

Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Пояснительная записка к курсовой работе
на тему:
“Обработка текста”
по дисциплине “Программирование”

Выполнил: Табаков А.В.

Группа 4306

Принял: к.т.н., доцент Сискович Т.И.

Санкт-Петербург
2014 г.

Оглавление

Цель	3
1. Задание	3
2. Уточнение задания	3
3. Контрольные примеры	3
4. Описание главной функции	4
5. Описание функций	6
5.1 Описание функции Menu	6
5.2 Описание функции Help	6
5.3 Описание функции Messages	6
5.4 Описание функции type_text	7
5.5 Описание функции print_text	9
5.6 Описание функции process_text	9
5.7 Описание функции String_check	11
5.8 Описание функции Charisdevide	13
5.9 Описание функции fr	13
6. Иерархическая структура вызова функций	14
7. Текст программы с комментариями	14
8. Инструкция пользователю	19
9. Набор тестов	19
10. Результаты решения задачи	20
Вывод	20

Цель

Получить практические навыки работы с символьными строками. Научиться выполнять основные функции с ними.

1. Задание

Разработать алгоритм и написать программу исходя из индивидуального задания.

Индивидуальное задание:

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 хотя бы два одинаковых слова.

2. Уточнение задания

В программе должно быть использовано простейшее меню. Выполнение программы должно быть многократным по желанию пользователя. В программе должны быть функции ввода текста, вывода текста, формирования нового текста и вывода его на экран. Все данные вводятся с клавиатуры. Для перехода к следующей строке использовать клавишу «Enter». Ввод заканчивается после двойного нажатия клавиши «Enter».

Пункты меню:

- 1: Справка
- 2: Ввод текста
- 3: Вывод исходного текста
- 4: Обработка текста
- 5: Вывод результата обработки
- 6: Выход

3. Контрольные примеры

Контрольные примеры представлены в Таблице 1.

Таблица 1. Контрольные примеры

№ примера	Исходные данные	Результат
1	Привет мир? Привет мир! Вася ходит по лесу.	Вася ходит по лесу.
2	-	Вы не ввели текст.
3	Чебурашка упал. Гена спал.	Новый текст не был сформирован. Во всех строках текста не было два одинаковых слова.
4	Лиса бегала по лесу Дома лучше, в лесу лучше В декабре снеговик лучше	Лиса бегала по лесу В декабре снеговик лучше

4. Описание главной функции

Назначение: начальная точка выполнения алгоритма.

Описание переменных функции

Описание переменных представлено в Таблице 2.

Таблица 2. Описание переменных главной функции

Имя переменной	Тип	Назначение
text	char**	Указатель на первый элемент первой строки исходного текста
rows	int	Количество строк исходного текста
Newtext	char**	Указатель на первый элемент первой строки сформированного текста
Newrows	int	Количество строк сформированного текста
check_process	int	Переменная проверки формирования текста
Q	int	Переменная выбора меню

Краткое описание алгоритма

Начало программы.

Шаг №1. Вывод меню.

Шаг №2. Выбор пользователем пункта меню.

Шаг №3. Переход к пункту, выбранным пользователем.

1: Справка. Переход к шагу 2

2: Ввод текста. Переход к шагу 2

3: Вывод исходного текста. Переход к шагу 2

4: Обработка текста. Переход к шагу 2

5: Вывод результата обработки. Переход к шагу 2

6: Выход. Переход к шагу 4.

Шаг №4. Конец программы.

Блок-схема главной функции

Блок-схема главной функции представлена на рисунке 1.

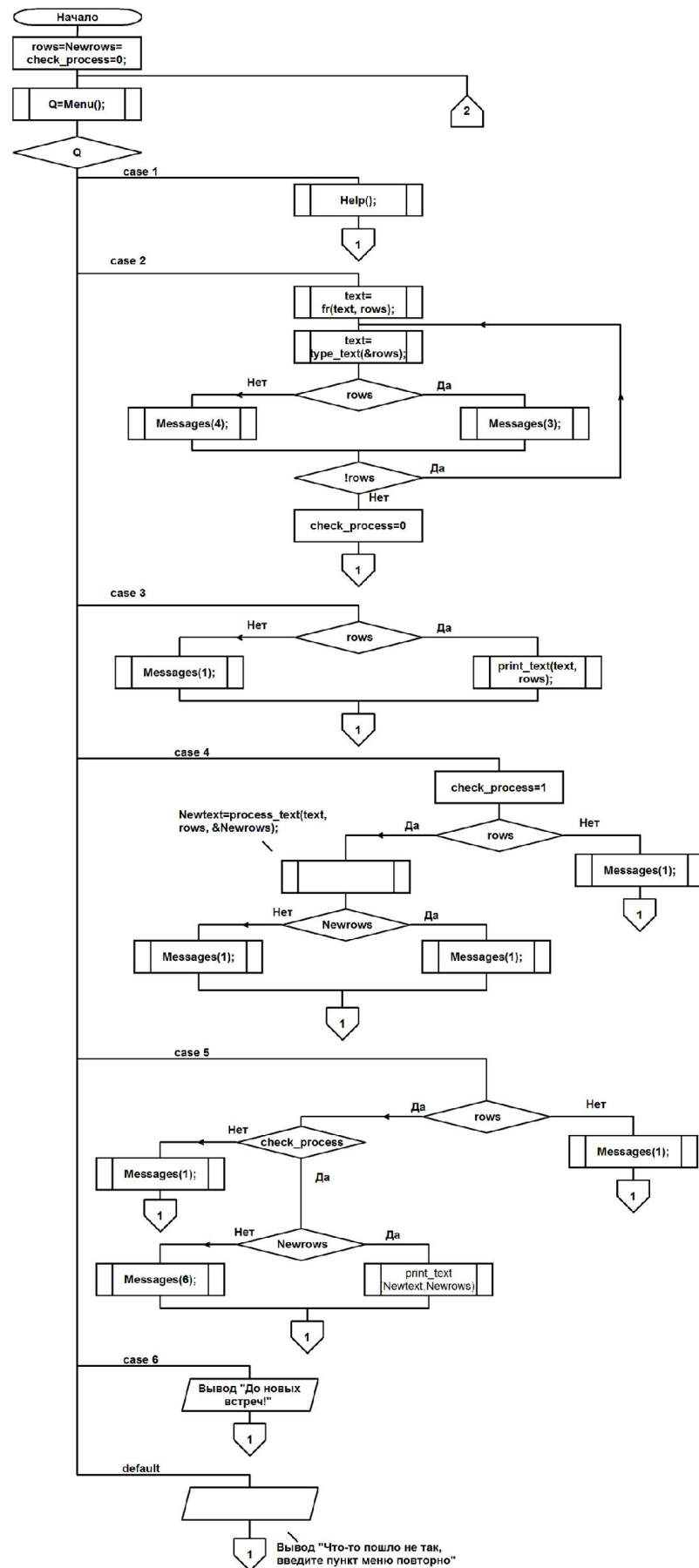


Рис. 1. Блок-схема главной функции (начало)

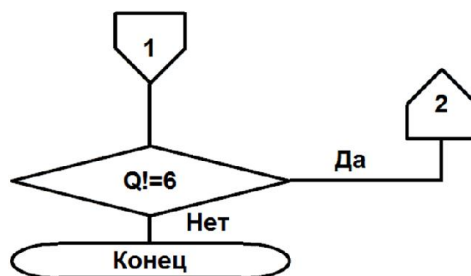


Рис. 1. Блок-схема главной функции (продолжение)

5. Описание функций

5.1 Описание функции Menu

Назначение: Функция используется для вывода меню программы.

Прототип: `int Menu();`, где `int` – тип возвращаемого значения.

Пример вызова: `Q=Menu();`, где `Q` – переменная выбора меню.

Вызывающая функция: `main`.

Блок схема функции не представлена из-за очевидности выполнения алгоритма.

5.2 Описание функции Help

Назначение: Функция используется для вывода справки.

Прототип: `void Help();`, где `void` – тип не возвращаемого значения.

Пример вызова: `Help();`

Вызывающая функция: `main`.

Блок схема функции не представлена из-за очевидности выполнения алгоритма.

5.3 Описание функции Messages

Назначение: Функция используется для ввода сообщений пользователю.

Прототип: `void Messages(int)`, где `int` номер вызываемого сообщения; `void` – тип не возвращаемого значения.

Пример вызова: `Messages(1);`, где «1» – номер вызываемого сообщения.

Вызывающая функция: `main`.

Блок схема функции не представлена из-за очевидности выполнения алгоритма.

Сообщения:

`Messages(1)`: "Сначала необходимо ввести текст"

`Messages(2)`: "Вы ввели текст, но не обработали его."

"Вам необходимо выбрать 4 пункт меню для обработки текста."

`Messages(3)`: "Текст введен успешно"

`Messages(4)`: "Ошибка, возможно, Вы не ввели текст?"

"Попробуйте ввести текст повторно."

`Messages(5)`: "Новый текст успешно сформирован."

`Messages(6)`: "Новый текст не был сформирован."

"В исходном тексте все строки с одинаковыми словами."

5.4 Описание функции type_text

Назначение: Функция используется для ввода текста.

Прототип: `char** type_text(int*)`;; где `int*` - указатель на количество строк исходного текста; `char**` – тип возвращаемого значения.

Пример вызова: `text=type_text(&rows)`;; где `text` указатель на первый элемент первой строки исходного текста, `&rows` – адрес кол-ва строк исходного текста.

Вызывающая функция: `main`.

Описание переменных

Описание переменных функции `type_text` представлены в Таблице 3.

Таблица 3. Описание переменных функции `type_text`

Имя переменной	Тип	Назначение
Локальные переменные		
text	char**	Указатель на адрес первого элемента первой строки исходного текста
s	char	Массив символов, размер которого 81
i,j	int	Счётчики
w,v	int	Флаги, отвечают за правильный ввод строки
Формальные переменные		
rows	int*	Указатель на кол-во строк исходного текста

Блок-схема type_text

Блок-схема функции `type_text` представлена на рисунке 2.

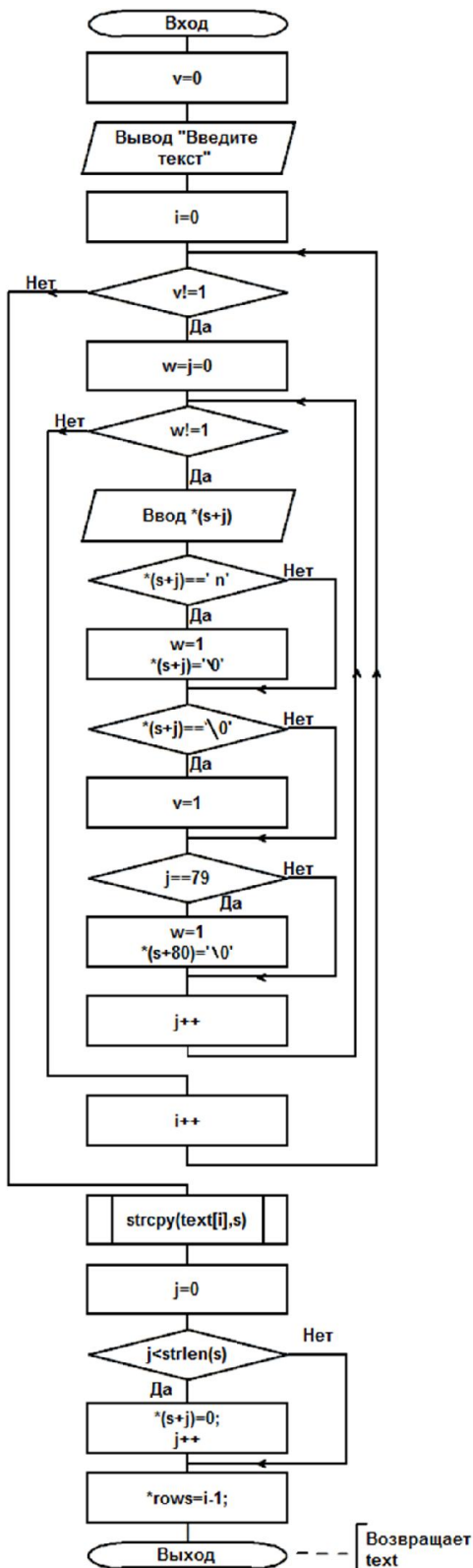


Рис. 2. Блок-схема функции type_text

5.5 Описание функции print_text

Назначение: Функция используется для вывода текста.

Прототип: void print_text(char**, int);, где char** - указатель на первый элемент первой строки текста, int – кол-во строк текста. void - тип не возвращаемого значения.

Пример вызова: print_text(text, rows);, где text – указатель на первый элемент первой строки текста, rows – кол-во строк текста.

Вызывающая функция: main.

Описание переменных

Описание переменных функции print_text представлены в Таблице 4.

Таблица 4. Описание переменных функции print_text

Имя переменной	Тип	Назначение
Локальные переменные		
i	int	Счётчик
Формальные переменные		
text	char**	Указатель на первый элемент первой строки текста
rows	int	Кол-во строк текста

Блок-схема print_text

Блок-схема функции print_text представлена на рисунке 3.

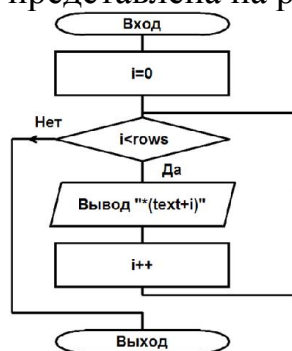


Рис. 3. Блок-схема функции print_text

5.6 Описание функции process_text

Назначение: Функция используется для обработки текста.

Прототип: char** process_text(char**, int, int*);, где char** - указатель на первый элемент первой строки исходного текста, int – кол-во строк исходного текста, int* - указатель на кол-во строк сформированного текста; char** – тип возвращаемого значения.

Пример вызова: Newtext=process_text(text, rows, &Newrows);, где Newtext – указатель на первый элемент первой строки нового текста text – указатель на первый элемент первой строки исходного текста, rows – кол-во строк исходного текста, &Newrows - адрес кол-ва строк нового текста.

Вызывающая функция: main
 Вызываемая функция: String_check.

Описание переменных

Описание переменных функции process_text представлены в Таблице 5.

Таблица 5. Описание переменных функции process_text

Имя переменной	Тип	Назначение
Локальные переменные		
Newtext	char**	Указатель на первый элемент первой строки сформированного текста
i,j,count	int	Счётчики
check	int	Флаг, отвечает присутствие в строке одинаковых слов
Формальные переменные		
text	char**	Указатель на первый элемент первой строки исходного текста
rows	int	Кол-во строк исходного текста
Newrows	int*	Указатель на кол-во строк сформированного текста

Блок-схема process_text

Блок-схема функции process_text представлена на рисунке 4.

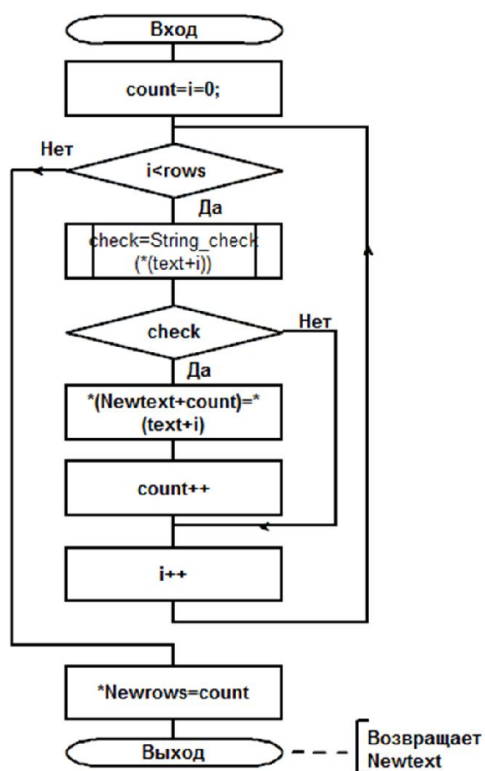


Рис. 4. Блок-схема функции process_text

5.7 Описание функции String_check

Назначение: Функция используется для проверки двух одинаковых слов в строке.

Прототип: int String_check(char*);, где char* - указатель на первый элемент строки; int – тип возвращаемого значения.

Пример вызова: check=String_check(*(text+i));, где check – флаг, отвечающий за проверку одинаковых слов в строке, *(text+i) – символьная строка.

Вызывающая функция: process_text.

Вызываемая функция: Charisdevide.

Описание переменных

Описание переменных функции String_check представлены в Таблице 6.

Таблица 6. Описание переменных функции String_check

Имя переменной	Тип	Назначение
Локальные переменные		
mas	int*	Массив длин слов
first	int	Номер первой буквы в слове
check	int	Флаг, отвечающий за проверку, есть ли в строке одинаковые слова
i, count, j, q, index	int	Счётчики
Формальные переменные		
text	char*	Указатель на первый элемент строки

Блок-схема String_check

Блок-схема функции String_check представлена на рисунке 5.

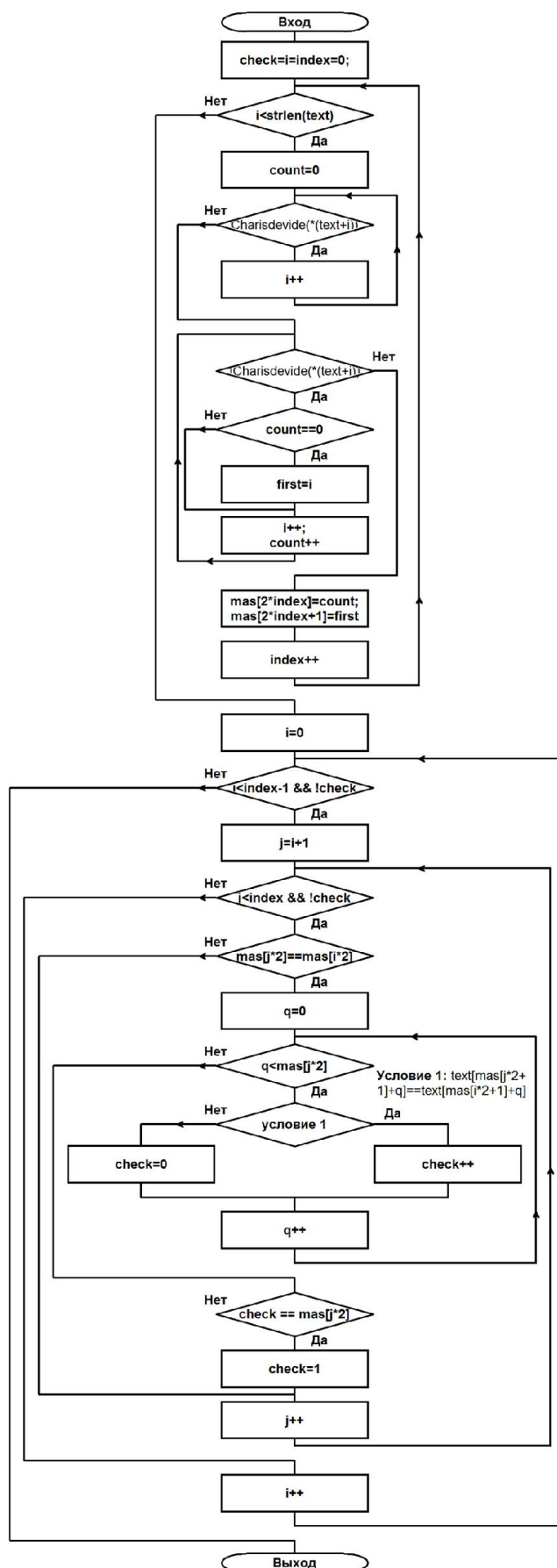


Рис. 5. Блок-схема функции String_check

5.8 Описание функции Charisdivide

Назначение: Функция используется для проверки символа на разделитель.

Прототип: `int Charisdivide(char);`, где `char` – проверяемый символ; `int`– тип возвращаемого значения.

Пример вызова: `Charisdivide(*(text+i))`, где `*(text+i)` *i*-ый символ в слове.

Вызывающая функция: `String_check`.

Описание переменных

Описание переменных функции представлены в Таблице 7.

Таблица 7. Описание переменных функции Charisdivide

Имя переменной	Тип	Назначение
Локальные переменные		
check	int	Флаг, отвечающий за проверку, является ли символ разделителем
i	int	Счётчик
Формальные переменные		
ch	char	Проверяемый символ

Блок-схема Charisdivide

Блок-схема функции Charisdivide представлена на рисунке 6.

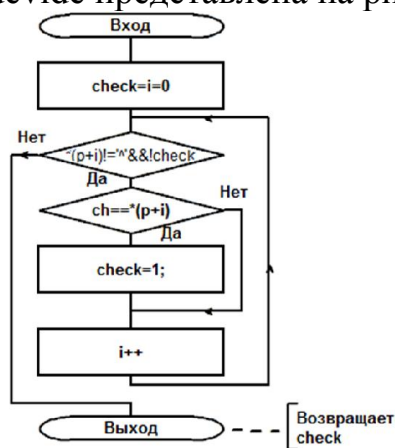


Рис. 6. Блок-схема функции Charisdivide

5.9 Описание функции fr

Назначение: Функция используется для освобождения памяти, выделенной ранее памяти под тексты.

Прототип: `char** fr(char**, int);`, где `char**` - указатель на первый элемент первой строки текста, `int` - кол-во строк текста; `char**` - тип возвращаемого значения.

Пример вызова: `text=fr(text, rows);`, где `text` – указатель на первый элемент первой строки текста, `rows` – кол-во строк текста.

Вызывающая функция: `main`.

Блок схема функции не представлена из-за очевидности выполнения алгоритма.

Описание переменных

Описание переменных функции fr представлены в Таблице 8.

Таблица 8. Описание переменных функции fr

Имя переменной	Тип	Назначение
Локальные переменные		
i	int	Счётчик
Формальные переменные		
text	int**	Указатель на первый элемент первой строки текста
rows	int	Кол-во строк текста

6. Иерархическая структура вызова функций

Иерархическая структура вызова функций представлена на рисунке 7.

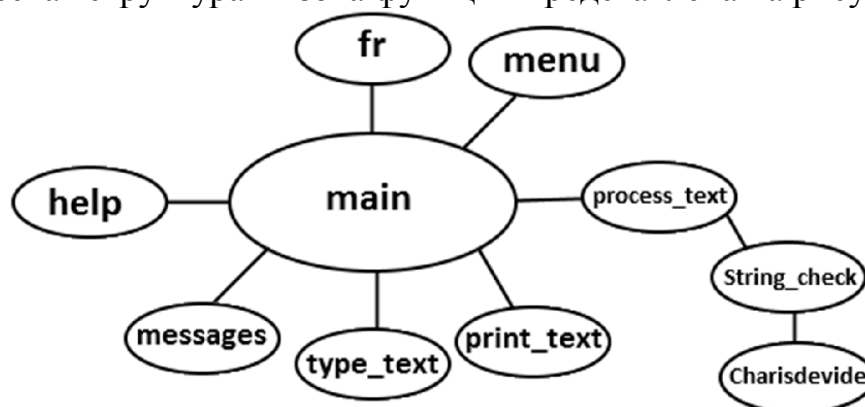


Рис. 7. Иерархическая структура вызова функций

7. Текст программы с комментариями

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <malloc.h>
#include <conio.h>
#include <string.h>
#include <windows.h>

int Menu(); //Прототип функции меню
void Help(); //Прототип функции справки
void Messages(int); //Прототип функции вывода сообщений пользователю
char** type_text(int*); //Прототип функции ввода текста
void print_text(char**, int); //Прототип функции вывода текста
char** process_text(char**, int, int*); //Прототип функции обработки текста
int String_check(char*); //Прототип функции проверки одинаковых слов в строке
int Charisdevide(char); //Прототип функции проверки разделителей
char** fr(char**, int); //Прототип функции освобождения памяти

int main()
{
    setlocale(LC_ALL, "RUS");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char** text=NULL;
```

```

char** Newtext=NULL;
int rows=0, Newrows=0, Q, check_process=0;
do
{
    Q = Menu();
    switch (Q)
    {
        case 1:                                     //Справка
            Help();
            break;

        case 2:                                     //Ввод текста
            text=fr(text, rows);
            do
            {
                text=type_text(&rows);
                if(rows)
                    Messages(3);
                else
                    Messages(4);
            }
            while(!rows);
            check_process=0;
            break;

        case 3:                                     //Вывод исходного текста
            if(rows)
                print_text(text, rows);
            else
                Messages(1);
            break;

        case 4:                                     //Обработка текста
            check_process=1;
            if(rows)
            {
                Newtext=process_text(text, rows, &Newrows);
                if(Newrows)
                    Messages(5);
                else
                    Messages(6);
            }
            else
                Messages(1);
            break;

        case 5:                                     //Вывод сформированного текста
            if(rows)
            {
                if(check_process)
                {
                    if(Newrows)
                        print_text(Newtext, Newrows);
                    else
                        Messages(6);
                }
                else
                    Messages(2);
            }
            else
                Messages(1);
            break;

        case 6:                                     //Выход
            system("cls");
            puts("До новых встреч!\n");
            system("pause");
            break;

        default:
            printf("Что-то пошло не так, введите пункт меню повторно\n");
            system("pause");
    }
}
while (Q!=6);
text=fr(text, rows);
Newtext=fr(Newtext, Newrows);
return 0;

```

```

    }

//*****
//Функция меню
//Возвращает номер пункта меню.
int Menu()
{
    int Q;
    system("cls");
    puts("Главное меню");
    puts("1 - Справка");
    puts("2 - Ввод текста");
    puts("3 - Вывод исходного текста");
    puts("4 - Обработка текста");
    puts("5 - Вывод сформированного текста");
    puts("6 - Выход");
    printf("Введите номер пункта - ");
    scanf("%d", &Q);
    printf("\n");
    fflush(stdin);
    return Q;
}

//*****
//Функция справка
void Help()
{
    system("cls");
    puts("\n\n Данная программа предназначена для формирования нового текста, который");
    puts(" является результатом следующего преобразования исходного текста: удалить");
    puts(" строки, содержащие хотя бы два одинаковых слова.");
    puts("\n Ввод текста пользователем заканчивается после нажатия двух раз клавиши Enter,");
    puts(" введенным через пробел от остального текста.\n");
    puts(" Если возникли проблемы обращайтесь, пожалуйста, на электронную почту:");
    puts(" komdosh@gelezo2.ru\n");
    system("pause");
}

//*****
//Функция вывода сообщений пользователю
//Принимает номер сообщения для вывода.
void Messages(int Q)
{
    system("cls");
    switch(Q)
    {
        case 1:
            puts("Сначала необходимо ввести текст");
            break;
        case 2:
            puts("Вы ввели текст, но не обработали его.");
            puts("Вам необходимо выбрать 4 пункт меню для обработки текста.");
            break;
        case 3:
            puts("Текст введен успешно");
            break;
        case 4:
            puts("Ошибка, возможно, Вы не ввели текст?");
            puts("Попробуйте ввести текст повторно.");
            break;
        case 5:
            puts("Новый текст успешно сформирован.");
            break;
        case 6:
            puts("Новый текст не был сформирован.");
            puts("В исходном тексте все строки с одинаковыми словами.");
            break;
    }
}

```



```

        system("pause");
    }

//*****
//Функция ввода текста
//Принимает указатель на кол-во строк.
//Возвращает адрес первого элемента первой строки исходного текста.
char** type_text(int *rows)
{
    system("cls");
    char** text=NULL;
    int i,j,w,v=0;
    char s[80];
    printf("Введите текст. Ввод текста заканчивается двойным нажатием клавиши «Enter».\n");
    for (i=0; v!=1;i++)
    {
        for(w=j=0;w!=1;j++)
        {
            *(s+j)=getchar();
            if(*(s+j)=='\n')
            {
                w=1;
                *(s+j)='\0';
            }
            if(*s=='\0')
                v=1;
            if(j==80)
            {
                w=1;
                *(s+81)='\0';
            }
        }
        text=(char**)realloc(text, (i+1)*sizeof(char*));
        *(text+i)=(char*)malloc((strlen(s)+1)*sizeof(char));
        strcpy(text[i],s);
        for(j=0;j<strlen(s);j++)
            *(s+j)=0;
    }
    *rows=i-1;
    return text;
}

//*****
//Функция вывода текста
//Принимает указатель на адрес первого элемента первой строки текста(text) и кол-во строк данного
//текста(rows).
void print_text(char **text, int rows)
{
    system("cls");
    for (int i = 0; i<rows; i++)
        puts(*(text+i));
    system("pause");
}

//*****
//Функция обработки текста
//Принимает указатель на адрес первого элемента первой строки текста(text), кол-во строк данного
//текста(rows), указатель на адрес кол-ва строк сформированного текста(Newrows).
//Возвращает указатель на адрес первого элемента первой строки нового текста(Newtext),
char** process_text(char** text, int rows, int* Newrows)
{
    int i,j,check,count=0;
    char** Newtext=NULL;
    system("cls");
    for(i=0;i<rows;i++)
    {
        check=String_check(*(text+i));
        printf("\nbreakpoint\n");
        system("pause");
    }
}

```

```

        if (check)
        {
            Newtext=(char**)realloc(Newtext, (count+1)*sizeof(char*));
            *(Newtext+count)=*(text+i);
            count++;
        }
    }
    *Newrows=count;
    return Newtext;
}

//*****
//Функция проверки одинаковых слов в строке
//Принимает указатель на первый элемент в строке(text).
//Возвращает значение переменной check. 1 если в строке есть одинаковые слова, 0 если таковое
//отсутствует
int String_check(char* text)
{
    int check=0,i=0,count,index=0, first, *mas=NULL, j, q;
    while(i<strlen(text))
    {
        count=0;
        while(Charisdevide(*(text+i)))
            i++;
        while(!Charisdevide(*(text+i)))
        {
            if(count==0)
                first=i;
            i++;
            count++;
        }
        mas=(int*)realloc(mas, 2*(index+1)*sizeof(int));
        mas[2*index]=count;
        mas[2*index+1]=first;
        index++;
    }
    for(i=0; i<index-1 && !check; i++)
    for(j=i+1; j<index && !check; j++)
        if(mas[j*2]==mas[i*2])
        {
            //Сравнение слов
            for(q=0; q<mas[j*2]; q++)
                if(text[mas[j*2+1]+q]==text[mas[i*2+1]+q])
                    check++;
            else
                check=0;
            if(check == mas[j*2])
                check=1;
        }
    return check;
}

//*****
//Функция проверки разделителей
//Принимает символ (ch).
//Возвращает значение переменной check. 1 если символ является разделителем, 0 если нет.
int Charisdevide(char ch)
{
    char p[]={ '!', ' ', '(', ')', '*', '\'', '-', ';', ':', '?', '!', '(', ')', '@', '[', '+',
               '_', '\0', '=', '#', '^' };
    int check=0, i=0;
    while (*(p+i)!='^' && !check)
    {
        if (ch==*(p+i))
            check=1;
        i++;
    }
    return check;
}

```

```

//*****
//Функция освобождения памяти
//Принимает указатель на адрес первого элемента первой строки текста(text), кол-во строк данного
//текста(rows).
//Возвращает NULL.
Char** fr(char** text, int rows)
{
    if (text!=NULL)
    {
        for (int i = 0; i<rows; i++)
            free(*(text+i));
        free(text);
    }
    return NULL;
}

```

8. Инструкция пользователю

Для вызова справки необходимо выбрать 1-ый пункт меню.

Для того чтобы ввести текст необходимо выбрать 2-ой пункт меню.

Текст может содержать произвольное количество строк.

Для перехода на следующую строку необходимо нажать клавишу «Enter».

Для прекращения ввода текста необходимо два раза нажать клавишу «Enter».

Если необходимы пустые строки в тексте, то необходимо использовать один раз клавишу «Enter» и один раз клавишу «Пробел» на каждой пустой строке.

Для вывода исходного текста на экран необходимо выбрать 3-ий пункт меню.

Для того чтобы обработать исходный текст, необходимо выбрать 4-ый пункт меню.

Для вывода нового текста на экран необходимо выбрать 5-ый пункт меню.

Для выхода необходимо выбрать 6-ой пункт меню

9. Набор тестов

Проверка программы выполняется с помощью тестов. Тестирование – проверка определённой части программы, сравнение результатов выданных программой для специально выбранных исходных данных, с ожидаемыми результатами.

Тест 1. Проверка правильности работы всей программы.

Вводим контрольные примеры из таблицы 1, и получаем результаты, совпадающие с ожидаемыми значениями из таблицы.

Тест 2. Проверка правильности работы функции вывода.

Пытаемся вывести исходный текст на экран, выбрав пункт 3, не введя текст, получаем сообщение «Сначала необходимо ввести текст».

Тест 3. Попытаемся вывести сформированный текст, введя исходный, но, предварительно не выбрав 4-ый пункт обработки, получаем сообщение «Вы ввели текст, но не обработали его».

Тест 4. Проверка правильности работы функции формирования нового текста.

Пытаемся сформировать текст, не введя исходный, и получаем сообщение «Сначала необходимо ввести текст».

10.Результаты решения задачи

При выполнении программы были получены результаты, совпадающие со значениями, приведенными в Таблице 1. Также были проведены тесты из 9 п. Набор тестов. Ошибок не обнаружено.

Вывод

При выполнении курсовой работы были получены практические навыки работы с символьными строками, указателями, функциями, динамической памятью на языке программирования «C/C++».