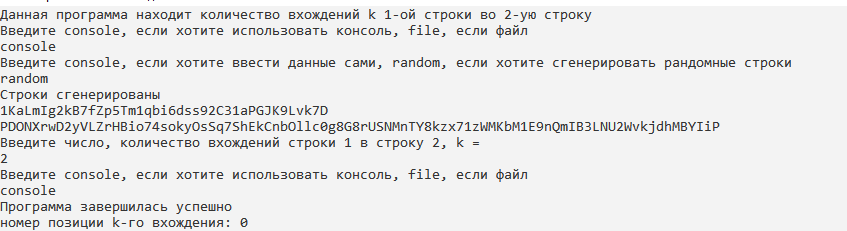
Код программы на **Java**:

import java.util.Scanner;  
import java.io.\*;  
  
public class LabFirstBlockThird {  
 static Scanner scan = new Scanner(System.in);  
  
 public static int inputAmountOfOccurences() {  
 int n = 0;  
 boolean isIncorrect;  
 final int MIN\_NUM = 1, MAX\_NUM = 255;  
 System.out.println("Введите число, количество вхождений строки 1 в   
 строку 2, k = ");  
 do {  
 isIncorrect = false;  
 try {  
 n = Integer.parseInt(scan.nextLine());  
 } catch (NumberFormatException E) {  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && (n < MIN\_NUM || n > MAX\_NUM)) {  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return n;  
 }  
  
 public static String chooseOption(){  
 boolean isIncorrect;  
 String input;  
 System.out.println("Введите console, если хотите использовать консоль, file, если файл");  
 do {  
 isIncorrect = false;  
 input = scan.nextLine();  
 if (!input.equalsIgnoreCase("console") && !input.equalsIgnoreCase("file")){  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return input;  
 }  
  
 public static String chooseConsoleWayToFill() {  
 boolean isIncorrect;  
 String input;  
 System.out.println("Введите console, если хотите ввести данные с консоли, random, чтобы сгенерировать строки файл");  
 do {  
 isIncorrect = false;  
 input = scan.nextLine();  
 if (!input.equalsIgnoreCase("console") && !input.equalsIgnoreCase("random")){  
 System.err.println("Повторите ввод: ");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return input;  
 }  
  
 public static String inputStrConsole() {  
 String str;  
 boolean isIncorrect;  
 do {  
 isIncorrect = false;  
 str = scan.nextLine();  
 if (str.isEmpty() || str.charAt(0) == ' ') { //str.equals("");  
 isIncorrect = true;  
 System.err.println("Повторите ввод");  
 }  
 } while (isIncorrect);  
 return str;  
 }  
  
 public static int calculateAnswer(String str1, String str2, int k) {  
 int countN = 0, n = 0;  
 for (int i = 0; i <= str2.length() - str1.length(); i++) {  
 if (str2.substring(i, str1.length() + i).equals(str1)) {  
 ++countN;  
 if (countN == k) {  
 n = i + 1;  
 }  
 }  
 }  
 return n;  
 }  
  
 public static String randomStrConsole() {  
 double n;  
 String str = "", upper, lower, numbers, allCurrent;  
 upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
 lower = upper.toLowerCase();  
 numbers = "0123456789";  
 allCurrent = upper + lower + numbers;  
 n = Math.random() \* 255;  
 for (int i = 0; i < n; i++) {  
 str += allCurrent.charAt((int) Math.round(Math.random() \* 61));  
 }  
 return str;  
 }  
  
 public static int consoleInput(String input) {  
 boolean isLengthRight;  
 int k, n;  
 String str1 = "", str2 = "";  
 if (input.equalsIgnoreCase("console")) {  
 do {  
 isLengthRight = false;  
 System.out.println("Введите строку 1(первый символ не должен   
 быть пробелом)");  
 str1 = inputStrConsole();  
 System.out.println("Введите строку 2(первый символ не должен   
 быть пробелом)");  
 str2 = inputStrConsole();  
 if (str1.length() > str2.length()) {  
 System.err.println("Повторите ввод строк, чтобы размер   
 вашей первой строки был МЕНЬШЕ   
 размера второй строки");  
 isLengthRight = true;  
 }  
 } while (isLengthRight);  
 } else {  
 do {  
 isLengthRight = false;  
 str1 = randomStrConsole();  
 str2 = randomStrConsole();  
 if (str1.length() > str2.length()) {  
 isLengthRight = true;  
 }  
 } while (isLengthRight);  
 System.out.println("Строки сгенерированы.");  
 System.out.println(str1);  
 System.out.println(str2);  
 }  
 k = inputAmountOfOccurences();  
 n = calculateAnswer(str1, str2, k);  
 return n;  
 }  
  
 public static boolean isTakeInformationFromFile(String way) {  
 int k = 0;  
 final int MIN\_NUM = 1, MAX\_NUM = 255;  
 String str1, str2;  
 boolean isIncorrect = false;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 if ((str1 = br.readLine()) != null) {  
 if (str1.isEmpty() || str1.charAt(0) == ' ') {   
 isIncorrect = true;  
 System.err.println("Проверьте правильность введенной 1-ой   
 строки");  
 }  
 } else {  
 isIncorrect = true;  
 System.err.println("В файле нет данных для 1-ой строки");  
 }  
 if (!isIncorrect & (str2 = br.readLine()) != null) {  
 if (str2.isEmpty() || str2.charAt(0) == ' ') {   
 isIncorrect = true;  
 System.err.println("Проверьте правильность введенной 2-ой   
 строки");  
 }  
 } else {  
 isIncorrect = true;  
 System.err.println("В файле нет данных для 2-ой строки");  
 }  
 if (!isIncorrect & str1.length() > str2.length()) {  
 System.err.println("Размер вашей строки 2 должен быть БОЛЬШЕ   
 размера строки 1");  
 isIncorrect = true;  
 }  
 if (!isIncorrect) {  
 try {  
 k = Integer.parseInt(br.readLine());  
 } catch (NumberFormatException E) {  
 isIncorrect = true;  
 System.err.println("Проверьте соответствие k к целому   
 числу");  
 }  
 if (!isIncorrect & (k < MIN\_NUM || k > MAX\_NUM)) {  
 isIncorrect = true;  
 System.err.println("Ваше число должно быть в диапазоне "   
 + MIN\_NUM + " до " + MAX\_NUM);  
 }  
 }  
 if (!isIncorrect & br.readLine() != null) {  
 isIncorrect = true;  
 System.out.println("Уберите лишние данные");  
 }  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 isIncorrect = true;  
 }  
 return isIncorrect;  
 }  
  
 public static String takeFileWay(String temp) {  
 String way;  
 boolean isIncorrect;  
 do {  
 isIncorrect = false;  
 System.out.println("Введите путь к файлу");  
 way = scan.nextLine();  
 File file = new File(way);  
 if (!file.exists() || !file.canWrite() ||!file.canRead() ||   
 !way.endsWith(".txt") || file.isDirectory()) {  
 isIncorrect = true;  
 System.err.println("Файл не найден, или неверный формат   
 файла; повторите попытку ввода");  
 } else if (temp.equals("In")) {  
 isIncorrect = isTakeInformationFromFile(way);  
 }  
 } while (isIncorrect);  
 return way;  
 }  
  
 public static String inputStringFromFile(String way, int p) {  
 String str = "";  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 if (p == 1) {  
 br.readLine();  
 }  
 str = br.readLine();  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 return str;  
 }  
  
 public static int inputAmountOfOccurencesFromFile(String way) {  
 int k = 0;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 br.readLine();  
 br.readLine();  
 k = Integer.parseInt(br.readLine());  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 return k;  
 }  
 public static int takeFinalInformation(String input) {  
 String way, str1, str2;  
 int n = 0, k;  
 if (input.equalsIgnoreCase("console")) {  
 input = chooseConsoleWayToFill();  
 n = consoleInput(input);  
 } else {  
 way = takeFileWay("In");  
 str1 = inputStringFromFile(way, 0);  
 str2 = inputStringFromFile(way, 1);  
 k = inputAmountOfOccurencesFromFile(way);  
 n = calculateAnswer(str1, str2, k);  
 System.out.println("Данные считаны успешно");  
 }  
 return n;  
 }  
  
 public static void writeAnswerConsole(int n) {  
 System.out.println("Программа завершилась успешно.\n" + "номер   
 позиции k-го вхождения: " + n);  
 }  
  
 public static void writeAnswerFile(String way, int n) {  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, false))) {  
 pw.write("Программа завершилась успешно.\n" + "номер позиции k-го   
 вхождения: " + n);  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 }  
 public static void outputAnswer(String outputChoice, int n) {  
 String way;  
 if (outputChoice.equalsIgnoreCase("console")) {  
 writeAnswerConsole(n);  
 } else {  
 way = takeFileWay("Out");  
 writeAnswerFile(way, n);  
 }  
 }  
  
 public static void main(String[] args) {  
 String inputChoice, outputChoice;  
 int n;  
 System.out.println("Данная программа находит количество   
 вхождений(задаваемого пользователем) первой строки во   
 вторую");  
 inputChoice = chooseOption();  
 n = takeFinalInformation(inputChoice);  
 outputChoice = chooseOption();  
 outputAnswer(outputChoice, n);  
 }  
}

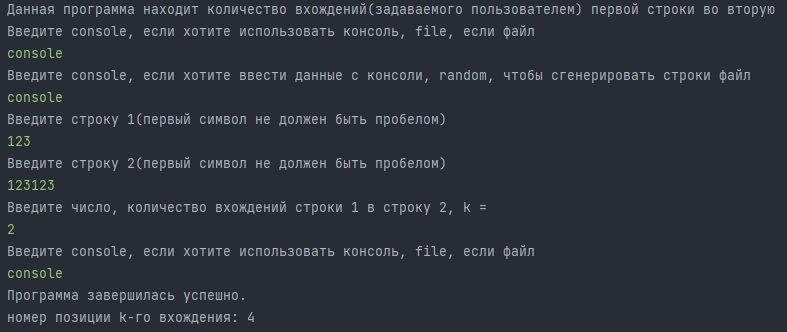
Результат на **Delphi**:



Результат на **C++**:

****

Результат на **Java**:



–