|  |  |
| --- | --- |
| Министерство образования Республики Беларусь | |
| Учреждение образования | |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ | |
| ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ | |
|  | |
|  | |
| Факультет компьютерных систем и сетей | |
| Кафедра программного обеспечения информационных технологий | |
| Дисциплина: **Название дисциплины (АББРЕВИАТУРА)** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| **ОТЧЁТ** | |
| по лабораторной работе № **X** | |
|  | |
| Тема работы: **Название темы** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
| Выполнил: | Фамилия И.О. |
|  | гр. **XXXXXX** |
|  | Вариант **X** |
|  |  |
| Проверил: | Фамилия И.О. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Минск **202X** | |

содержание

[1 Постановка задачи 3](#_Toc106355241)

[2 Методика решения задачи 4](#_Toc106355242)

[2.1 Использование текстовых файлов 4](#_Toc106355243)

[2.2 Циклический сдвиг строки 4](#_Toc106355244)

[2.3 Файлы в подпрограммах 4](#_Toc106355245)

[3 Описание алгоритмов решения задачи 5](#_Toc106355246)

[4 Структура данных 6](#_Toc106355247)

[4.1 Структура данных программы 6](#_Toc106355248)

[4.2 Структура данных алгоритма Count 6](#_Toc106355249)

[4.3 Структура данных алгоритма Shift 6](#_Toc106355250)

[4.1 Структура данных алгоритма LoadFile 7](#_Toc106355251)

[5 Схема алгоритма решения задачи по ГОСТ 19.701-90 8](#_Toc106355252)

[5.1 Схема алгоритма решения задачи 8](#_Toc106355253)

[5.2 Схема алгоритма Shift 9](#_Toc106355254)

[5.3 Схема алгоритма Count 10](#_Toc106355255)

[5.4 Схема алгоритма LoadFile 11](#_Toc106355256)

[6 Результаты расчетов 12](#_Toc106355257)

[Приложение А 13](#_Toc106355258)

[Приложение Б 16](#_Toc106355259)

# Постановка задачи

Постановка 1 (исходная)

Даны текстовый файл F, текстовый файл F1 и натуральное число К.

Файл F содержит 30 слов, каждое из которых будем называть ключевым. Сформировать файл G, который содержит строки файла F1, циклически сдвинутые так, чтобы каждое ключевое слово, входящее в строку, начиналось с К-ой позиции.

Строки, не содержащие ключевых слов, в файл G не включаются.

Строки, которые содержат N ключевых слов, записываются в файле G N раз.

Постановка 2 (реализованная)

Даны текстовый файл F, текстовый файл F1 и натуральное число К.

Файл F содержит до 100 слов, каждое из которых будем называть ключевым. Сформировать файл Res, который содержит строки файла F1, циклически сдвинутые так, чтобы каждое ключевое слово, входящее в строку, начиналось с К-ой позиции.

Строки, не содержащие ключевых слов, в файл G не включаются.

Строки, которые содержат N ключевых слов, записываются в файле G N раз.

Если K больше, чем длина строки, то об этом сообщается пользователю.

Имеется поддержка файлов с символами из отличных от латинского алфавитов.

# Методика решения задачи

## Использование текстовых файлов

Согласно постановке, необходимо использовать текстовые файлы, имеющие расширение .txt. В лабораторной работе будет иметь место:

* определение конца файла;
* чтение данных из файлов;
* запись данных в файл.

Для этого будут применяться функции Eof, ReadLn и WriteLn соответственно.

Для поддержки русского языка необходимо сохранять текстовые файлы в текстовом формате ANSI.

## Циклический сдвиг строки

Циклический сдвиг – такой тип сдвига, при котором данные не удаляются безвозвратно, а добавляются в противоположный конец строки согласно порядку «First Out, First In».

В поставленной задаче циклический сдвиг осуществляется не на X позиций, а к позиции номер X, из-за чего появляется лимит на максимальную позицию: позиция не может превышать длину строки.

## Файлы в подпрограммах

Особенностью файлового типа является то, что он не может передаваться в процедуру/функцию в качестве адреса или защищённого адреса.

Таким образом, формальный параметр файлового типа употребляется только с var. Также файловый тип может быть возвращаемым значением функции.

# Описание алгоритмов решения задачи

Таблица 1 – Описание алгоритмов решения задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п.п | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Предполагаемый тип реализации |
| 1. | Основной алгоритм | Вызывает следующие подпрограммы:  Shift, Count, LoadFile |  |  |
| 2. | Shift  (Key, Sus, K, Res) | Осуществляет циклический сдвиг строки Sus таким образом, чтобы первая буква фрагмента Key была K-тым по номеру символом в строке, и сохраняет полученную строку в Res. | K – получает адрес от фактического параметра;  Key, Sus, Res – получает значение от фактического параметра. | Процедура |
| 3. | Count  (Key, Sus, Amogus) | Подсчитывает количество фрагментов Key в строке Sus и сохраняет это значение в Amogus | Key, Sus – получает адрес от фактического параметра;  Amogus – получает значение от фактического параметра. | Процедура |
| 4. | LoadFile  (F, I, Arr, S) | Загружает данные из файла F с именем S в массив Arr с сохранением его длины в I. | S – получает защищённый адрес от фактического параметра;  F, I, Arr – получает значение от фактического параметра. | Процедура |

# Структура данных

## Структура данных программы

Таблица 2 – Структура данных

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| F | TextFile | Текстовый файл |
| Res | String | Циклически сдвинутая строка |
| I, J, K, L, M, N | Integer | Счётчики циклов |
| Amogus | Integer | Количество ключевых слов в строке |
| Data | Array[1..100] Of String | Массив строк, считанных из текстового файла |
| Keys | Array[1..100] Of String | Массив ключевых слов, считанных из текстового файла |

## Структура данных алгоритма Count

Таблица 3 – Структура данных алгоритма Count(Key, Sus, Amogus)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Key | String | Ключевое слово | Формальный |
| Sus | String | Строка, которая может содержать ноль и более ключевых слов | Формальный |
| Amogus | Integer | Количество найденных ключевых слов в строке | Формальный |

## Структура данных алгоритма Shift

Таблица 4 – Структура данных алгоритма Shift(Key, Sus, K, Res)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Key | String | Ключевое слово | Формальный |

Продолжение Таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| Sus | String | Строка, содержащая одно или более ключевых слов | Формальный |
| K | Integer | Позиция, с которой начнётся ключевое слово после циклического сдвига | Формальный |
| Res | String | Циклически сдвинутая строка | Формальный |
| W | String | Сохранённая часть строки, которая отрезается с одной стороны и добавляется с другой | Локальный |
| L | Integer | Расстояние между текущей и будущей позицией ключевого слова | Локальный |

## Структура данных алгоритма LoadFile

Таблица 5 – Структура данных алгоритма LoadFile(F, I, Arr, S)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Тип параметра |
| F | TextFile | Файл | Формальный |
| I | Integer | Счётчик цикла и, по окончании, итоговое число строк (длина массива) | Формальный |
| Arr | TArr | Массив, каждый элемент которого – строка из файла | Формальный |
| S | String | Имя файла | Формальный |

# Схема алгоритма решения задачи по ГОСТ 19.701-90

## Схема алгоритма решения задачи

|  |
| --- |
|  |
| Рисунок – Схема алгоритма решения задачи |

## Схема алгоритма Shift

|  |
| --- |
|  |
| Рисунок 2 – Схема алгоритма Shift |

## Схема алгоритма Count

|  |
| --- |
|  |
| Рисунок 3 – Схема алгоритма Count |

## Схема алгоритма LoadFile

|  |
| --- |
|  |
| Рисунок 4 – Схема алгоритма LoadFile |

# Результаты расчетов

Вследствие результатов программы мы получаем следующие результаты:

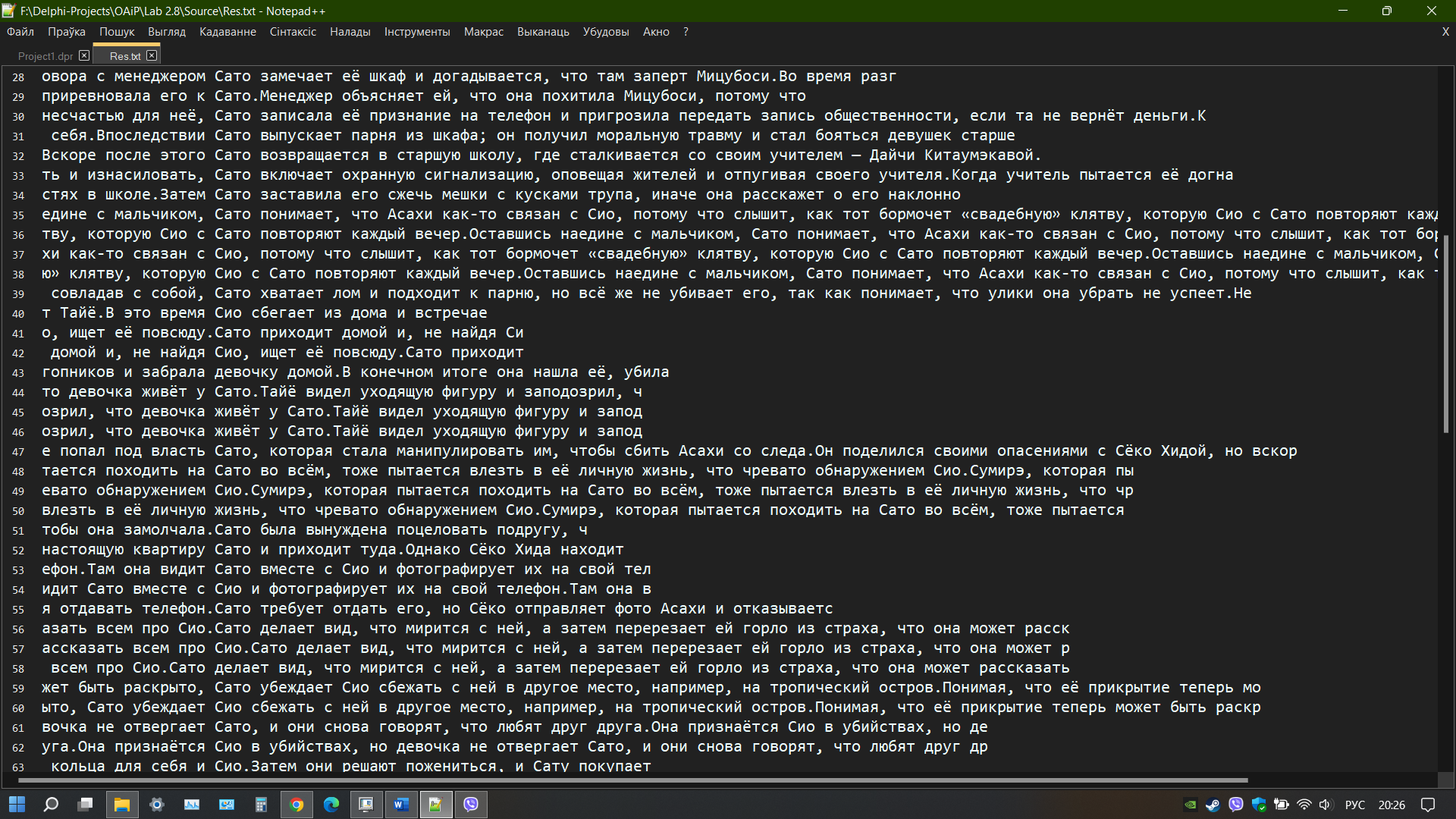


Рисунок 5 – Результаты циклического сдвига

Приложение А

(обязательное)

Исходный код программы

Program Lab8;

{Program loads key dict and does cycle shift}

{$APPTYPE CONSOLE}

Uses

SysUtils,

Windows;

Type

TArr = Array[1..100] Of String;

Var

F: TextFile;

Res: String;

I, J, K, L, M, N, Amogus: Integer;

Data, Keys: TArr;

//F - file

//I, J, K, L, M, N - counters

//Amogus - amount of positions

//Keys - array of dict

//Data - array of data

//Res - result array

//Calculates how many duplicates will be

//Key - word, Sus - line, Amogus - amount

Procedure Count(Key, Sus: String; Var Amogus: Integer);

Begin

While AnsiPos(Key, Sus) > 0 Do

Begin

Inc(Amogus);

Delete(Sus, 1, AnsiPos(Key, Sus));

End;

End;

//Does cycle shift

//Key - word, Sus - line, K - length of step, Res - result

Procedure Shift(Var Key, Sus: String; K: Integer; Var Res: String);

Var

W: String;

L: Integer;

Begin

Res:= Sus;

L:= AnsiPos(Key, Res) - K;

//Cut from start or end

If L >= 0 Then

Begin

W:= Copy(Res, 1, L);

Delete(Res, 1, L);

Res:= Res + W;

End

Else

Begin

W:= Copy(Res, Length(Res) + L + 1, Length(Res));

Delete(Res, Length(Res) + L + 1, Length(Res));

Res:= W + Res;

End;

//Prepare for the next key

L:= AnsiPos(Key, Sus) + Length(Key);

W:= Copy(Sus, 1, L);

Delete(Sus, 1, L);

Sus:= Sus + W;

End;

//Loads file data into array

//F - file, I - size, Keys - array, S - filename

Procedure LoadFile(var F: TextFile; var I: Integer; var Arr: TArr; const S: String);

Begin

AssignFile(F, S);

Reset(F);

I:= 1;

While Not Eof(F) Do

Begin

ReadLn(F, Arr[I]);

Inc(I);

End;

CloseFile(F);

End;

Begin

//Cyrillic support

SetConsoleCP(1251);

SetConsoleOutPutCP(1251);

//Load dictionary and data

LoadFile(F, L, Keys, 'F.txt');

LoadFile(F, N, Data, 'F1.txt');

Write('Enter K: ');

ReadLn(K);

//Create file with shifted data

AssignFile(F, 'Res.txt');

Rewrite(F);

For I:= 1 To N Do

For J:= 1 To L Do

Begin

//How many additions

Amogus:= 0;

Count(Keys[J], Data[I], Amogus);

//If should be added

If (Amogus > 0) Then

Begin

For M:= 1 To Amogus Do

Begin

//Prevent from border

If Length(Data[I]) >= K Then

Begin

Shift(Keys[J], Data[I], K, Res);

WriteLn(F, Res);

End

Else

WriteLn(F, 'K is too big');

End;

End;

End;

CloseFile(F);

Write('Saved to file Res.txt');

ReadLn;

End.

Приложение Б

(обязательное)

Тестовые наборы

Пусть есть файл F, в котором содержится 39 разных ключевых слов, и файл F1, в котором находится 68 разных предложений разной длины, в которых встречаются некоторые из 39-ти ключевых слов. Пусть K = 5.

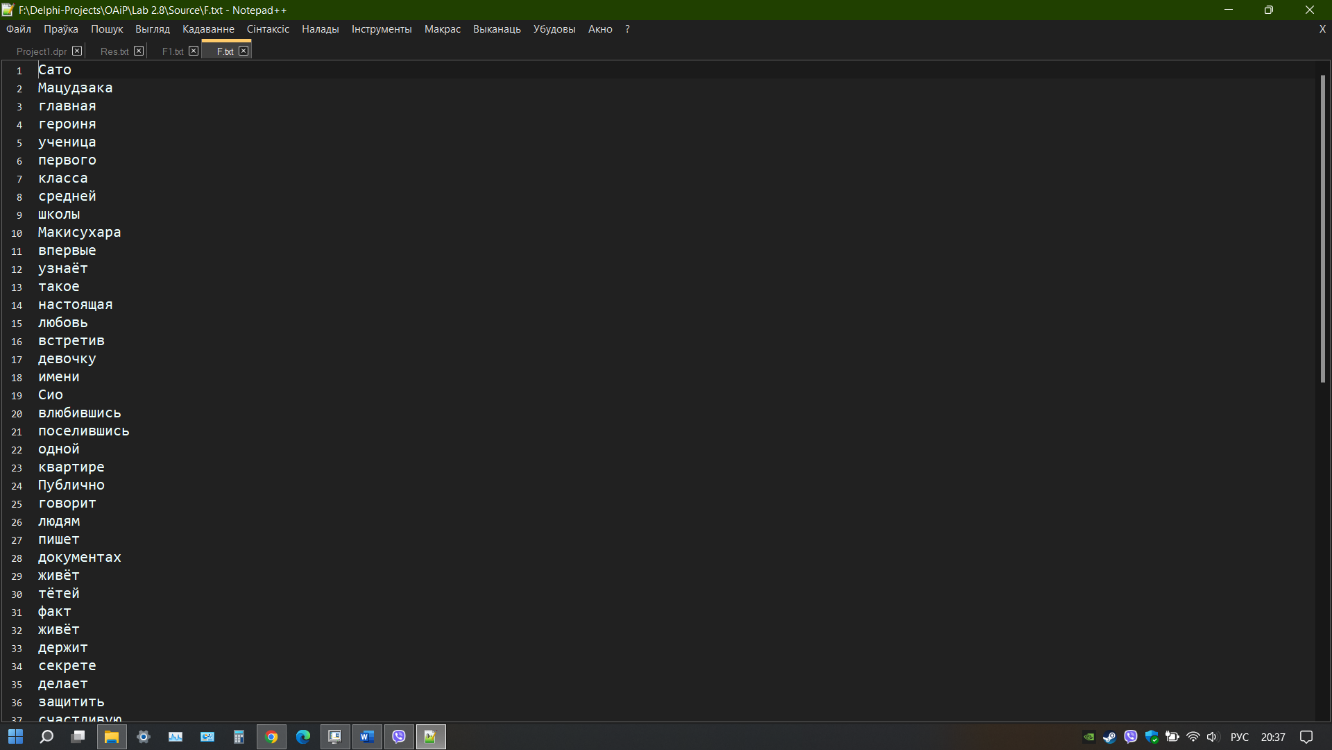


Рисунок 6 – часть содержимого файла F

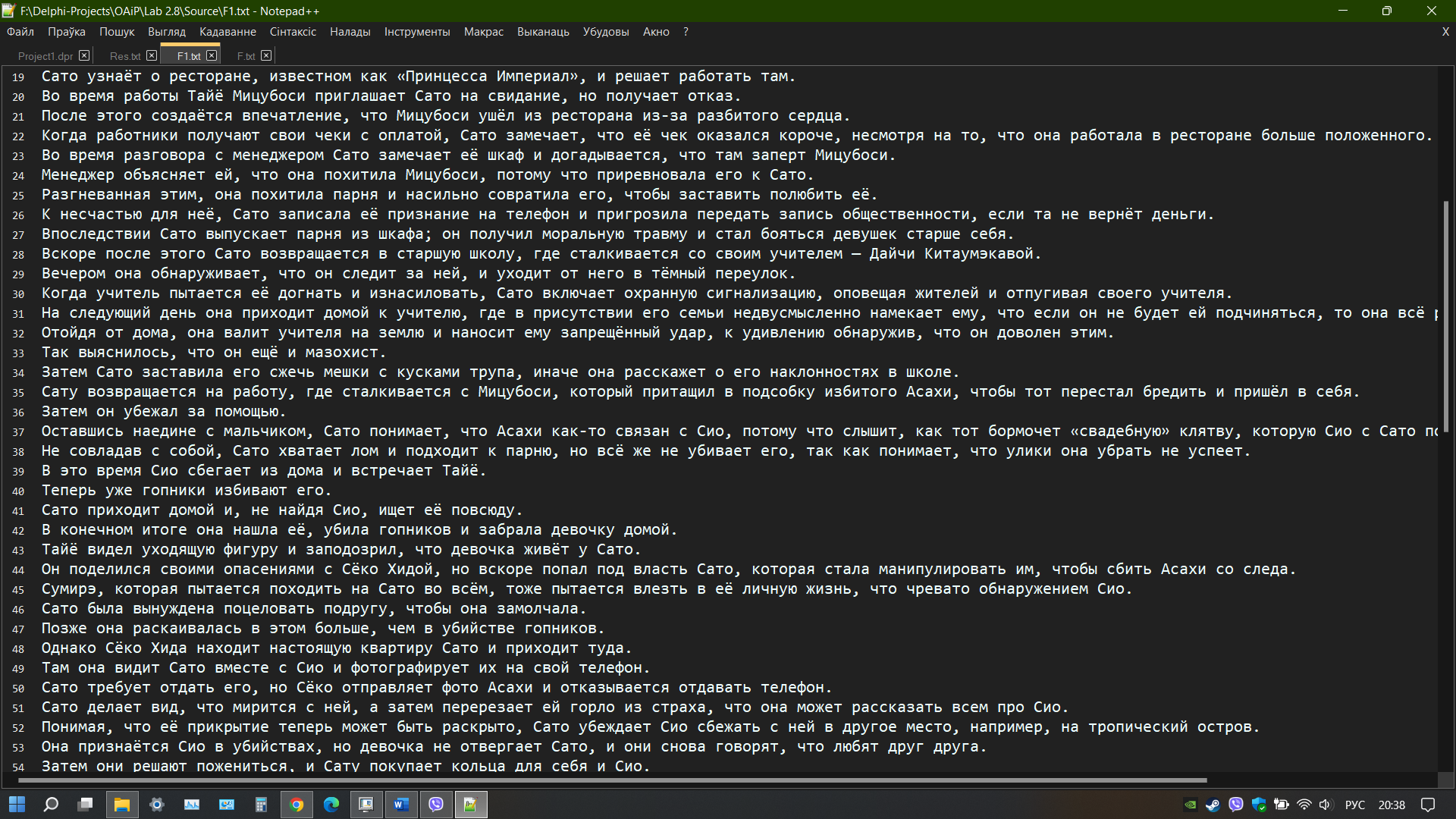


Рисунок 7 – часть содержимого файла F1

В результате видим:

* строки без ключевых слов отсутствуют;
* строки с одним ключевым словом упомянуты один раз и в них первый символ ключевого слова является пятым по счёту в строке;
* строки с двумя и более ключевыми словами встречаются несколько раз, причём каждый раз пятым по счёту является символ нового ключевого слова, встреченного в строке.

Также очевидно, что циклический сдвиг осуществлён успешно, так как предложения осмысленны, их части не утрачены.

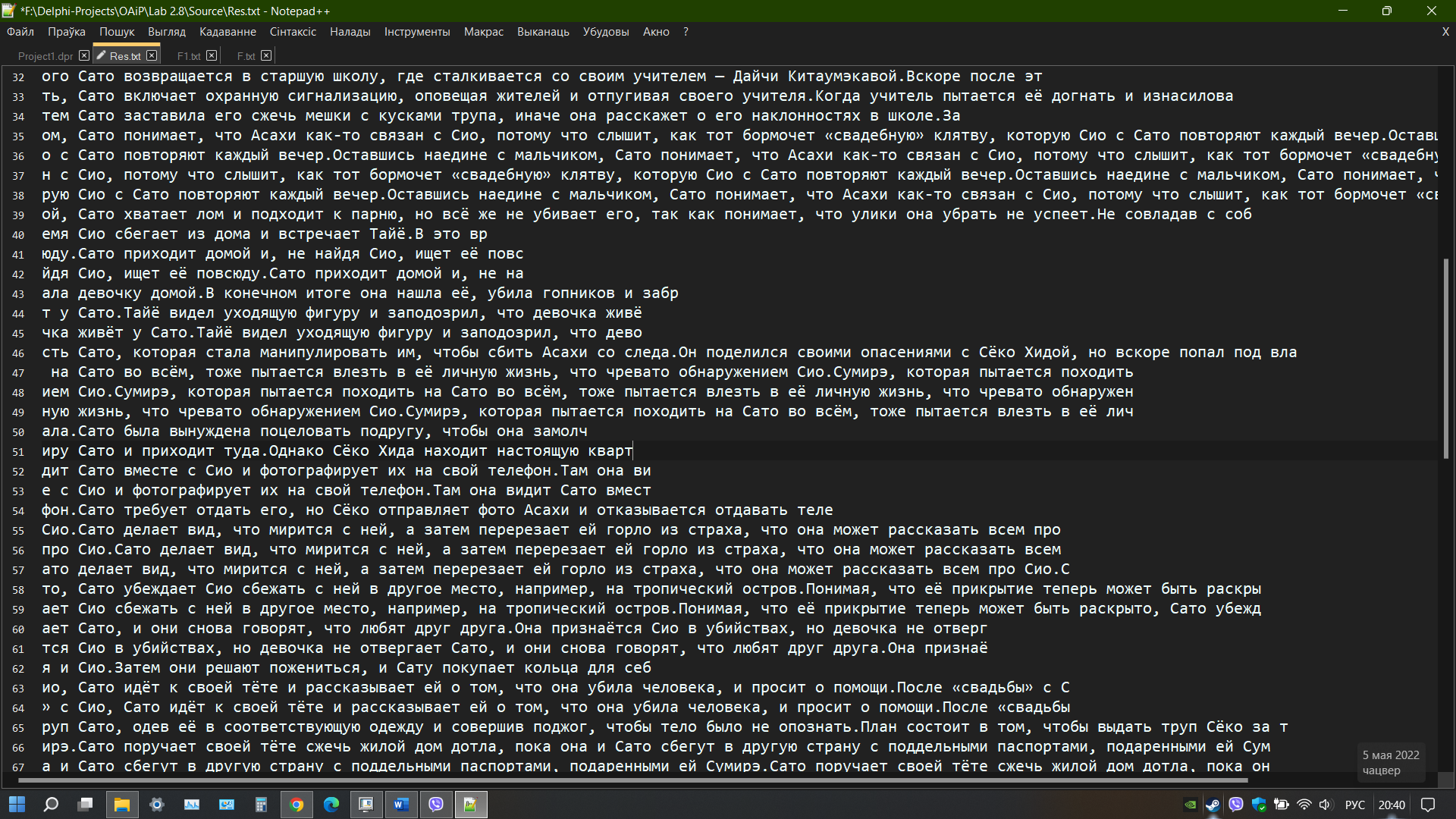


Рисунок 8 – результаты

Если ввести значение, заведомо большее длины строки, то циклический сдвиг не будет совершён, поскольку ни при каком случае невозможно сделать так, чтобы у слова из строки была позиция большая длины строки.

Пусть N = 1999.

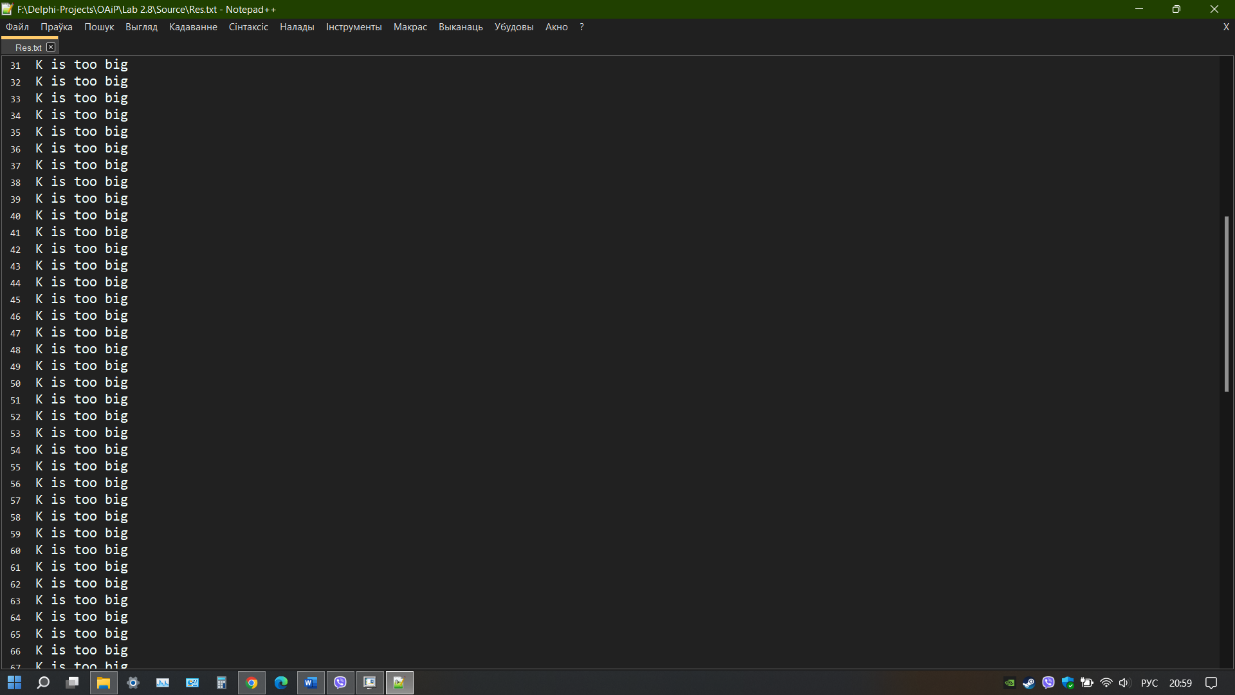


Рисунок 9 – позиция за гранью строки