Министерство образования и науки РФ
Государственное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

ВВОД-ВЫВОД В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ С

Лабораторная работа № 7

по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601 _____ Белым А.А.
(подпись)

Проверил: _____ Сулимова В.В.
(подпись)

Тула 2011

Цель работы

Целью работы является ознакомление с функциями ввода-вывода, изучение функций языка С, необходимых для работы с файлами. В данной работе реализовать программу, производящую обработку данных, содержащихся в файле. Результаты программы также записываются в файл.

Задание

Определить, какая буква чаще всего встречается в заданном тексте.

Теоретическая справка

Схема алгоритма

На рисунке 1 представлена схема алгоритма ввода данных, поиска самых частых букв и их вывода.

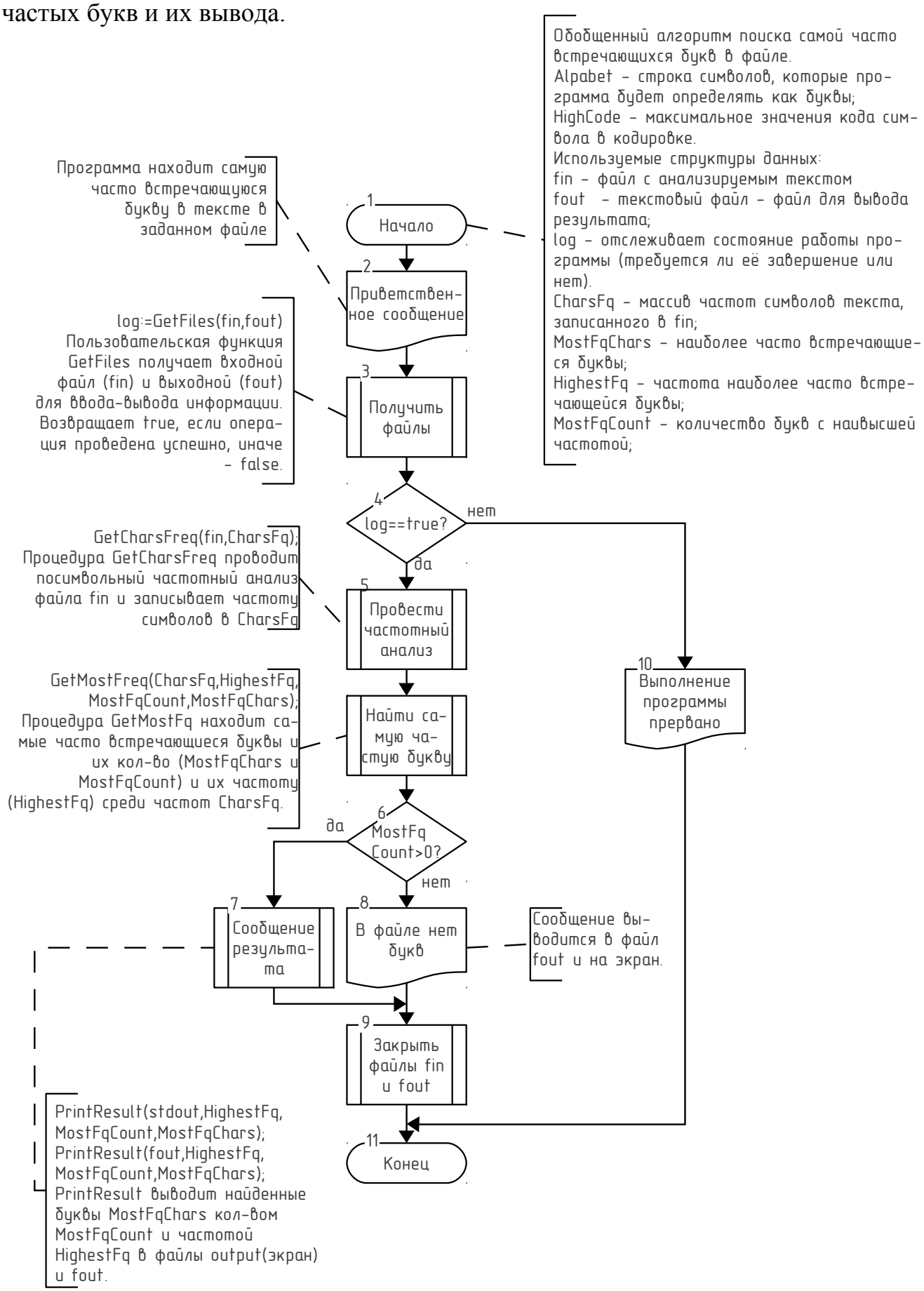

Рисунок 1 — Блок-схема обобщенного алгоритма поиска самой часто встречающейся буквы

На рисунке 2 представлена схема алгоритма получения необходимых для работы файлов.

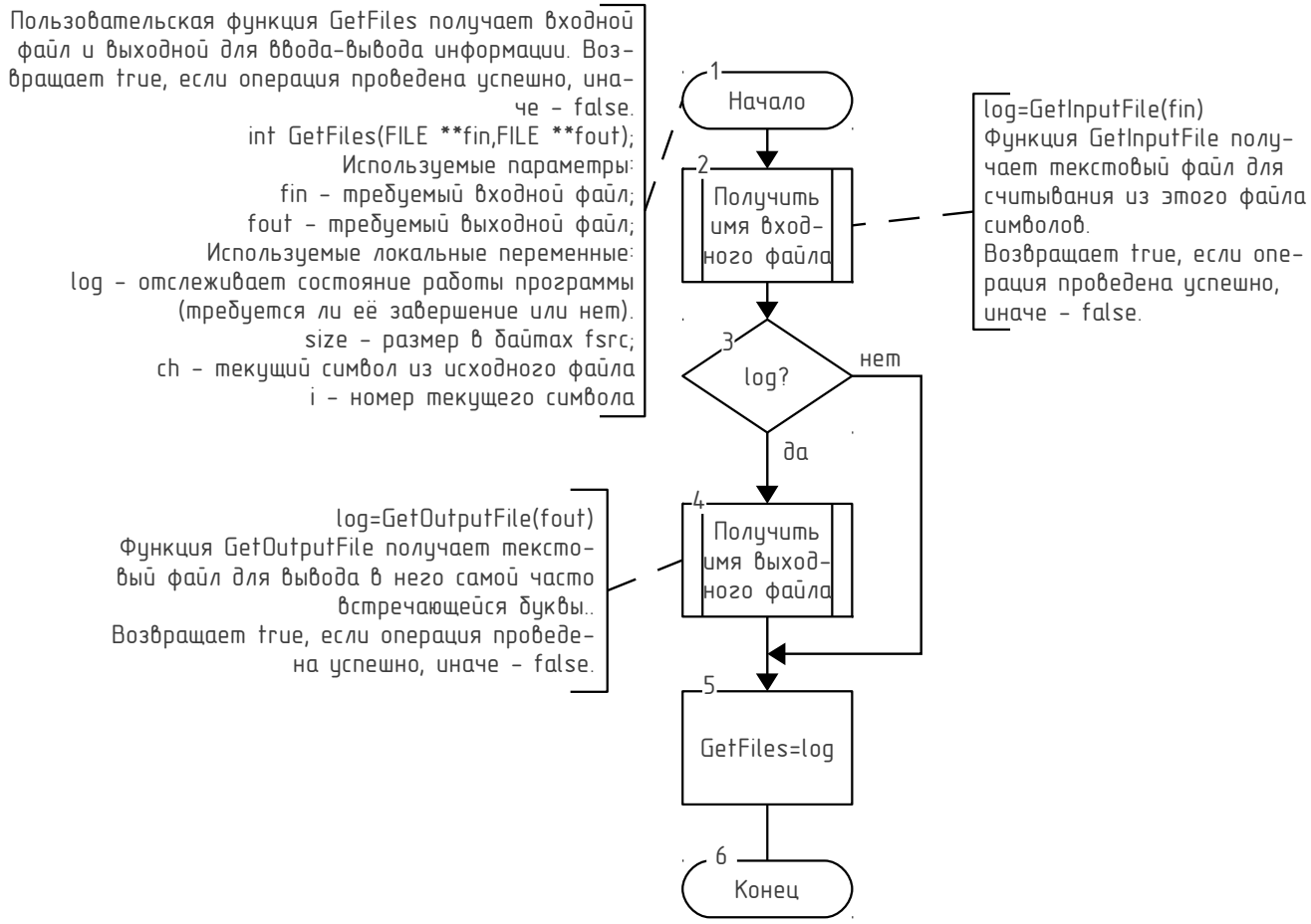

Рисунок 2 — Блок-схема алгоритма получения необходимых файлов

На рисунке 3 представлена схема алгоритма получения входного файла.

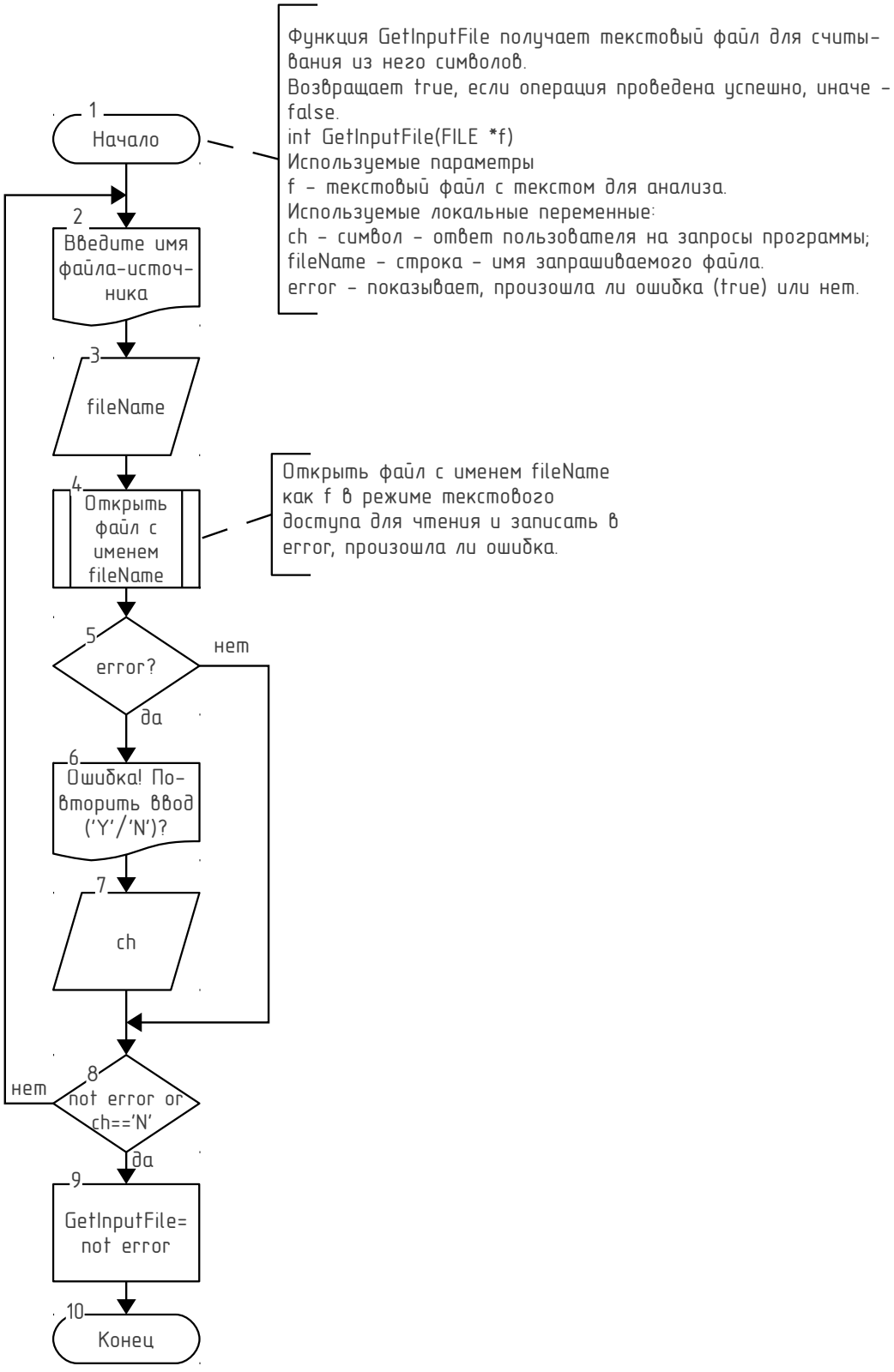

Рисунок 3 — Блок-схема алгоритма получения имени входного файла

На рисунке 4 представлена схема алгоритма получения выходного файла.

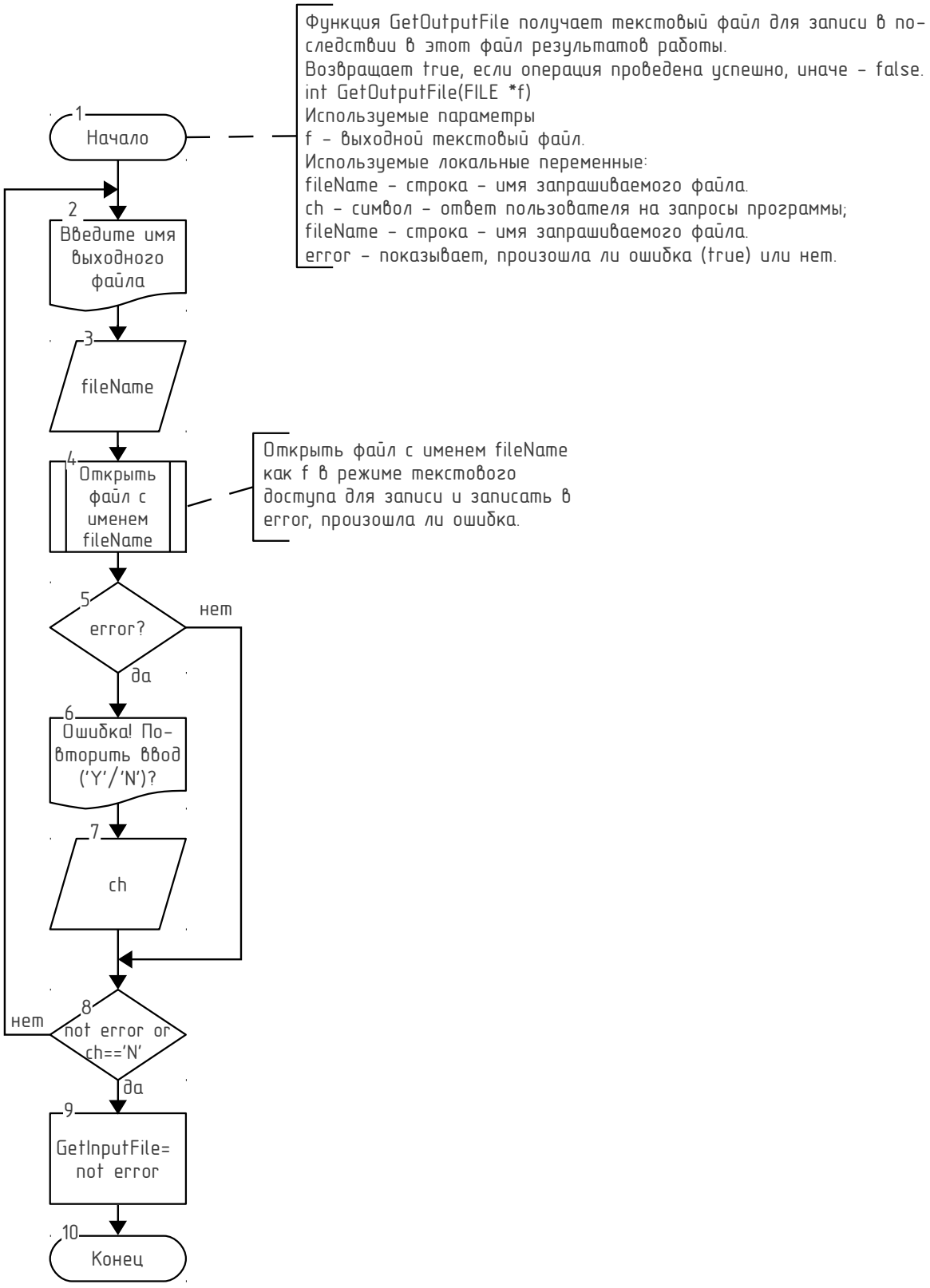

Рисунок 4 — Блок-схема алгоритма получения имени выходного файла

На рисунке 5 представлена схема посимвольного частотного анализа исходного текста с использованием кодов символов из входного файла.

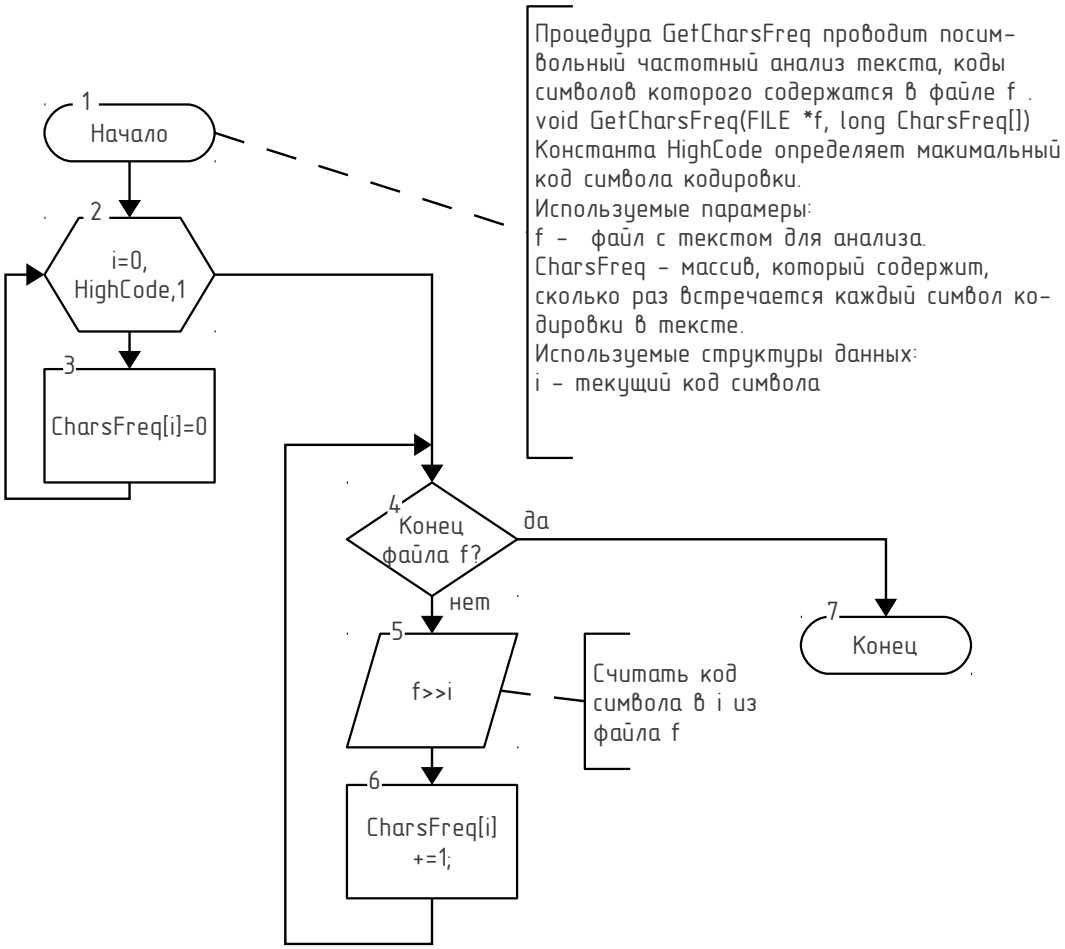

Рисунок 5 — Блок-схема алгоритма посимвольного частотного анализа

На рисунке 6 представлена схема алгоритма поиска самых частых букв при помощи данных частотного анализа.

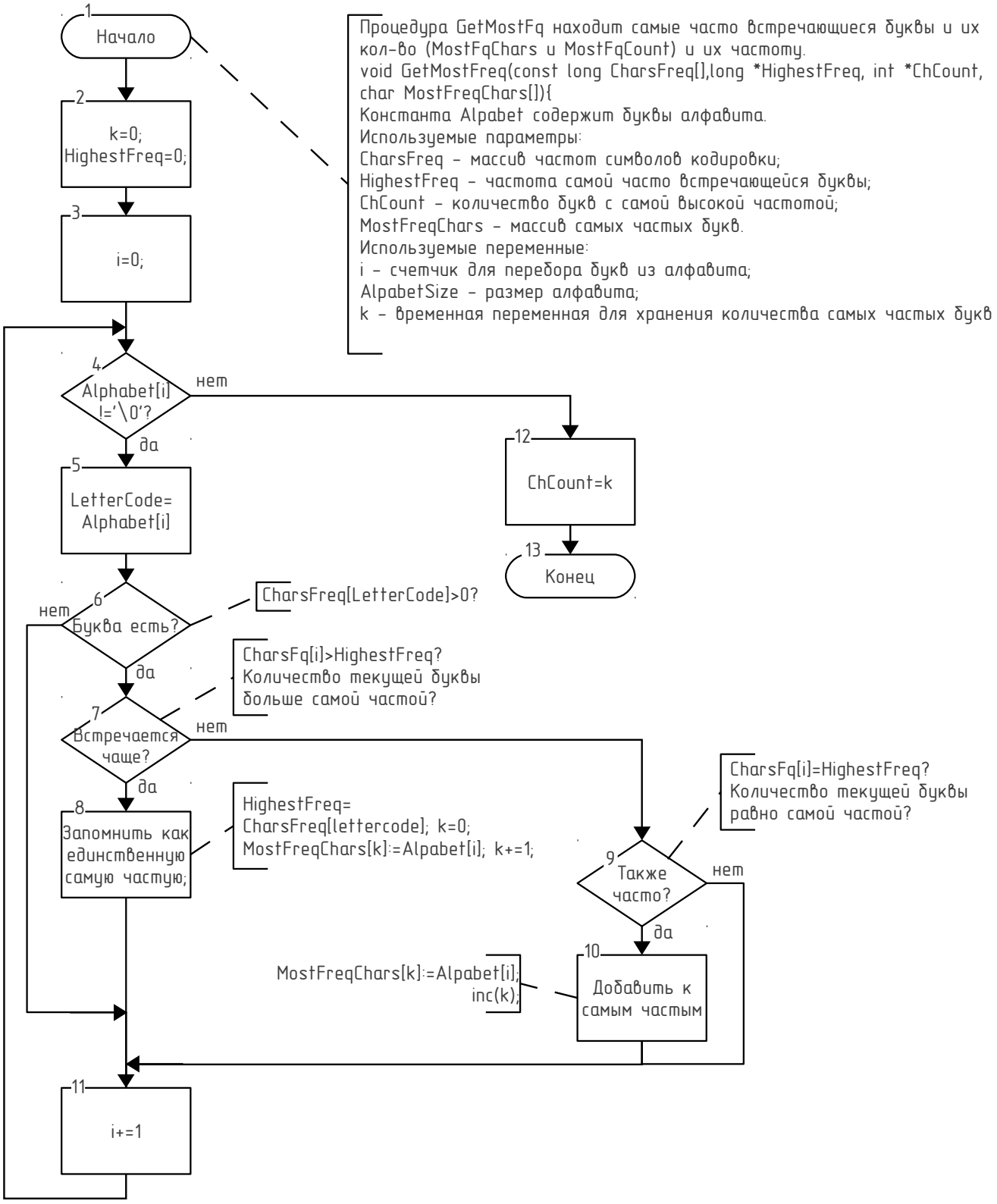

Рисунок 6 — Блок-схема алгоритма поиска самых часто встречающихся букв

На рисунке 7 представлена схема алгоритма вывода в файл самых часто встречающихся букв исходного текста.

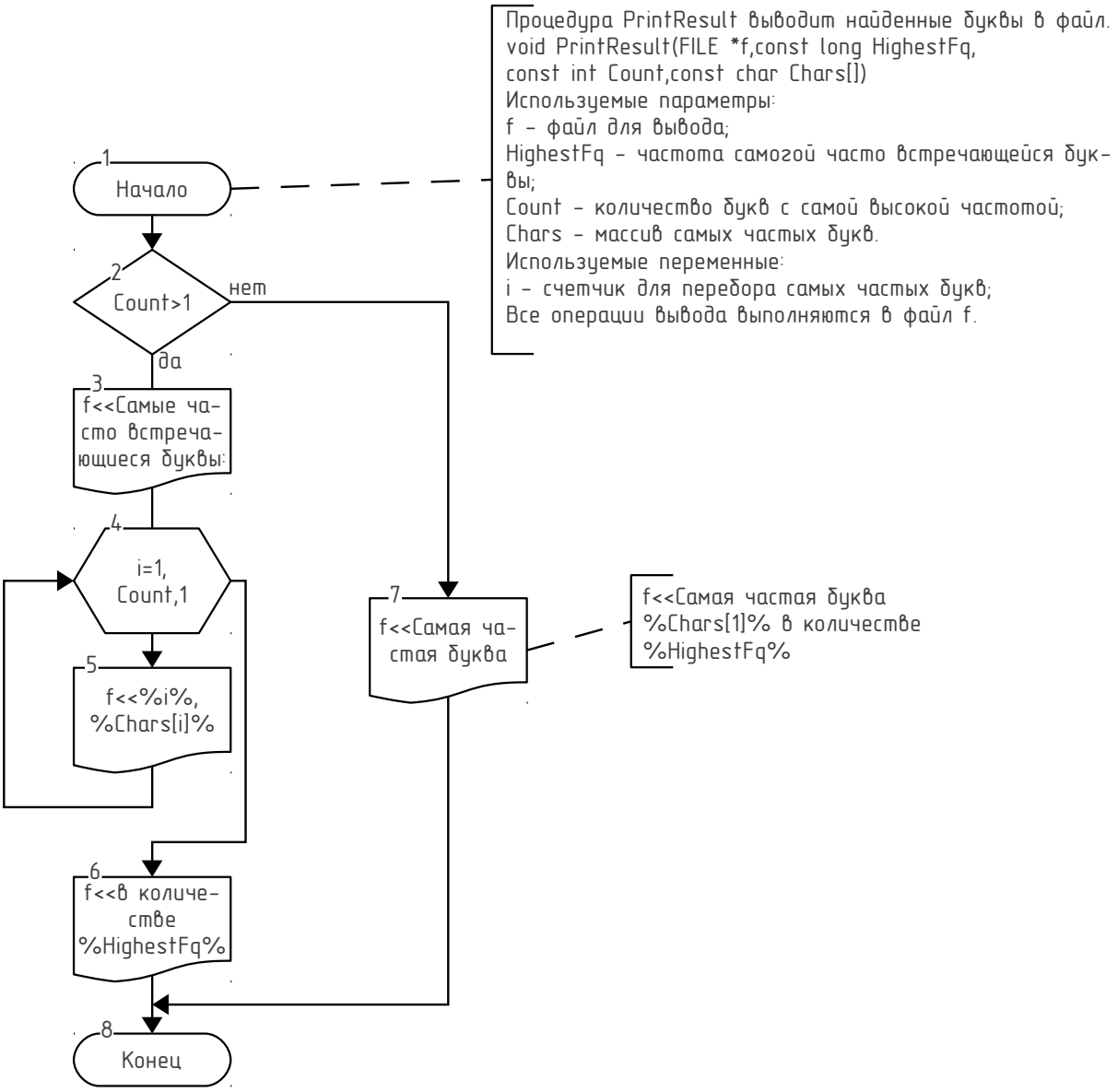

Рисунок 7 — Схема алгоритма вывода в файл самых частых букв

Инструкция пользователю

Данная программа позволяет найти самую часто используемую букву в текстовом файле.

Для работы программы необходимо ввести имена некоторых файлов. Первый файл - это файл, в котором находится текст для поиска. Его имя длинной не более 255 символов нужно передать программе. Далее передайте программе имя файла для вывода результатов поиска. Внимание! Если файл с указанным именем существует, то вся информация в нем будет стерта! Пожалуйста, проверьте имя файла, так как восстановить потерянные данные буде невозможно. В случае неправильного ввода имени файла-источника (если такой не существует), или если он имеет слишком большой размер, или при невозможности получить доступ к файлу для вывода, имеется возможность ввести заново имя этого файла, ответив на запрос программы о продолжении ввода 'Y', или отказаться от повторного ввода и завершить программу, ответив 'N'.

Если найдена всего одна самая часто встречающаяся буква, то будет выведена она и её частота в указанный файл и на экран; если существует несколько букв с самой высокой частотой появления, то они будут выведены нумерованным списком, и в конце будет выведена их частота, в указанный файл и на экран.

Инструкция программисту

При создании программы поиска одногруппников были предприняты следующие действия.

Были импортированы заголовочные файлы stdio.h - для функций ввода-вывода, ctype.h - для функции toupper(), limits.h - для константы UCHAR_MAX.

В основной части программы определены константы:

1. Alphabet - строка (char []), символы которой должны определяться как буквы;

Были введены структуры данных, описание которых представлено в таблице 1.

Таблица 1 - Структуры данных, используемые в в основной части программы поиска самой частой буквы в тексте

имя	тип	предназначение
fin	FILE *	Входной файл с текстом.
fout	FILE *	Выходной файл для записи результатов работы
HighestFq	long	Частота наиболее часто встречающейся буквы.
MostFqCount	int	Количество наиболее часто встречающихся букв.
CharsFq	long[255]	Массив частот символов кодировки в тексте.
MostFqChars	char[255]	Наиболее часто встречающиеся буквы текста.
log	int	Указывает на отсутствие ошибок (значение 1) ввода данных или желания пользователя прервать программу.

Кроме того, в процессе создания вышеуказанной программы были определены следующие подпрограммы:

1. Функция GetFiles запрашивает у пользователя все необходимые для работы файлы. Возвращает 1, если все необходимые файлы получены, иначе возвращается 0.

int GetFiles(FILE **fin,FILE **fout);

В теле функции через локальные функции GetInputFile, GetOutputFile запрашиваются имена файла-источника с текстом и текстового выходного файла соответственно. Если на каком-то этапе возникла ошибка, то последующие этапы

не выполняются и возвращается 0. Иначе функция завершает работу и возвращает 1.

Используемые функцией параметры-переменные приведены в таблице 2, локальные переменные - в таблице 3.

Таблица 2 - Параметры-переменные функции получения файлов

имя	**тип**	**предназначение**
fin	FILE **	Файл-источник с исходным текстом.
fout	FILE **	Выходной файл.

Таблица 3 - Локальные переменные функции получения файлов

имя	**тип**	**предназначение**
log	int	Указывает на отсутствие ошибок (значение 1) ввода данных.

Для получения файла каждого типа были введены следующие функции:

1.1 Функция GetInputFile запрашивает у пользователя имя файла-источника с анализируемым текстом, и для проверки открывает его для чтения. Возвращает 1, если операция проведена успешно, иначе возвращается 0. Для работы должна быть доступна функция toupper().

int GetInputFile(FILE **f)

В теле функции в цикле с постусловием происходит запрос у пользователя имени файла-источника, и проводится попытка его окрытия в режиме для чтения. Если файл открылся, функция завершает работу и передает открытый файл, и значение функции, равное 1. Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение 0. Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 4, локальные переменные - в таблице 5.

Таблица 4 - Параметры-переменные функции получения файла-источника

имя	**тип**	**предназначение**
f	FILE **	Файл-источник с исходным текстом.

Таблица 5 - Локальные переменные функции получения файла-источника

имя	тип	предназначение
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
fileName	char []	Имя файла-источника.
error	int	Флаг ошибки, возникшей при открытии файла(1 - нет ошибки) .

1.2 Функция GetOutputFile запрашивает у пользователя имя входного файла, и пытается открыть его для записи. Возвращает 1, если операция проведена успешно, иначе возвращается 0. Для работы должна быть доступна функция toupper().

int GetOutputFile(FILE **f)

В теле функции в цикле с постусловием происходит запрос у пользователя имени выходного файла, и проводится попытка его окрытия в режиме для записи. Если файл открылся успешно, функция завершает работу и передает открытый файл, и значение функции, равное 0. Если нет, то производится запрос пользователя о продолжении работы. Если пользователь ответил 'N', функция завершает работу и возвращает значение ошибки ввода-вывода, Иначе операция повторяется до достижения двух вышеуказанных условий.

Используемые функцией параметры-переменные приведены в таблице 6, локальные переменные - в таблице 9.

Таблица 6 - Параметры-переменные функции получения текстового выходного файла

имя	тип	предназначение
f	FILE **	Выходной файл.

Таблица 7 - Локальные переменные функции получения текстового выходного файла

имя	тип	предназначение
ch	char	Содержит ответ пользователя на запросы программы о повторении ввода данных.
fileName	char []	Имя выходного файла.
error	integer	Флаг ошибки, возникшей при открытии файла(1 - нет ошибки) .

2. Функция GetCharsFreq проводит частотный анализ текста, коды которого записаны в динамическом массиве. Использует константу UCHAR_MAX.

В начале массив частот символов кодировки (от 0 до UCHAR_MAX) обнуляется. Затем из файла считываются коды символов, и элемент массива частот, имеющий индекс, равный коду символа, увеличивается на единицу.

Используемые функцией параметры-переменные приведены в таблице 8; локальные переменные - в таблице 9.

Таблица 8 - Параметры-переменные процедуры посимвольного частотного анализа

имя	**тип**	**предназначение**
f	FILE *	Файл-источник с текстом для анализа.
CharsFreq	long []	Массив частот символов кодировки в тексте.

Таблица 9 - Локальные переменные процедуры посимвольного частотного анализа

имя	**тип**	**предназначение**
i	int	Код текущего символа, считанный из типизированного файла.

3. Функция GetMostFreq с помощью данных частотного анализа находит самые частые буквы в тексте. Использует константу Alpabet.

void GetMostFreq(const long CharsFreq[],long *HighestFreq, int *ChCount,

char MostFreqChars[])

При инициализации количество самых частых букв и наивысшая частота обнуляются.

В цикле с параметром перебираются буквы из строки Alpabet, и если частота её появления не равна 0 (элемент массива частот имеющий индекс, равный коду символа, не равен 0), то тогда если её частота больше максимальной, то эта частота запоминается как максимальная, количество букв устанавливается в 1, и буква добавляется в массив; если равна, то количество букв увеличивается на 1, и буква добавляется в массив.

Используемые функцией параметры-константы приведены в таблице 10; параметры-переменные - в таблице 11; локальные переменные - в таблице 12.

Таблица 10 - Параметры-константы процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
CharsFreq	long []	Массив частот символов кодировки в тексте.

Таблица 11 - Параметры-переменные процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
HighestFreq	long *	Частота наиболее часто встречающейся буквы.
ChCount	int *	Количество наиболее часто встречающихся букв.
MostFreqChars	char []	Наиболее часто встречающиеся буквы текста.

Таблица 12 - Локальные переменные процедуры поиска самых часто встречающихся букв

имя	тип	предназначение
i	int	Переменная для перебора ,букв - элементов строки Alpabet.
k	int	Временная переменная для хранения количества наиболее часто встречающихся букв.
lettercode	int	Код текущей буквы из Alphabet.

4. Процедура PrintResult выводит найденные самые частые буквы в тексте.

void PrintResult(FILE *f,const long HighestFq,const int Count,

const char Chars[])

Если в переданном массиве всего одна буква, то она и её частота просто выводятся в соответствующем сообщении; иначе выводятся нумерованный список букв, а в конце - частота их появлению.

Используемые процедурой параметры-константы приведены в таблице 13, параметры-константы - в таблице 14, локальные переменные - в таблице 15 .

Таблица 13 - Параметры-константы процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
HighestFq	long	Частота наиболее часто встречающейся буквы.
Count	int	Количество наиболее часто встречающихся букв.
Chars	char []	Наиболее часто встречающиеся буквы текста.

Таблица 14 - Параметры-переменные процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
f	FILE *	Выходной файл для записи результатов.

Таблица 15 - Локальные переменные процедуры вывода самых часто встречающихся букв

имя	тип	предназначение
i	int	Переменная-счетчик для обработки массива.

Текст программы

Ниже представлен текст программы, написанной на языке ANSI C для компиляторов Borland C++ 3.1 и GCC 4.5.2 (Linux 2.6.37 x86_64 и MinGW Windows 2000 SP4), которая находит самую часто встречающуюся букву в тексте.

```
#include <stdio.h> //для fopen/fclose,scanf/fscanf,printf/fprintf
#include <ctype.h> //для toupper()
#include <limits.h> //для UCHAR_MAX
////////////////////////////////////////////////////////////////////////////
const char
    //алфавит букв
    ALPHABET[]="qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM"
        "ёйцукенгшщзхъфывапролджэячсмитьбюЁЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮ";
////////////////////////////////////////////////////////////////////////////
/*-----ФУНКЦИЯ ПОЛУЧЕНИЯ ФАЙЛОВ-----*/
/*Параметры:
fin - файл для анализа,
fout - файл для вывода.*/
int GetFiles(FILE **fin,FILE **fout);

/*-----ФУНКЦИЯ ПОСИМВОЛЬНОГО ЧАСТОТНОГО АНАЛИЗА-----*/
/*Параметры:
f - файл для анализа,
CharsFreq  массив частот символов.*/
void GetCharsFreq(FILE *f, long CharsFreq[]);

/*-----ФУНКЦИЯ ПОЛУЧЕНИЯ САМЫХ ЧАСТЫХ БУКВ-----*/
/*Параметры:
CharsFreq - массив частот символов, HighestFreq - наивысшая частота букв,
ChCount - количество букв, MostFreqChars - самые частые буквы*/
void GetMostFreq(const long CharsFreq[],long *HighestFreq,
                                        int *ChCount, char MostFreqChars[]);

/*-----ФУНКЦИЯ ВЫВОДА САМЫХ ЧАСТЫХ БУКВ НА ЭКРАН-----*/
/*Параметры:
f - файл для вывода, HighestFq - наивысшая частота букв,
Count - количество букв, Chars - буквы для вывода*/
void PrintResult(FILE *f,const long HighestFq,const int Count,const char Chars[]);

////////////////////////////////////////////////////////////////////////////
int main(){
  FILE *fin,*fout; //входной/выходной файл
  int log; //флаг ошибки
  int MostFqCount; long HighestFq,CharsFq[255];
```

```
  //количество самых частых, наивысшая частота, массив частот
  char MostFqChars[255];
  //самые частые буквы
    printf("Программа находит самые часто встречающиеся буквы в
                                                          заданном тексте;\n");
    printf("и выводит результат на экран и в указанный текстовый файл.\n");
    log = GetFiles(&fin,&fout); //получить файлы
    if (log){
        printf("Выполняется посимвольный частотный анализ...");
        GetCharsFreq(fin,CharsFq);//провести частотный анализ
        fclose(fin); printf("Выполнено!\n");
        printf("Выполняется поиск самой частой буквы...\n");
        //найти самые частые буквы
        GetMostFreq(CharsFq,&HighestFq,&MostFqCount,MostFqChars);
        //что-нибудь найдено?
        if (MostFqCount!=0){
        PrintResult(stdout,HighestFq,MostFqCount,MostFqChars);
        PrintResult(fout,HighestFq,MostFqCount,MostFqChars);
    } else {
        printf("В тексте нет букв.\n");
        fprintf(fout,"В тексте нет букв.\n");
    }

    fclose(fout);
    }else{
    printf("Файлы не были открыты. Программа завершена.\n");
    }
    printf("Нажмите <Enter>...\n");
    (void)getchar();
    return 0;
}
////////////////////////////////////////////////////////////////////////////
int GetFiles(FILE **fin,FILE **fout){
//////////////////////////////////////
  /*-----ФУНКЦИЯ ПОЛУЧЕНИЯ ФАЙЛА ИСТОЧНИКА------*/
  int GetInputFile(FILE **f);
  /*-----ФУНКЦИЯ ПОЛУЧЕНИЯ ВЫХОДНОГО ФАЙЛА-----*/
  int GetOutputFile(FILE **f);
//////////////////////////////////////
  int log;
    log=GetInputFile(fin);
    if (log){
    log=GetOutputFile(fout);
    };
```

```
    return log;
};
//////////////////////////////////////
  int GetInputFile(FILE **f){
  char ch='Y';int error;
  char fileName[255];
    do{
        printf("Введите имя файла для анализа\n");
        //считать строку (gets() deprecated!)
        printf("Файл: ");scanf("%255[^\n]",fileName);
        while (!feof(stdin)&&(getc(stdin)!='\n'));
        *f=fopen(fileName,"r");
        error=*f==NULL;
        if (error) {
            printf("Неправильное имя файла! Повторить ввод? <Y>/<N>\n");
            ch=getchar();
            if (ch!='\n')    //Очистка буфера
              while (!feof(stdin)&&(getc(stdin)!='\n'));
        };
    } while (toupper(ch)!='N'&&error);
    return !error;
  };
//////////////////////////////////////
  int GetOutputFile(FILE * * f){
  char fileName [255];
  char ch='Y'; int error;
    do{
        printf("Введите имя файла-результата.\n");
        //считать строку (gets() deprecated!)
        printf("Файл: ");scanf("%255[^\n]",fileName);
        *f=fopen(fileName,"w");
        while (!feof(stdin)&&(getc(stdin)!='\n'));
        error=*f==NULL;
        if (error) {
            printf("Неправильное имя файла! Повторить ввод? <Y>/<N>\n");
            ch=getchar();
            if (ch!='\n')    //Очистка буфера
              while (!feof(stdin)&&(getc(stdin)!='\n'));
        };
    } while (toupper(ch)!='N'&&error);
    return !error;
  };
//////////////////////////////////////////////////////////////////////////
void GetCharsFreq(FILE *f, long CharsFreq[]){
```

```
  int i;
    for (i=0;i<=UCHAR_MAX;i++){
    CharsFreq[i]=0;
    };
    while ((i=getc(f))!=EOF){
    i=(unsigned char)i;
    CharsFreq[i]++;
    };
};
////////////////////////////////////////////////////////////////////////////
void GetMostFreq(const long CharsFreq[],long *HighestFreq, int *ChCount, char
MostFreqChars[]){
  int k,i; unsigned char lettercode;
    k=0; *HighestFreq=0;
    //просмотрим буквы алфавита
    for (i=0;ALPHABET[i];i++){
        lettercode=(unsigned char)ALPHABET[i];
        //такая буква встречается в тексте?
        if (CharsFreq[lettercode]>0) {
            //самая-самая частая буква
            if (CharsFreq[lettercode]>*HighestFreq) {
                *HighestFreq=CharsFreq[lettercode];
                k=0;  //забыть все уже не самые частые буквы
                MostFreqChars[k]=ALPHABET[i]; k++;
            } else
                //такая же частая?
                if (CharsFreq[lettercode]==*HighestFreq) {
                   MostFreqChars[k]=ALPHABET[i]; k++; //добавим к самым частым
                };
        };
    };
    *ChCount=k;
};
////////////////////////////////////////////////////////////////////////////
void PrintResult(FILE *f,const long HighestFq,const int Count,
                                                const char Chars[]){
  int i;
    if (Count==1){
        fprintf(f,"Наиболее часто встречается буква\n'%c' в количестве
                                                %ld шт.\n",Chars[0],HighestFq);
    } else {
        fprintf(f,"Наиболее часто встречаются буквы\n");
        for (i=0;i<=Count-1;i++){
            fprintf(f,"%u.'%c'. ",i+1,Chars[i]);
```

```
        };
        fprintf(f,"\n\rв количестве %ld шт.\n",HighestFq);
    };
};
```

Тестовые примеры

На рисунке 8 представлен пример работы программы для текста на русском языке размером примерно 90 Кбайт.

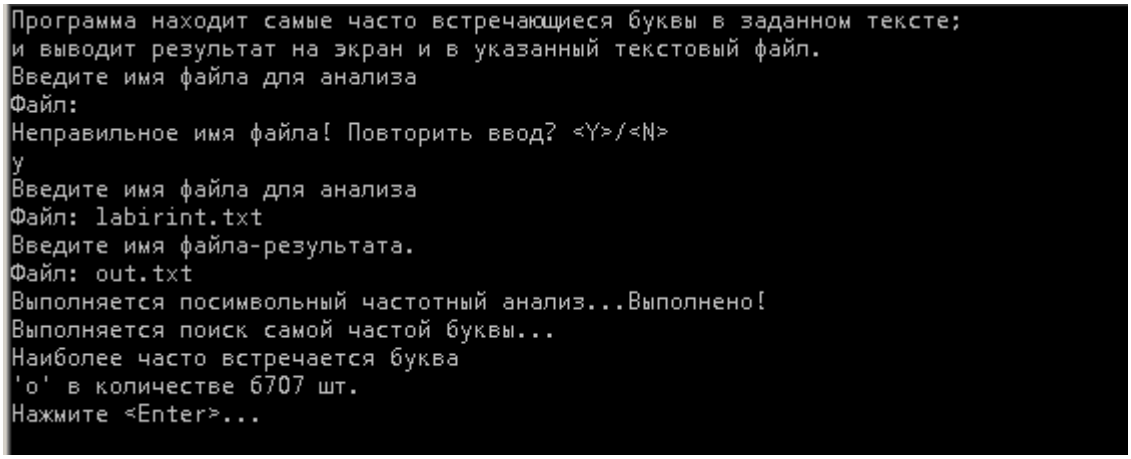

Рисунок 7 - Пример работы программы для произвольного файла

На рисунке 9 представлен пример работы программы для файла, содержащего строку “sssdddfffggghj”.

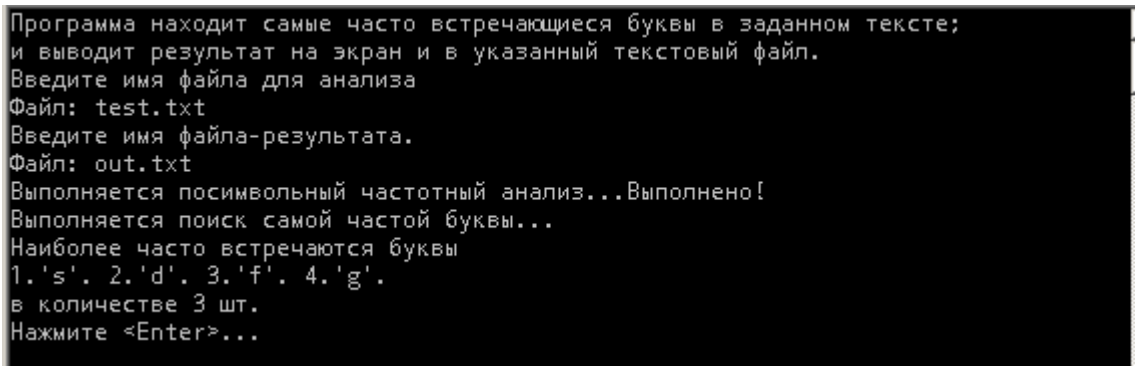

Рисунок 9 - Пример работы программы для файла с несколькими самыми частыми буквами

На рисунке 10 представлен результат обработки программой собственного текста.

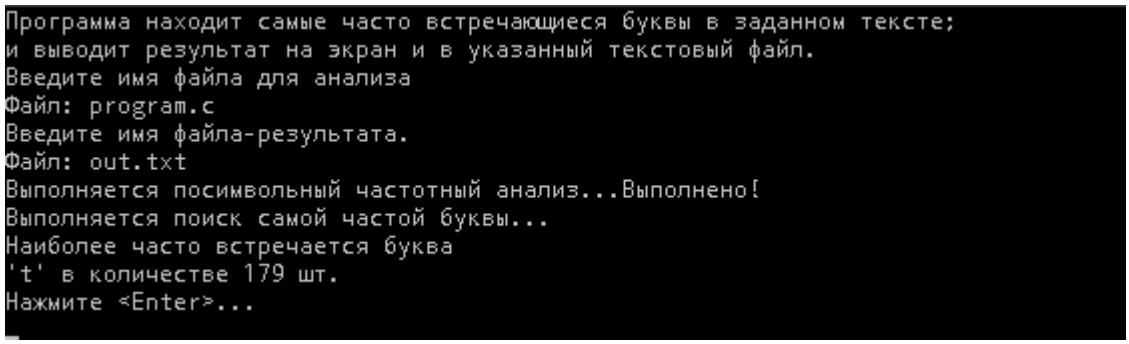

Рисунок 10 - Пример обрабоки программой собственного текста

Вывод

В ходе выполнения данной лабораторной работы я познакомился с языком Си и научился использовать функции ввода-вывода стандартной библиотеки этого языка. Данный язык представляет мощное средство для решения огромного пласта задач, но и как любой мощный инструмент, требует предельного внимания и аккуратности. Также в духе языка выдержаны и его функции - они чрезвычайно мощны, но взамен требуют большой внимательности.