

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

Курсовая работа
по дисциплине «Программирование»
ТЕМА: «Обработка текстовой информации»

Студент гр. 3311

Сапронов К.Д.

Преподаватель

Хахаев И. А.

Санкт-Петербург

2023

Цель работы.

Законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование)

Задание (вариант 73)

Задан текст, содержащий произвольное количество строк, в которых отдельные слова могут разделяться одним или несколькими пробелами и знаками пунктуации (перенос слов с одной строки на другую не используется). Сформировать новый текст, который является результатом следующего преобразования исходного текста: удалить строки, содержащие слова, принадлежащие заданному набору слов.

Постановка задачи и описание решения

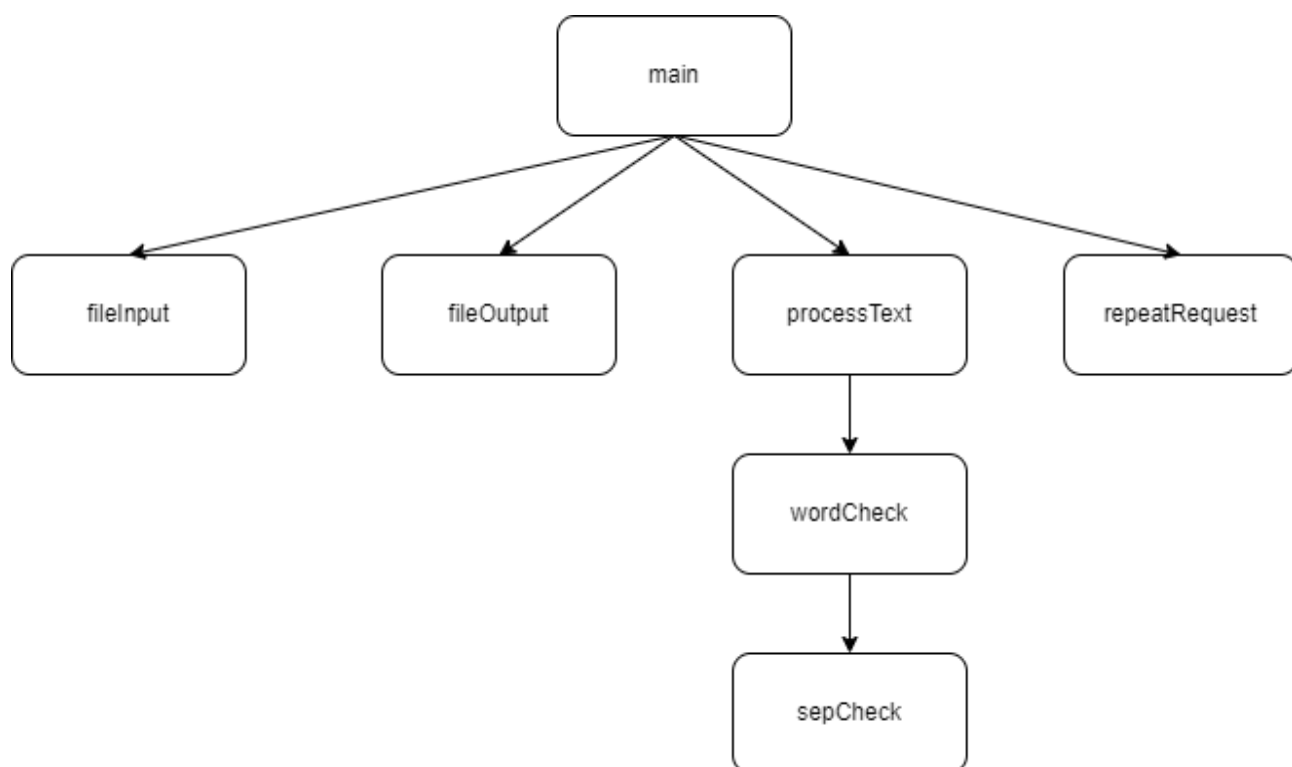
Для выполнения данной задачи необходимо написать программу, которая будет принимать от пользователя многострочный текст (с помощью клавиатуры или готового файла), а также слова, строки с которыми будут исключены из текста. После этого текст построчно проверяется на наличие заданных слов, и если они не обнаружены, то строка записывается в новый файл. В конце содержимое нового файла выводится.

1. Пользователь вводит с клавиатуры количество строк в исходном файле и количество задаваемых слов, после чего вводит эти слова. После этого пользователю дается выбор, вводить текст с клавиатуры или из другого файла. Если пользователь выбирает ввод с клавиатуры, то он вводит текст, который заносится в файл для исходного текста. Если же он выбирает ввод из файла, то он вводит имя файла, который будет использоваться как файл с исходным текстом.

2. Внутри файла текст построчно проверяется на удовлетворение условию. Для этого в каждой строке каждое слово сравнивается с каждым словом, заданным изначально, и, если в строке не обнаруживается ни одного совпадения, то строка заносится в новый созданный файл.

3. Из нового файла на экран построчно выводится его содержимое

Структура вызовов функций



Описание переменных

SIZE - константа типа int. Предназначена для выделения памяти

main()

№	Имя переменной	Тип	Назначение
1	numWords	int	Количество задаваемых слов
2	i	int	Переменная в цикле
3	numOfInputStr	int	Количество строк в исходном тексте
4	numOfOutputStr	int	Количество строк в новом тексте
5	seps	char	Строка разделителей
6	words	char	Заданные слова
7	answer	char	Ответ пользователя
8	ownFilename	char	Имя собственного файла
9	mainFilename	const char	Имя исходного файла
10	newFilename	const char	Имя нового файла

fileInput() - ввод текста в файл с клавиатуры

№	Имя переменной	Тип	Назначение
1	mainFilename	char	Имя файла
2	numOfStrings	int	Количество строк в тексте
3	seps	char	Строка разделителей
4	string	char	Введенная строки
5	i	int	Переменная в цикле
6	len	int	Длина введенной строки
7	file	FILE	Файл для записи текста

fileOutput() - вывод текста из файла

№	Имя переменной	Тип	Назначение
1	filename	char	Имя файла
2	numOfStrings	int	Количество строк в тексте
3	seps	char	Строка разделителей
4	string	char	Выводимая строка
5	i	int	Переменная в цикле
7	file	FILE	Файл, из которого выводится текст

sepCheck() - проверка на то, является ли символ разделителем

№	Имя переменной	Тип	Назначение
1	ch	char	Проверяемый символ
2	seps	char	Строка разделителей

wordCheck() - проверка строки на наличие заданных слов

№	Имя переменной	Тип	Назначение
1	string	char	Проверяемая строка
2	seps	char	Строка разделителей
3	words	char	Заданные слова
4	numWords	int	Количество заданных слов
5	res	int	Значение, возвращаемое функцией
6	i	int	Переменная в цикле
7	line	char	Строка для записи проверяемого слова

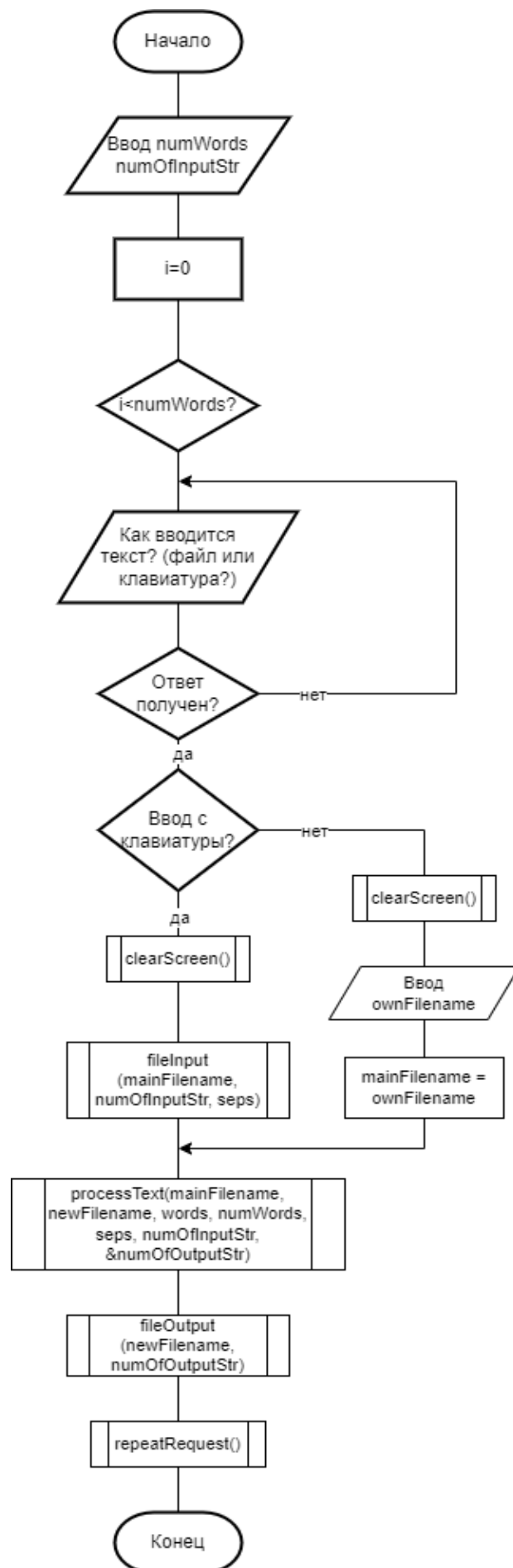
processText() - обработка текста

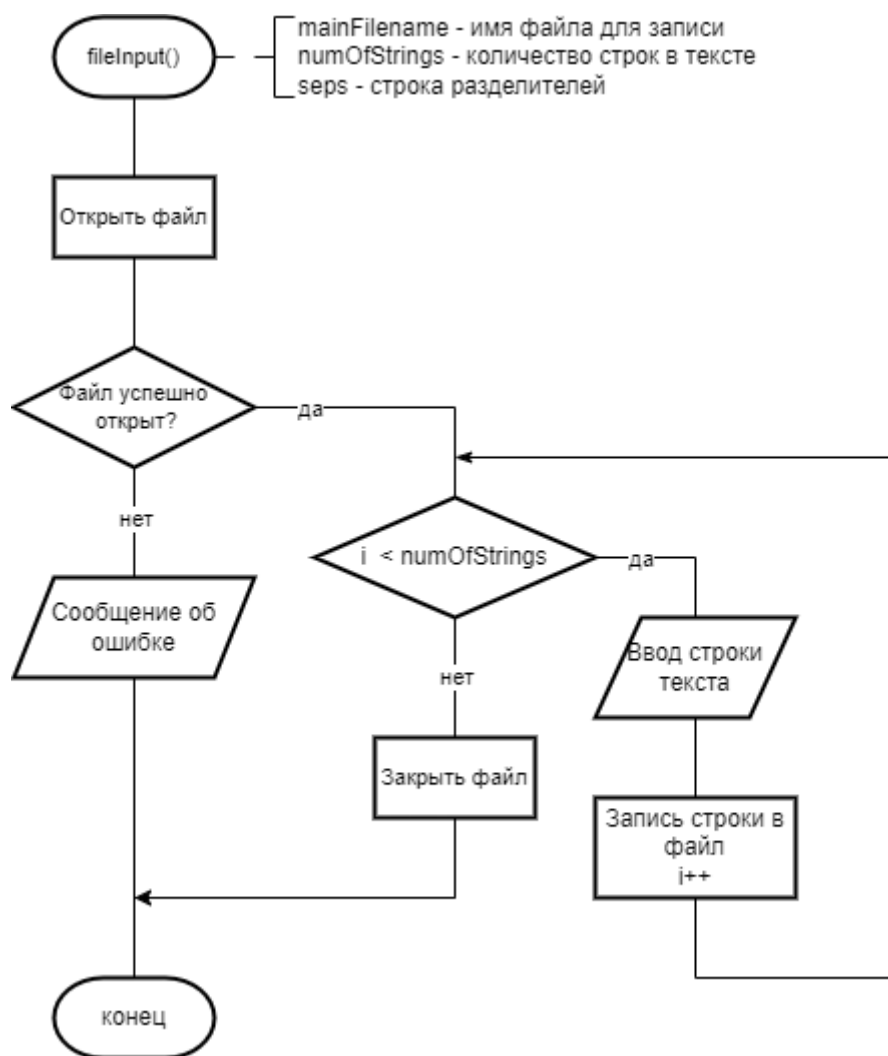
№	Имя переменной	Тип	Назначение
1	mainFilename	const char	Имя исходного файла
2	newFilename	const char	Имя нового файла
3	seps	char	Строка разделителей
4	words	char	Заданные слова
5	numWords	int	Количество заданных слов
6	numOfInputStr	int	Количество строк в исходном тексте
7	numOfOutputStr	int	Количество строк в новом тексте
8	string	char	Обрабатываемая строка
9	k	int	Номер считываемой строки
10	len	int	Длина считываемой строки
11	mainFile	FILE	Файл с исходным текстом
12	newFile	FILE	Файл для обработанного текста

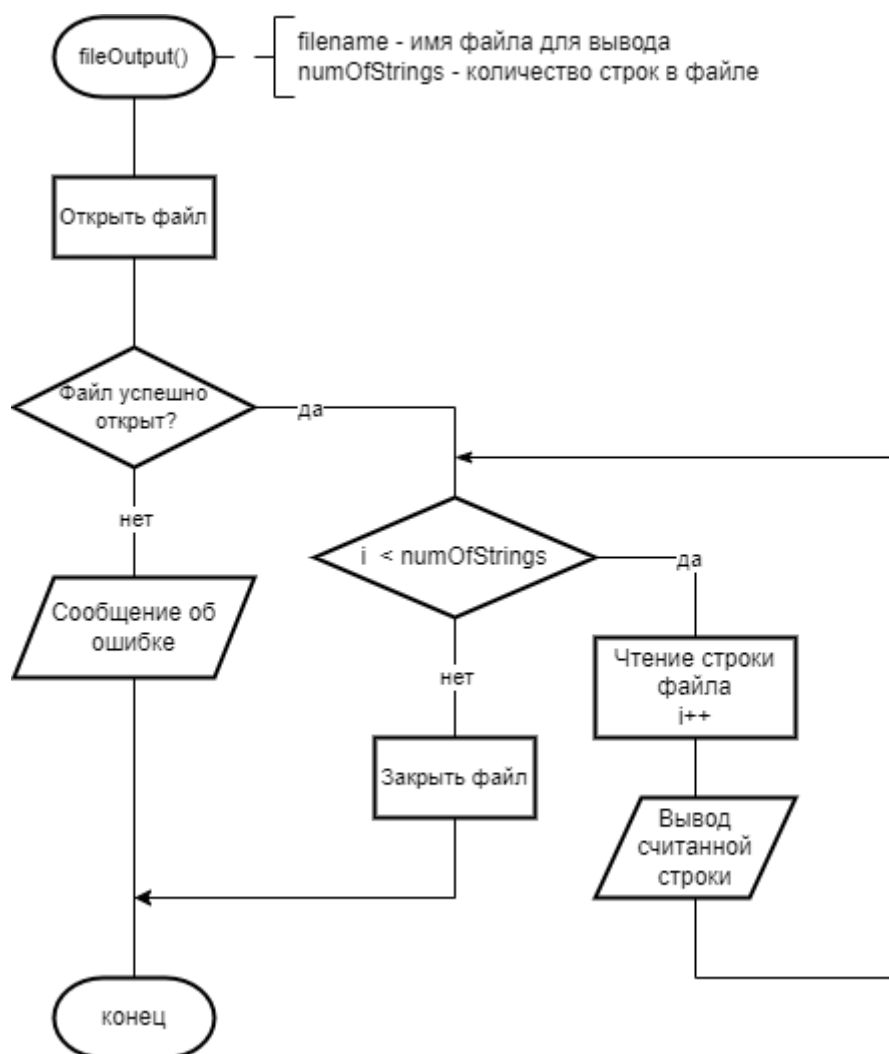
repeatRequest() - повторение программы

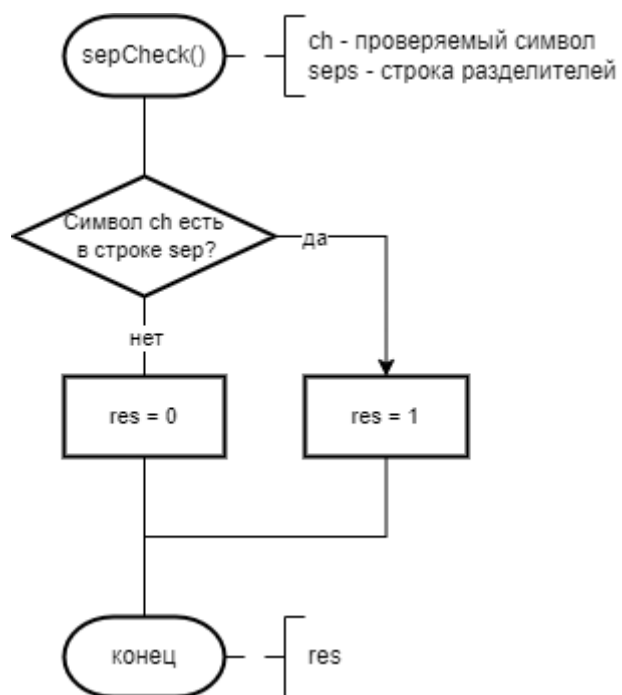
№	Имя переменной	Тип	Назначение
1	answer	char	Ответ пользователя

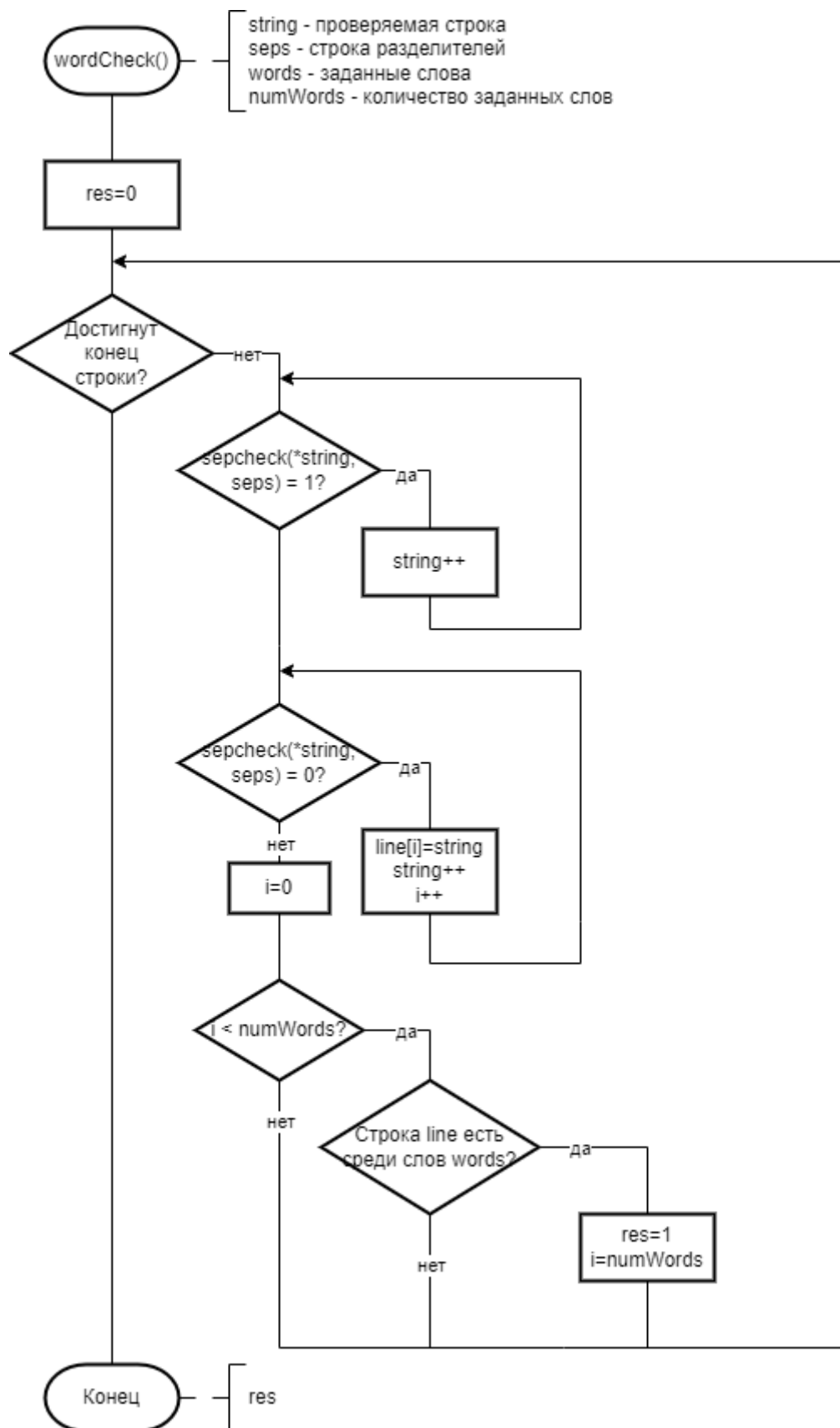
Схема алгоритма

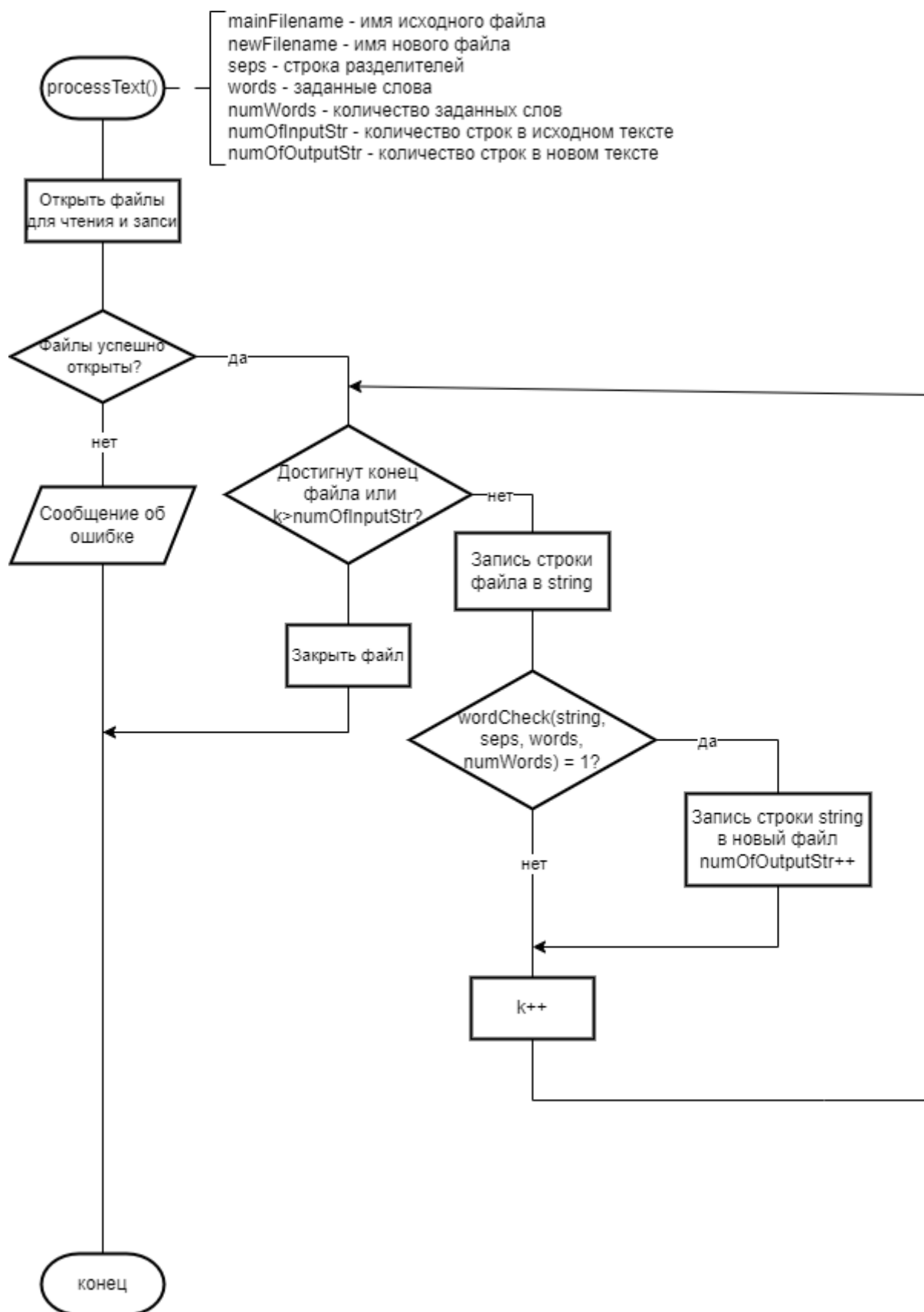


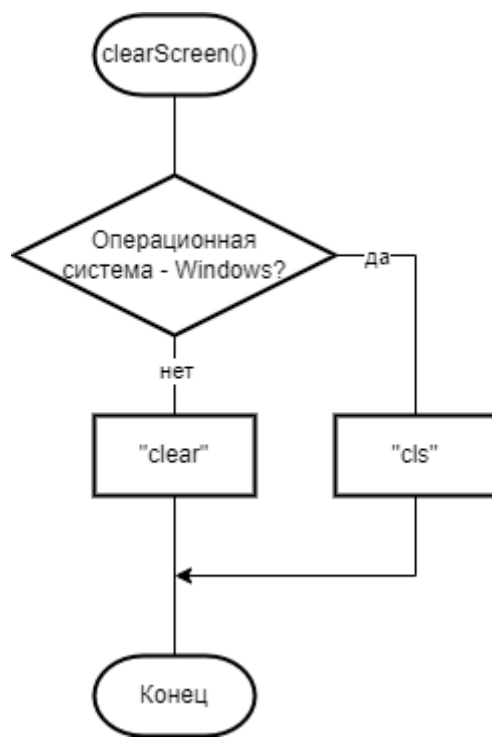


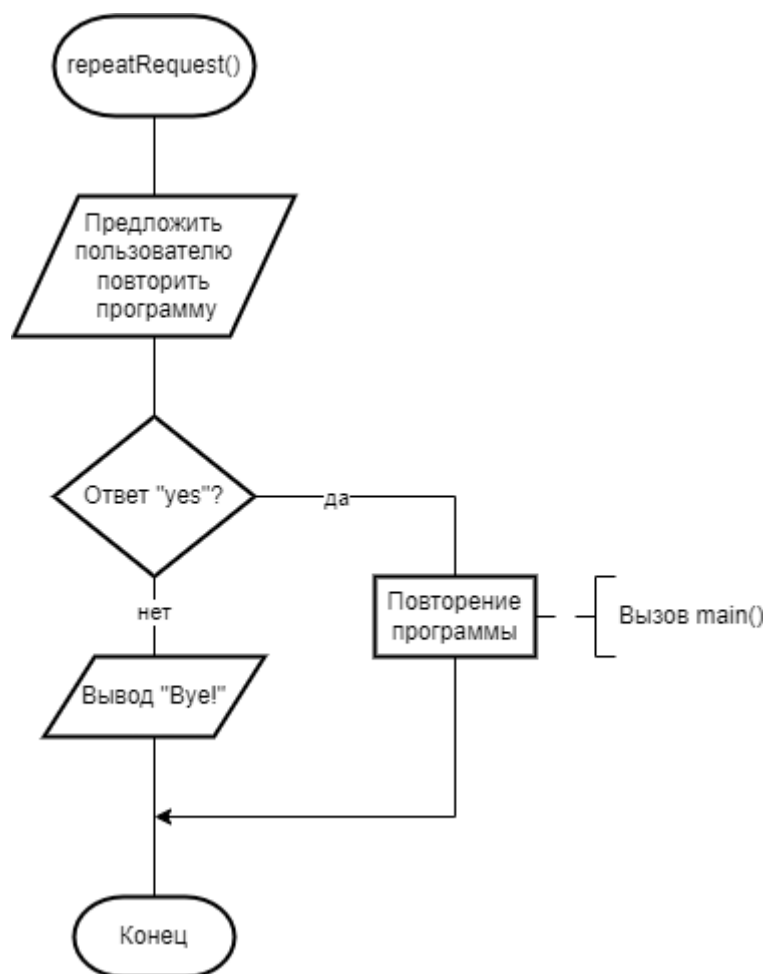












Контрольные примеры

Пример 1.

Ввод:

Количество строк: 4

Количество слов: 2

Слова: 1, 34

Ввод: keyboard

Текст:

1 1 1 1 1 1 1

1234567890

134 341

1.34

Повтор программы: no

Вывод:

1234567890

134 341

Пример 2.

Ввод:

Количество строк: 3

Количество слов: 2

Слова: a, аaaa

Ввод: keyboard

Текст:

a aa aaa

aaaaaaaaaaaaaaaa

ab aaaaab aa aaa aaaaa

Повтор программы: yes

Количество строк: 6

Количество слов: 2

Слова: is, day

Ввод: file

Имя файла: new_year.txt

Текст:

It is December the thirty-first.

Something is about to burst.

The thirty-first of December is the day

When we have a great holiday.

The New Year usually comes at midnight

And brings us presents so sweet and bright.

Повтор программы: no

Вывод:

1. aaaaaaaaaaaaaaaa

ab aaaab aa aaa aaaaa

2. When we have a great holiday.

The New Year usually comes at midnight

And brings us presents so sweet and bright.

Текст программы

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define SIZE 1000

/* @brief Reads text from the file or keyboard and writes it into a file.
   @param mainFilename Name of the file to write to.
   @param numOfStrings Number of strings to be read.
   @param seps String containing separator characters.
   @return Void.
*/
void fileInput(const char *mainFilename, int numOfStrings, const char *seps);

/* @brief Reads text from a file and prints it to the console.
   @param filename Name of the file to read from.
   @param numOfStrings Number of strings to be read.
   @return Void.
*/
void fileOutput(const char *filename, int numOfStrings);

/* @brief Checks if a character is a separator.
   @param ch Character to check.
   @param seps String containing separator characters.
   @return 1 if character is a separator else 0.
*/
int sepCheck(char ch, const char *seps);

/* @brief Checks if any word from the list exists in the string.
```

@param string String to check.
@param seps String containing separator characters.
@param words Array of words to check against.
@param numWords Number of words in the array.
@return 1 if any word is found else 0.

*/

```
int wordCheck(const char *string, const char *seps, char words[][SIZE], int numWords);
```

/* @brief Processes the text from the main file, excludes lines containing specified words, and writes the result to a new file.

@param mainFilename Name of the input file.
@param newFilename Name of the output file.
@param words Array of words to check against.
@param numWords Number of words in the array.
@param seps String containing separator characters.
@param numOfInputStr Number of input strings.
@param numOfOutputStr Pointer to the number of output strings.
@return Void.

*/

```
void processText(const char *mainFilename, const char *newFilename, char words[][SIZE], int numWords, const char *seps, int numOfInputStr, int *numOfOutputStr);
```

/* @brief Clears the console screen based on the operating system.

@return Void.

*/

```
void clearScreen();
```

/* @brief Prompts the user to repeat the program or exit.

@return Void.

```

*/
void repeatRequest();

int main() {
    int numWords;
    int i;
    int numOfInputStr = 0;
    int numOfOutputStr = 0;
    char seps[10] = ".,!?:;—";
    char words[SIZE][SIZE];
    char answer[SIZE] = {'\0'};
    char ownFilename[SIZE] = {'\0'};
    const char *mainFilename = "text.txt";
    const char *newFilename = "new_text.txt";

    /* User input requests */
    printf("Enter number of strings in your text: ");
    scanf("%d", &numOfInputStr);
    printf("Enter the number of words: ");
    scanf("%d", &numWords);
    getchar();
    for (i = 0; i < numWords; i++) {
        printf("Enter the word #%d: ", i + 1);
        scanf("%s", words[i]);
        getchar();
    }
    /* Loop until valid input received */
    while (strcmp(answer, "file") != 0 && strcmp(answer, "keyboard") != 0) {
        printf("How would you like to enter your text? (keyboard/file): ");
        scanf("%s", answer);
    }
}

```

```

    if (strcmp(answer, "keyboard") == 0) {
        /* Input with keyboard */
        clearScreen();
        fileInput(mainFilename, numOfInputStr, seps);
    }
    else if (strcmp(answer, "file") == 0) {
        /* Use own file as a main file */
        clearScreen();
        printf("Enter name of your file(name.txt): ");
        scanf("%s", ownFilename);
        getchar();
        mainFilename = ownFilename;
    }
}

/* Process text and output result*/
    processText(mainFilename, newFilename, words, numWords, seps,
numOfInputStr, &numOfOutputStr);
    printf("\nYour new text:\n");
    fileOutput(newFilename, numOfOutputStr);
    repeatRequest();
    return 0;
}

```

```

void fileInput(const char *mainFilename, int numOfStrings, const char *seps) {
    int i = 0, len;
    char string[SIZE];
    FILE *file = fopen(mainFilename, "w");
    /* Check if file opened successfully */
    if (file != NULL) {
        printf("Enter the text:\n");

```

```

    getchar();
    /* Loop until all strings entered */
    while (i < numOfStrings) {
        fgets(string, SIZE, stdin);
        len = strlen(string);
        if (string[len - 1] == '\n') {
            string[len - 1] = '\0';
        }
        fprintf(file, "%s\n", string);
        i++;
    }
    fclose(file);
}

else perror("Error opening file\n");
}

void fileOutput(const char *filename, int numOfStrings) {
    int i = 0;
    char string[SIZE];
    FILE *file = fopen(filename, "r");
    /* Check if file opened successfully */
    if (file != NULL) {
        /* Loop until all strings read */
        while (i < numOfStrings) {
            if (fgets(string, SIZE, file) != NULL){
                printf("%s", string);
                i++;
            }
            else perror("Error reading file\n");
        }
    }
}

```

```

        fclose(file);
    }
    else perror("Error opening file\n");
}

```

```

int sepCheck(char ch, const char *seps) {
    return strchr(seps, ch) != NULL;
}

```

```

int wordCheck(const char *string, const char *seps, char words[][SIZE], int
numWords) {
    int res = 0;
    int i;
    char line[SIZE];
    /* Iterate over the string until the end is reached */
    while (*string) {
        i = 0;
        /* Skip any separator characters at the start of the word */
        while (*string && sepCheck(*string, seps)) {
            string++;
        }
        /* Copy the word into line */
        while (*string && !sepCheck(*string, seps)) {
            line[i] = *string;
            string++;
            i++;
        }
        line[i] = '\0';
        /* Check if the word is in the list of words */

```

```

    for (i = 0; i < numWords; i++) {
        if (strcmp(line, words[i]) == 0) {
            res = 1;
            i = numWords;
        }
    }
}
return res;
}

```

```

void processText(const char *mainFilename, const char *newFilename, char
words[][SIZE], int numWords, const char *seps, int numOfInputStr, int *numOfOutputStr)
{
    char string[SIZE];
    int k = 0;
    int len;
    FILE *mainFile = fopen(mainFilename, "r");
    FILE *newFile = fopen(newFilename, "w");
    /* Check if files opened successfully */
    if (mainFile != NULL && newFile != NULL) {
        /* Read lines from the input file until the end is reached or the maximum number
of input strings is reached */
        while (fgets(string, SIZE, mainFile) != NULL && k < numOfInputStr) {
            len = strlen(string);
            if (string[len - 1] == '\n') {
                string[len - 1] = '\0';
            }
            /* If the current line does not contain any words from list, write the line to the
output file */
            if (!wordCheck(string, seps, words, numWords)) {

```

```

        fprintf(newFile, "%s\n", string);
        (*numOfOutputStr)++;
    }
    k++;
}
}
else perror("Error opening file\n");
fclose(mainFile);
fclose(newFile);
}

```

```

void clearScreen() {
    #ifdef _WIN64
        system("cls");
    #else
        system("clear");
    #endif
}

```

```

void repeatRequest() {
    char answer[SIZE];
    printf("\nWould you like to repeat the program? (yes/no): ");
    scanf("%s", answer);
    getchar();
    /* If the user answers 'yes', clear the screen and call the main function again */
    if (strcmp(answer, "yes") == 0) {
        clearScreen();
        main();
    }
    else printf("\nBye!\n");
}

```


}

Примеры выполнения программы

Пример 1.

```
Enter number of strings in your text: 4
Enter the number of words: 2
Enter the word #1: 1
Enter the word #2: 34
How would you like to enter your text? (keyboard/file): keyboard|
```

```
Enter the text:
```

```
1 1 1 1 1 1 1
```

```
1234567890
```

```
134 341
```

```
1.34
```

```
Your new text:
```

```
1234567890
```

```
134 341
```

```
Would you like to repeat the program? (yes/no): no
```

```
Bye!
```

```
Process returned 0 (0x0)   execution time : 52.581 s
```

```
Press any key to continue.
```

```
|
```

Пример 2.

```
Enter number of strings in your text: 3
Enter the number of words: 2
Enter the word #1: a
Enter the word #2: aaaa
How would you like to enter your text? (keyboard/file): keyboard|
```

```
Enter the text:
a aa aaa
aaaaaaaaaaaaaaaaa
ab aaaab aa aaa aaaaa

Your new text:
aaaaaaaaaaaaaaaaa
ab aaaab aa aaa aaaaa

Would you like to repeat the program? (yes/no): yes|
```

```
Enter number of strings in your text: 6
Enter the number of words: 2
Enter the word #1: is
Enter the word #2: day
How would you like to enter your text? (keyboard/file): file|
```

```
Enter name of your file(name.txt): new_year.txt
```

```
Your new text:
When we have a great holiday.
The New Year usually comes at midnight
And brings us presents so sweet and bright.
```

```
Would you like to repeat the program? (yes/no): no
```

```
Bye!
```

```
Process returned 0 (0x0)    execution time : 66.227 s
Press any key to continue.
|
```

Заключение:

Заголовочные файлы:

<stdio.h>:

fclose & fopen (Закрытие и открытие файлов)

fgets (Ввод строки)

fprintf (Печать строки в файл)

ferror (Запись сообщения об ошибке)

getchar (Очистка буфера)

scanf (Ввод количества строк)

printf (Вывод информации на экран)

<string.h>:

strcmp (Лексикографическое сравнение строк (возвращает 0, если строки одинаковые))

strlen (Для хранения длины строки)

strchr (Для проверки является ли символ разделителем)

<stdlib.h>:

system (Для функции clearScreen; обращение к системе для очистки командной строки)

Выводы:

В результате выполнения работы изучены методы работы с текстом. Получены практические навыки при программировании на языке Си.