

Министерство науки и высшего 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

Отчёт по курсовой работе
по дисциплине «Программирование»
Тема: «Обработка текстовой информации»

Выполнил: студент гр.0306

Семёнов М.Д.

Проверила: к.т.н., доцент

Сискович Т.И.

Санкт-Петербург
2020

Содержание

Введение.....	3
1.Задание(Вариант 44).....	3
2.Уточнение задания.....	3
3.Описание решения.....	3
4.Контрольные примеры.....	4
5.Структура вызова функций.....	5
6.Функции.....	6-20
6.1 Функция main.....	6-7
6.2 Функция menu.....	8
6.3 Функция menu2.....	9-11
6.4 Функция count_words.....	12-13
6.5 Функция delete_small_words.....	14-15
6.6 Функция get_file.....	16-17
6.7 Функция get_console.....	18-19
6.8 Функция print_text.....	20
7.Краткое описание алгоритма.....	21
8.Текст программы.....	22-25
9.Результаты выполнения программы.....	26
Заключение.....	27

Введение

Целью работы является применение полученных за семестр теоретических знаний в языке С и получение практических навыков в обработке текстовой информации.

1.Задание (вариант 44)

Ввести текст с заданным количеством строк и строку символов-разделителей. Из строк введённого текста сформировать другой текст, в который войдут только строки, содержащие нечетное количество слов. Одновременно из этих строк удалить слова с минимальным в этой строке количеством символов. Вывести исходный и сформированный текст.

2.Уточнение задания

Из введённого текста удалить строки текста, в которых чётное число слов. В строках текста, в которых нечётное количество слов, удалить слова с наименьшим количеством букв.

3.Описание решения

Считываем текст, проходимся по строкам текста, считая количество слов в строке. Работаем над теми строками текста, в которых нечётное число слов. Выводим обработанный текст.

4.Контрольные примерые

Контрольные примеры представлены в табшлице 1.

Таблица 1. Контрольные примеры.

№	Исходные данные			Символы-разделители	Результат
	Файл (1-имя файла)/Консоль (2)	Кол-во строк	Исходные строки		
1	2	3	abra_cadabra_tundra a roza+upala na-lapu)azora mama mila+ramu	- _+)=	_cadabra_tundra +
2	1(test.txt)	4	qwer_tyu__yuiyu+++ tyui tyity++++tyi++++++u -----w-e-t	_+-	qwer__yuiyu+++ tyity++++tyi++++++ -----
3	2	1	Tyhty-tyht=t thty-y	_ -	
4	1(test1.txt)	7	There are wild and domestic animals. Wild animals live in the forest or in the zoo. Some animals are dangerous. A fox, a wolf, a bear, a hare, a tiger, an elephant are wild animals. Domestic animals live at home or at the farm. They are dogs, cats, rabbits, cows, horses, parrots. Animals that we have at home are called pets.	- .<,_=!&?	There wild domestic animals. Wild animals live Some animals
5	2	4	Autumn is yellow, Winter is white, Spring is green, Summer is bright!	, !	Autumn yellow, Winter white, Spring green, Summer bright!

5. Структура вызова функций

Структура вызова функций представлена на рисунке 1.

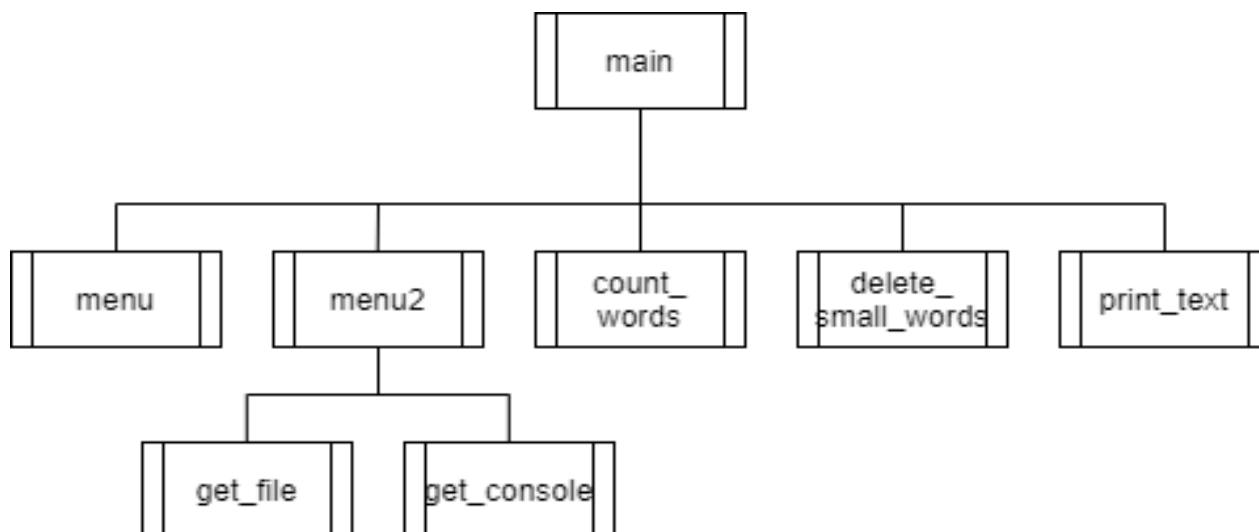


Рис. 1. Структура вызова функций

6.Функции

6.1 Функция main

Назначение: Служит стартовой точкой программы

Прототип функции : int main()

Функция возвращает код завершения программы

Используемые переменные представлены в таблице 2.

Таблица 2. Переменные функции main

Имя	Тип	Описание
vibor	int	Продолжить/закончить работу программы
option	int	Выбор работы программы (открытие файла или работа в консоле)
i	int	Счётчик
text	char**	Введённый пользователем массив символов
restext	char**	Обработанный массив символов
separators[MAXLEN]	char	Массив символов-разделителей
num_strings	int	Количество строк ,с которыми нужно работать программе.
num_r_strings	int	Количество строк ,которые содержатся в файле или кооличество строк,которые ввёл пользователь с консоли
itog_strings	int	Количество строк,над которыми нужно провести преобразования
itog_r_strings	int	Количество строк, которые уже преобразованы
deistv_words	int	Переменная ,которой присваивается количество слов в строке
menu1[2]	int	Массив для основного меню (Ввод/вывод текста и строки символов-разделителей)

Схема алгоритма представлена на рисунке 2.

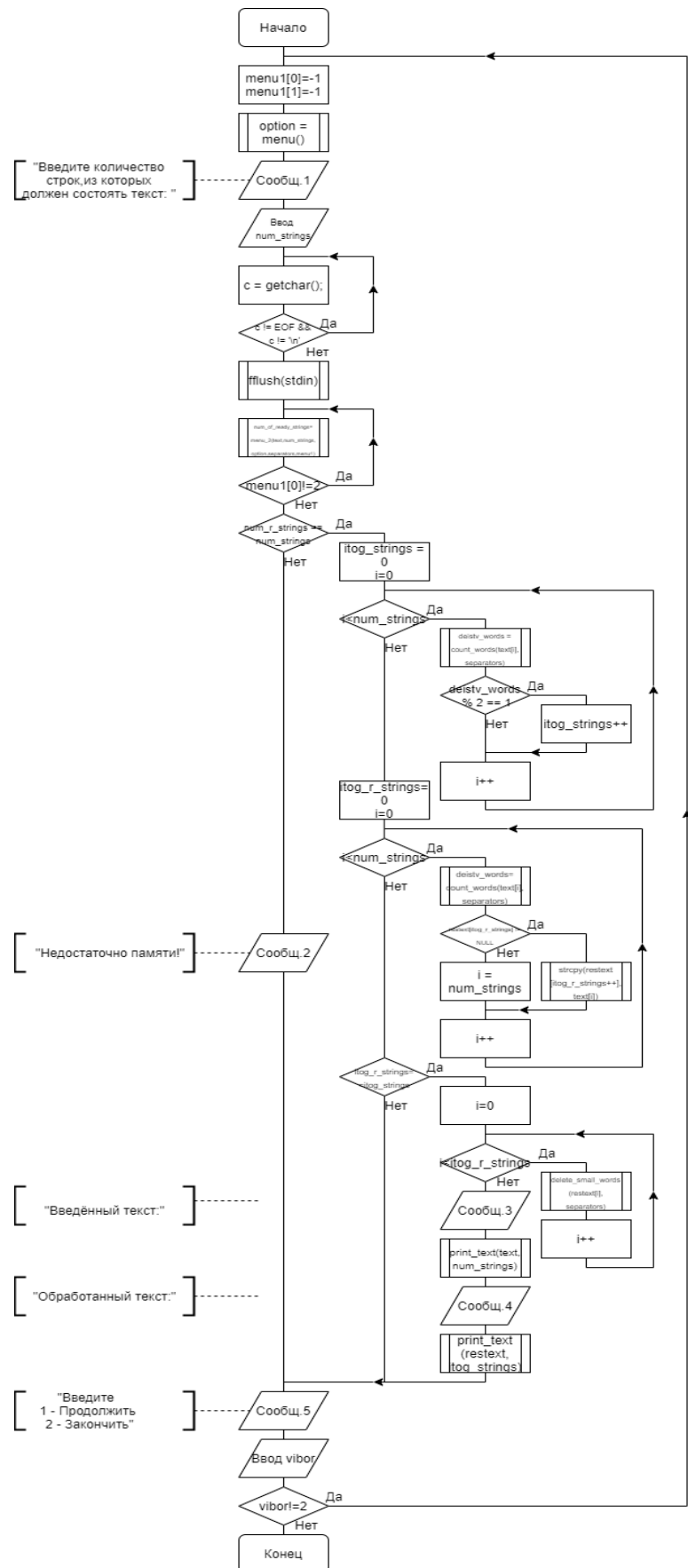


Рис. 2. Схема алгоритма функции main

6.2 Функция menu

Назначение: Служит для выбора пользователя

Прототип функции : int menu()

Функция возвращает значение, которое указывает откуда должен быть введен текст

Используемые переменные представлены в таблице 3.

Таблица 3. Переменные функции menu

Имя	Тип	Описание
input	int	1 - Работа в файле, 2 - Работа в консоле

Схема алгоритма представлена на рисунке 3.



Рис. 3. Схема алгоритма функции menu

6.3 Функция menu2

Назначение: Служит для создания основного меню

Прототип функции : `int menu2(char**text,int num_of_strings,int option,char* separators,int menu1[2])`

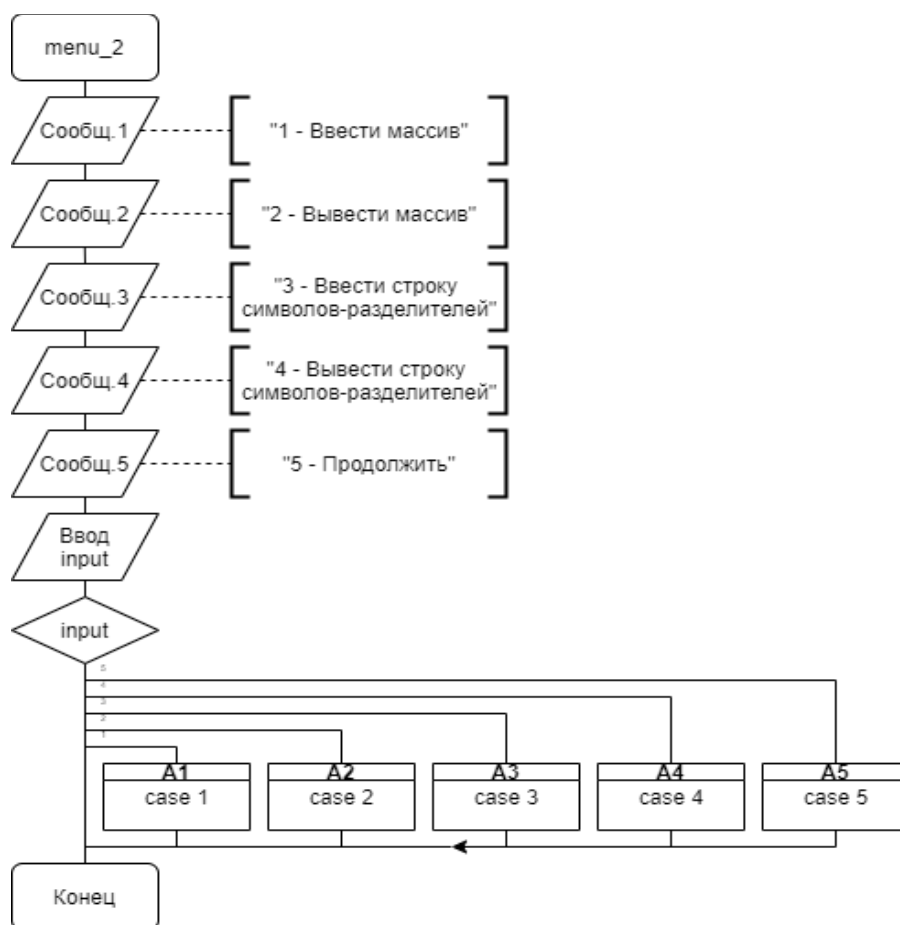
Функция возвращает значение,которое указывает количество строк в введённом тексте

Используемые переменные представлены в таблице 4.

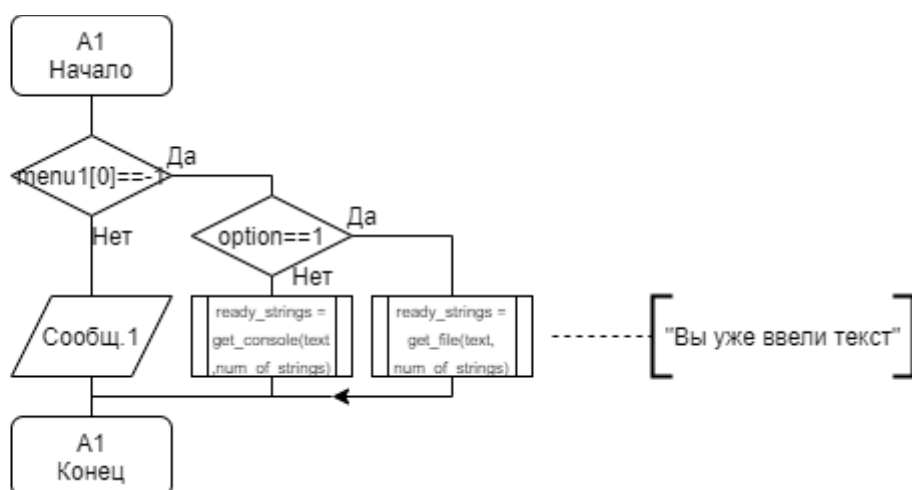
Таблица 4. Переменные функции menu2

Имя	Тип	Назначение
ready_strings	int	Количество строк,над которыми нужно провести изменения.
input	int	Выбор пользователем пункта меню.

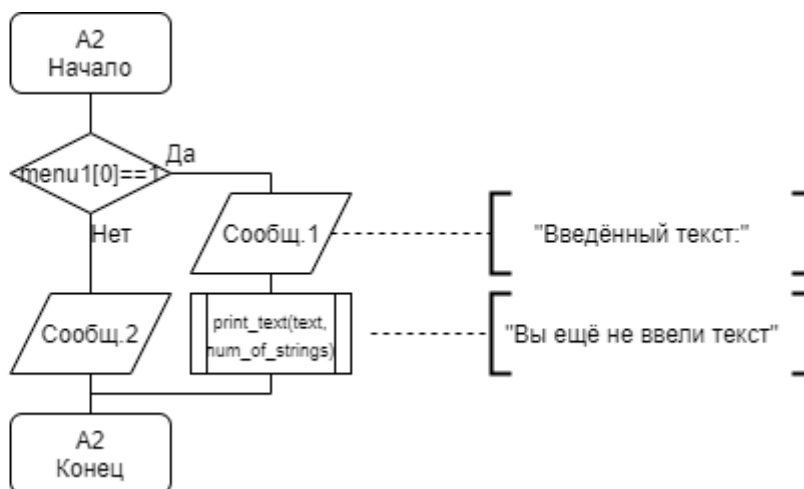
Схема алгоритма представлена на рисунке 4.



A1
case 1



A2
case 2



A3
case 3



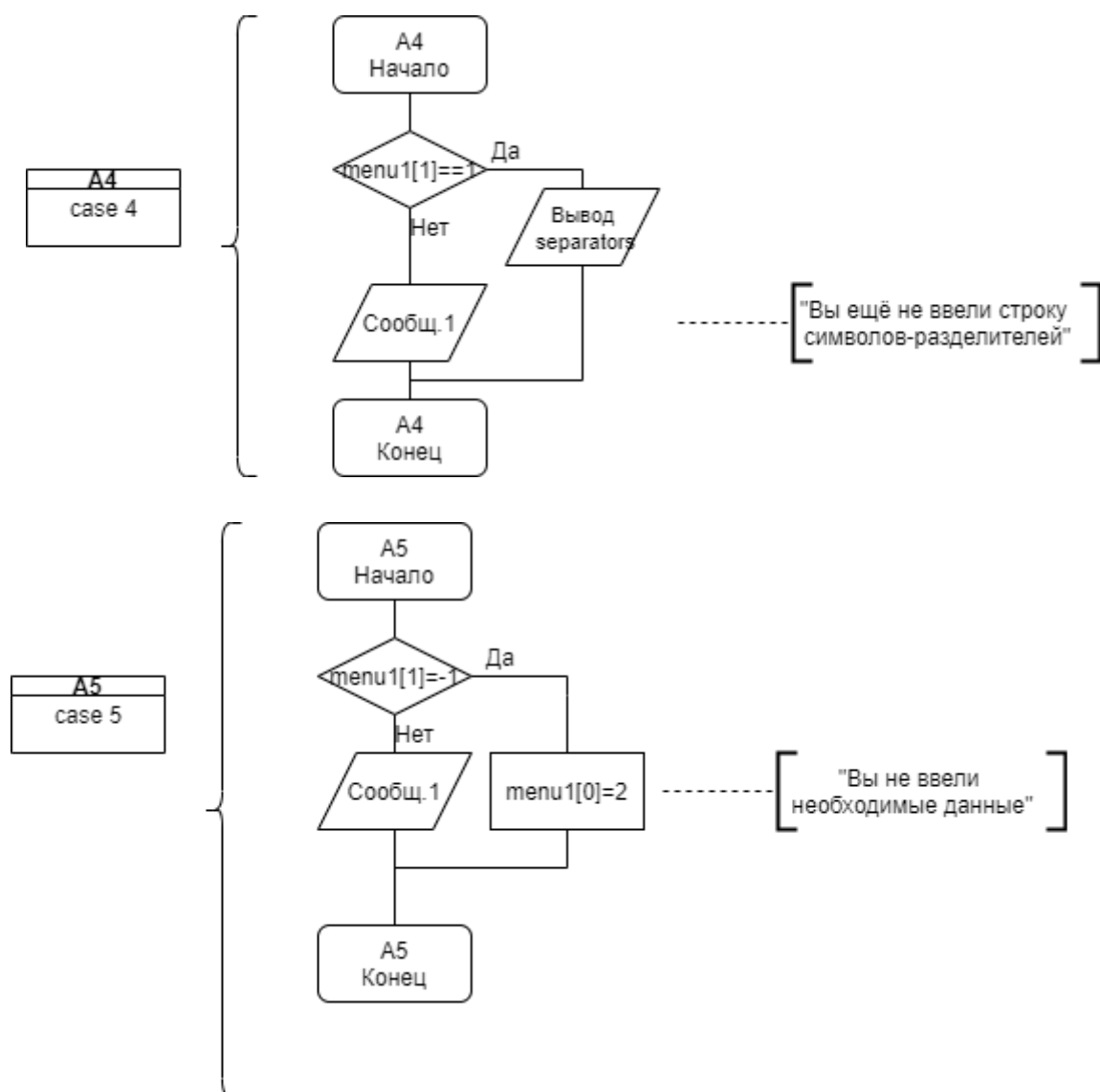


Рис. 4. Схема алгоритма функции `menu2`

6.4 Функция count_words

Назначение: Служит для подсчёта слов в строке

Прототип функции : int count_words(char* string, char* seps)

Функция возвращает количество слов в строке

Используемые переменные представлены в таблице 5.

Таблица 5. Переменные функции count_words

Имя	Тип	Назначение
i,j	int	Счётчики
flag	int	Определяет разделители в строке текста
schetchik_slov	int	Определяет количество слов в строке
deistv_word_len	int	Определяет равна ли длина слова 0 или нет.
string_len	int	Длина строки текста
seps_len	int	Длина строки символов-разделителей

Схема алгоритма представлена на рисунке 5.

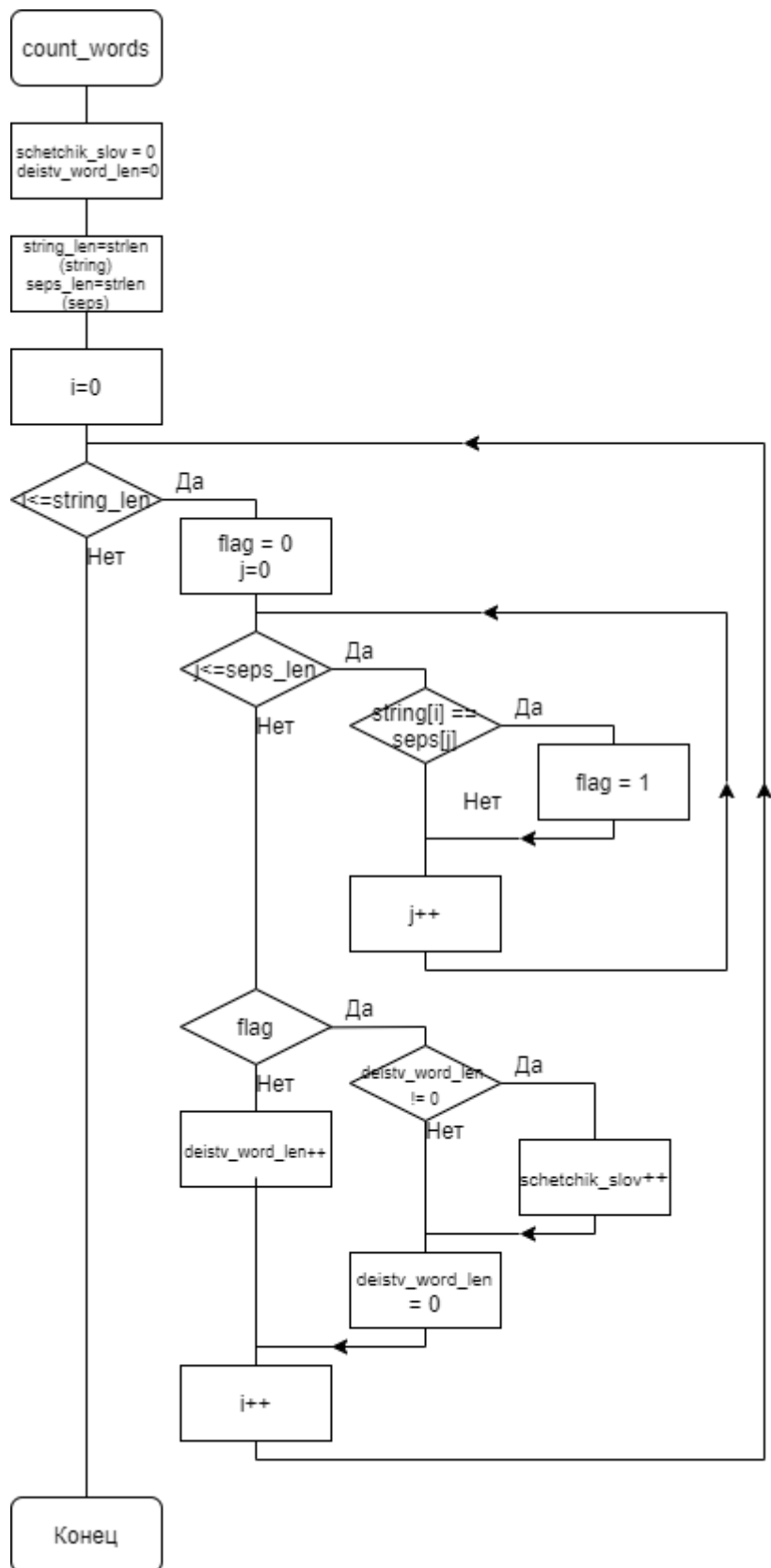


Рис. 5. Схема алгоритма функции `count_words`

6.5 Функция delete_small_words

Назначение: Удаление самых коротких слов из строки

Прототип функции : void delete_small_words(char* string, char* separators)

Функция удаляет из строки слова с наименьшим количеством букв

Используемые переменные представлены в таблице 6.

Таблица 6. Переменные функции delete_small_words

Имя	Тип	Назначение
i, j, k	int	Счётчики
flag	int	Определяет разделители в строке текста
min_word_len	int	Определяет минимальную длину слов, которые содержатся в строке
deistv_word_len	int	Определяет длины слов в строке
start	int	Начало слова
end	int	Конец слова

Схема алгоритма представлена на рисунке 6.

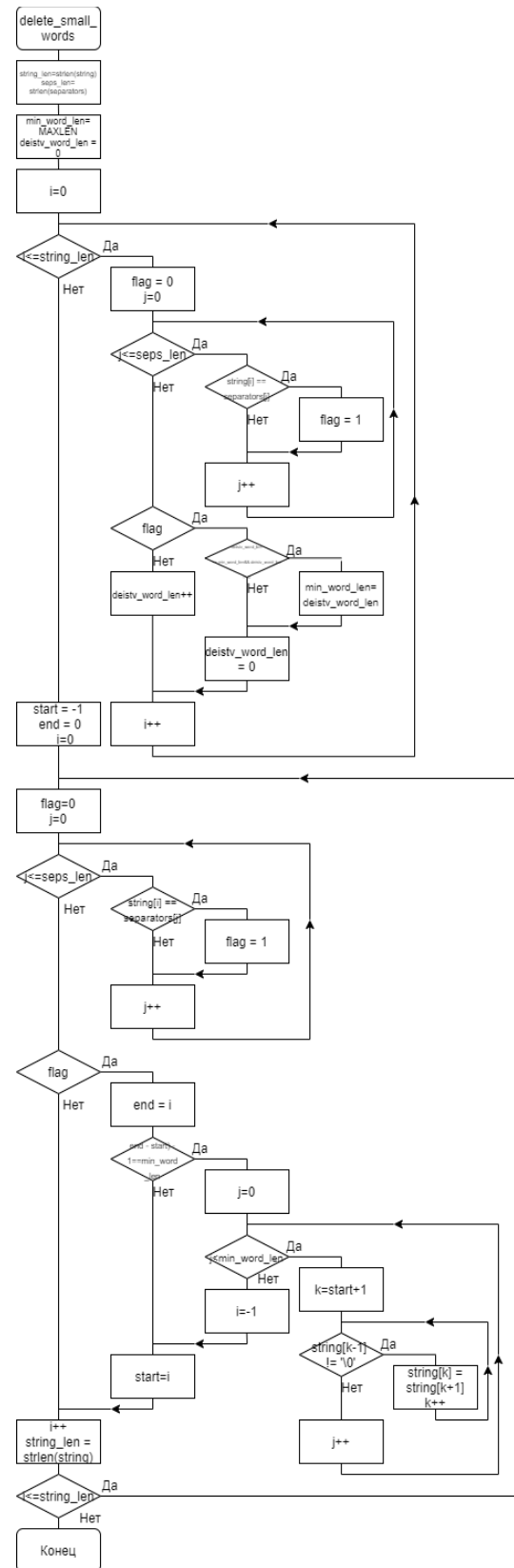


Рис. 6. Схема алгоритма функции delete_small_words

6.6 Функция `get_file`

Назначение: Получение текста из файла

Прототип функции : `int get_file(char** text, int size)`

Функция получает текст с заданным количеством строк из файла

Используемые переменные представлены в таблице 7.

Таблица 7. Переменные функции `get_file`

Имя	Тип	Назначение
<code>df</code>	<code>FILE*</code>	Указатель на файл, с которым ведется работа
<code>file_name</code>	<code>char*</code>	Строка, в которой задаётся имя файла
<code>deistv_string</code>	<code>char*</code>	Текущая строка
<code>i</code>	<code>int</code>	Счётчик
<code>deistv_string_len</code>	<code>int</code>	Длина текущей строки
<code>schetchik_bukv</code>	<code>int</code>	Количество созданных строк

Схема алгоритма представлена на рисунке 7.

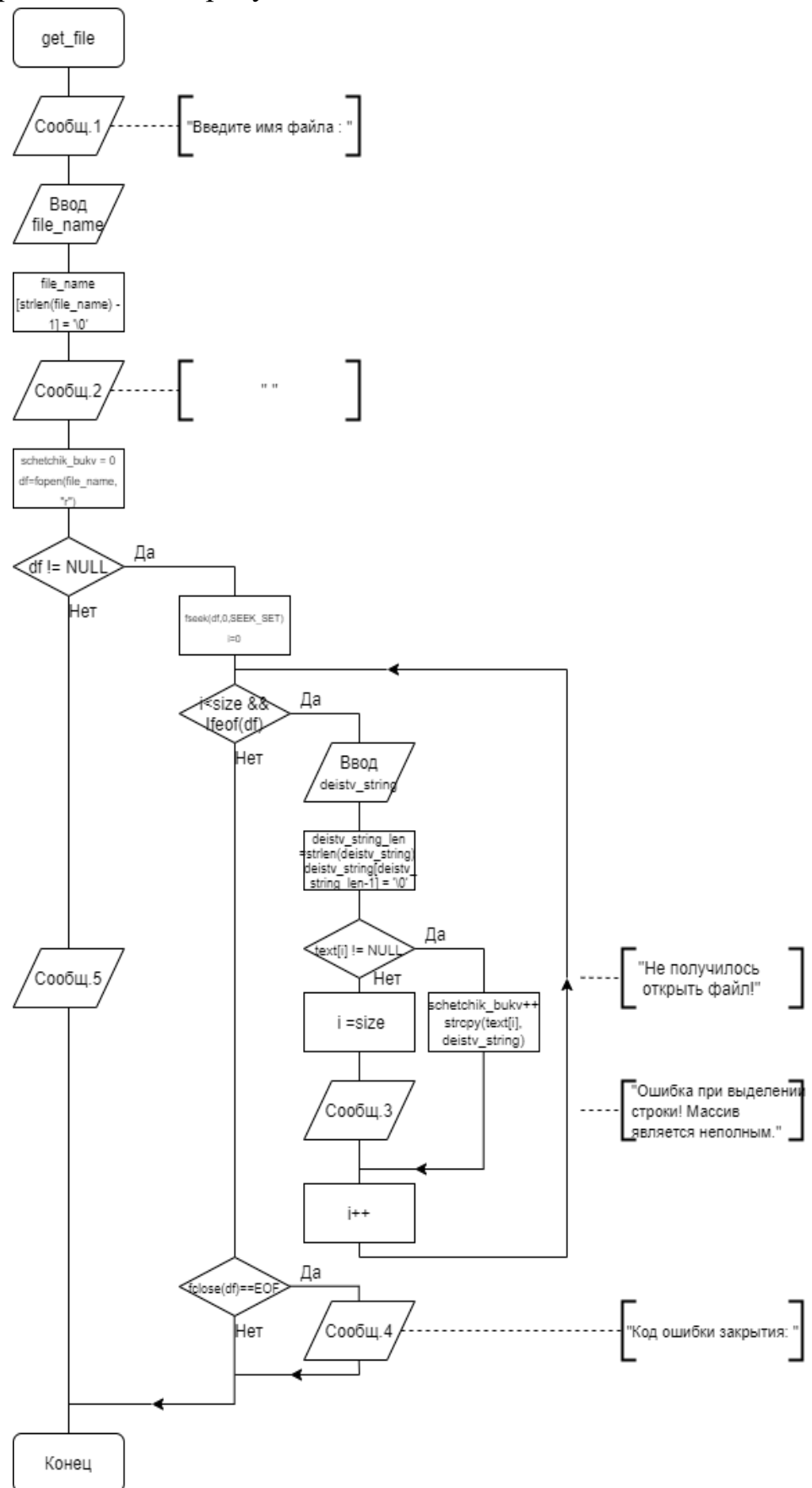


Рис. 7. Схема алгоритма функции get_file

6.7 Функция `get_console`

Назначение: Получение текста из консоли

Прототип функции : `int get_console(char** text, int size)`

Функция получает текст с заданным количеством строк из консоли

Используемые переменные представлены в таблице 8.

Таблица 8. Переменные функции `get_console`

Имя	Тип	Назначение
<code>deistv_string</code>	<code>char*</code>	Текущая строка
<code>i</code>	<code>int</code>	Счётчик
<code>string_count</code>	<code>int</code>	Количество созданных строк
<code>deistv_string_len</code>	<code>int</code>	Длина текущей строки

Схема алгоритма представлена на рисунке 8.

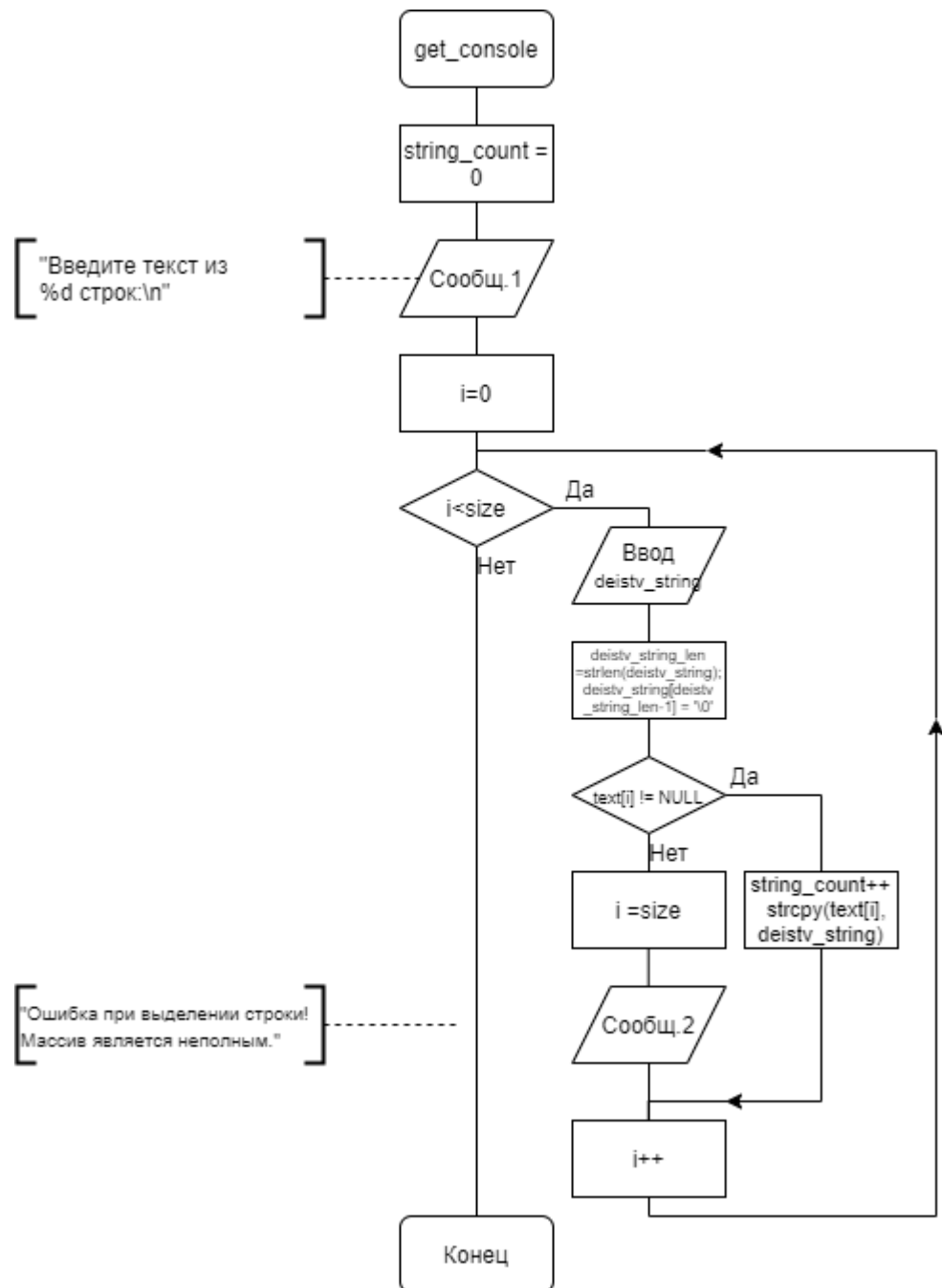


Рис. 8. Схема алгоритма функции `get_console`

6.8 Функция

Назначение: Вывод текста

Прототип функции : `int print_text(char** text, int size)`

Функция выводит текст на экран

Используемые переменные представлены в таблице 9.

Таблица 9. Переменные функции `print_text`

Имя	Тип	Назначение
i	int	Счётчик

Схема алгоритма представлена на рисунке 9.

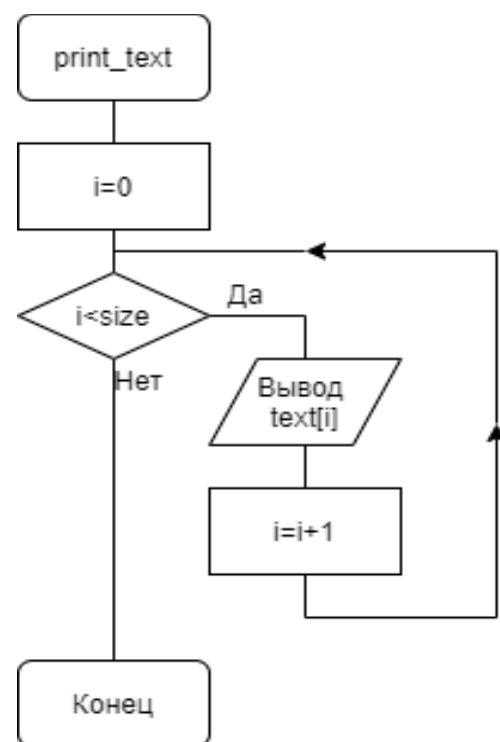


Рис. 9. Схема алгоритма функции `print_text`

7.Краткое описание алгоритма

Начало

1)Изначально пользователю предлагается 2 варианта продолжения работы программы (либо открыть файл,либо работать в консоле).Для этого используем функцию menu,которая получает введённое пользователем значение.

2)После выбора варианта работы необходимо запросить у пользователя количество строк,над которыми необходимо провести преобразования.

3)После открываем меню,где пользователю предлагается ввести текст и строку символов-разделителей.

4)Когда получены введённые данные начинаем обработку заданного текста.

5)В каждой строке текста начинаем проверку .Если в строке содержится нечётное количество слов,то в строке удаляем наименьшие слова;Иначе строка не подходит и выводить её не нужно.

6)В строке находим количество слов сравнивая каждый элемент строки текста с каждым элементом строки символов-разделителей.И если сравниваемые символы не равны, то символ строки является частью слова,иначе это символ-разделитель.Если длина слова не равна 0,то увеличиваем счётчик слов в строке на 1.

7)Находим минимальное значение длины слова в строке, и если в слове число символов равно минимальному значению длины слова,то удаляем это слово.

8)Выводим построчно обработанный текст на экран

9)Запрашиваем у пользователя ,хочет ли он продолжить работу программы.Если «Да» ,то ещё раз запускаем программу.Иначе работа программы завершается.

Конец

8.Текст программы

Исходный код программы представлен на рисунке 10.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <errno.h>
5  #include <limits.h>
6  #include <locale.h>
7  #define MAXLEN 256
8
9
10 void print_text(char** text, int size) { //функция вывода текста
11     int i;
12     for(i=0; i<size;i++) //условие выполнения цикла
13         printf("%s\n",text[i]); //вывод строки на экран
14     puts(""); //отступ
15
16 }
17
18 int get_file(char** text, int size) { //функция получения текста из файла
19     FILE *df;
20     char file_name[MAXLEN], deistv_string[MAXLEN];
21     int i;
22     int schetchik_bukv, deistv_string_len;
23     printf("Введите имя файла : "); //сообщение пользователю
24     fgets(file_name,MAXLEN,stdin); //получение имени файла
25     file_name[strlen(file_name) - 1] = '\0';
26     puts("");
27     schetchik_bukv = 0;
28     df=fopen(file_name, "r"); //открытие файла на чтение
29     if (df != NULL) {
30         fseek(df,0,SEEK_SET); //устанавливаем указатель
31         for (i=0;i<size && !feof(df);i++) { //условие выполнения цикла
32             fgets(deistv_string, MAXLEN, df); //считывание строки из файла
33             deistv_string_len = strlen(deistv_string);
34             deistv_string[deistv_string_len-1] = '\0';
35
36             text[i] = (char*)malloc(deistv_string_len*sizeof(char));
37
38             if (text[i] != NULL) {
39                 schetchik_bukv++; //увеличение на 1
40                 strcpy(text[i], deistv_string); //копирование строки
41             } else {
42                 i =size;
43                 puts("Ошибка при выделении строки! Массив является неполным.");
44             }
45         }
46
47         if(fclose(df)==EOF) printf("Код ошибки закрытия: %d\n", errno);
48
49     } else puts("Не получилось открыть файл!");
50
51     return schetchik_bukv;
52 }
53
```

```

54 int get_console(char** text, int size) { //функция получения файла из консоли
55     char deistv_string[MAXLEN];
56     int i;
57     int string_count, deistv_string_len;
58
59     string_count = 0;
60
61     printf("Введите текст из %d строк:\n", size); //сообщение пользователю
62
63     for(i=0; i<size; i++) {
64         fgets(deistv_string, MAXLEN, stdin); //ввод строки в консоли
65         deistv_string_len = strlen(deistv_string);
66         deistv_string[deistv_string_len-1] = '\0';
67
68         text[i] = (char*) malloc(deistv_string_len*sizeof(char));
69
70         if (text[i] != NULL) {
71             string_count++; //увеличение на 1
72             strcpy(text[i], deistv_string); //копирование строки
73         } else {
74             i = size;
75             puts("Ошибка при выделении строки! Массив является неполным."); //сообщение пользователю
76         }
77     }
78     return string_count;
79 }
80

```

```

81 int menu() {
82     setlocale(LC_ALL, "Рус"); //русский язык
83     int input;
84     do {
85         printf("Выберите:\n"); //сообщение пользователю
86         printf("1 - Открыть файл\n"); //сообщение пользователю
87         printf("2 - Работать в консоли\n"); //сообщение пользователю
88         printf("Введите ваш выбор: "); //сообщение пользователю
89
90         scanf("%d", &input); //ввод input
91     } while(((input<1) || (input>2))); //условие выполнения цикла
92     puts("");
93
94     return input;
95 }
96
97 int count_words(char* string, char* seps) { //функция подсчёта слов в строке
98     int i, j, flag;
99     int schetchik_slov, deistv_word_len;
100     int string_len, seps_len;
101
102     schetchik_slov = 0;
103     deistv_word_len = 0;
104     string_len = strlen(string);
105     seps_len = strlen(seps);
106
107     for(i=0; i<=string_len; i++) { //условие выполнения цикла
108         flag = 0;
109         for(j=0; j<=seps_len; j++) { //условие выполнения цикла
110             if (string[i] == seps[j]) flag = 1;
111         }
112         if (flag) {
113             if (deistv_word_len != 0) schetchik_slov++; //увеличиваем на 1
114             deistv_word_len = 0;
115         } else deistv_word_len++; //увеличиваем на 1
116     }
117     return schetchik_slov;
118 }
119

```

```

121 void delete_small_words(char* string, char* separators) { //функция удаления наименьшего слова из строки
122     int i, j, k, flag;
123     int string_len, seps_len;
124     int min_word_len, deistv_word_len;
125     int start, end;
126     string_len = strlen(string); //нахождение длины строки
127     seps_len = strlen(separators); //нахождение длины строки
128
129     min_word_len = MAXLEN;
130     deistv_word_len = 0;
131     for(i=0; i<=string_len; i++) { //условие выполнения цикла
132         flag = 0;
133         for (j=0; j<=seps_len; j++) if (string[i] == separators[j]) flag = 1; //условие выполнения цикла
134
135         if (flag) {
136             if (deistv_word_len <= min_word_len && deistv_word_len != 0) min_word_len = deistv_word_len;
137             deistv_word_len = 0;
138         } else deistv_word_len++;
139     }
140     start = -1; end = 0;
141     i=0;
142     do {
143         flag=0;
144         for(j=0; j<=seps_len; j++) if (string[i] == separators[j]) flag = 1;
145
146         if (flag) {
147             end = i;
148             if ((end - start) - 1 == min_word_len) {
149                 for(j=0; j<min_word_len; j++) { //условие выполнения цикла
150                     for(k=start+1; string[k-1] != '\0'; k++) string[k] = string[k+1]; //удаление символа
151                 }
152                 i = -1;
153             }
154             start=i;
155         }
156         i++;
157         string_len = strlen(string);
158     } while (i<=string_len); //условие выполнения цикла
159 }

```

```

161 int menu_2(char**text, int num_of_strings, int option, char* separators, int menul[2]) { //функция создания меню
162     int ready_strings;
163     int input;
164     printf("\n1 - Ввести массив\n"); //сообщение пользователю
165     printf("\n2 - Вывести массив\n"); //сообщение пользователю
166     printf("\n3 - Ввести строку символов-разделителей\n"); //сообщение пользователю
167     printf("\n4 - Ввести строку символов-разделителей\n"); //сообщение пользователю
168     printf("\n5 - Продолжить\n"); //сообщение пользователю
169     scanf("%d", &input);
170     system("cls"); //очистка окна
171     switch(input) {
172     case 2:
173         if (menul[0]==1) {
174             puts("\nВведенный текст:\n"); //сообщение пользователю
175             print_text(text, num_of_strings);
176             else puts("Вы еще не ввели текст"); //сообщение пользователю
177             break;
178         case 3: if (menul[1]==-1) {
179             puts("Введите строку символов-разделителей"); //сообщение пользователю
180             getchar();
181             fgets(separators, MAXLEN, stdin); //считывание строки
182             separators[strlen(separators)-1] = '\0';
183             menul[1]=1;
184             else puts("Вы уже ввели строку символов-разделителей"); //сообщение пользователю
185             break;
186         case 4:
187             if (menul[1]==1) printf("Строка символов-разделителей : %s", separators); //сообщение пользователю
188             else puts("Вы еще не ввели строку символов-разделителей"); //сообщение пользователю
189             break;
190         case 5:
191             if (menul[0]==1 && menul[1]==1) menul[0]=2;
192             else puts("Вы не ввели необходимые данные"); //сообщение пользователю
193             break;
194         case 1:
195             if (menul[0]==-1) {
196                 getchar();
197                 if (option==1) {
198                     ready_strings = get_file(text, num_of_strings); //вызов функции
199                     else ready_strings = get_console(text, num_of_strings); //вызов функции
200                     menul[0]=1;
201                     else puts("Вы уже ввели текст"); //сообщение пользователю
202                 }
203             }
204     }

```



```

205 int main(){
206     int vibor;
207     do{
208         int option;
209         int i, c;
210         char **text, **retext;
211         char separators[MAXLEN];
212         int num_strings, num_r_strings;
213         int itog_strings, itog_r_strings;
214         int deistv_words;
215         int menul[2];
216         menul[0]=-1;
217         menul[1]=-1;
218         option = menu();
219         printf("Введите количество строк, из которых должен состоять текст: "); //сообщения пользователю
220         scanf("%d", &num_strings); //ввод переменной
221
222         do {c = getchar();} while (c != EOF && c != '\n'); //условие выполнения цикла
223         fflush(stdin);
224
225         text = (char**)malloc(num_strings * sizeof(char*)); //выделение памяти
226
227         if (text!=NULL) {
228
229             do{
230                 num_r_strings= menu_2(text,num_strings,option,separators,menul); //вызов функции
231                 }while((menul[0]!=2)); //условие выполнения цикла
232
233                 if (num_r_strings == num_strings) {
234                     itog_strings = 0;
235                     for(i=0;i<num_strings;i++) { //условие выполнения цикла
236                         deistv_words = count_words(text[i], separators); //вызов функции
237                         if (deistv_words % 2 == 1) itog_strings++; //начётное количество слов в строке
238                     }
239                     retext = (char**)malloc(itog_strings*sizeof(char*)); //выделение памяти
240
241                     if (retext != NULL) {
242                         itog_r_strings= 0;
243                         for(i=0;i<num_strings;i++) { //условие выполнения цикла
244                             deistv_words= count_words(text[i], separators); //вызов функции
245                             if (deistv_words % 2 == 1) { //нечётное количество слов в строке
246                                 retext[itog_r_strings] = (char*)malloc((strlen(text[i])+1)*sizeof(char));
247                                 if (retext[itog_r_strings] != NULL) {
248                                     strcpy(retext[itog_r_strings++], text[i]); //копирование строки
249                                 } else {
250                                     puts("Ошибка при выделении памяти строки для обработанного текста!"); //сообщение пользователю
251                                     i = num_strings;
252                                 }
253                             }
254                         }
255                     }
256                     if (itog_r_strings==itog_strings) {
257                         for(i=0;i<itog_r_strings;i++) delete_small_words(retext[i],separators); //вызов функции
258                         puts("\nВведённый текст:\n"); //сообщения пользователю
259                         print_text(text, num_strings); //вызов функции
260                         printf("\nОбработанный текст:\n");
261                         print_text(retext,itog_strings); //вызов функции
262                     }
263                     for(i=0;i<itog_r_strings;i++) { //условие выполнения цикла
264                         free(retext[i]);
265                         retext[i] = NULL;
266                     }
267                     free(retext);
268                     retext = NULL;
269                 } else puts("Недостаточно памяти!"); //сообщения пользователю
270             } else puts("Недостаточно памяти!"); //сообщения пользователю
271             for (i=0;i<num_r_strings;i++) { //условие выполнения цикла
272                 free(text[i]);
273                 text[i] = NULL;
274             }
275             free(text);
276             text = NULL;
277         } else puts("Недостаточно памяти!"); //сообщения пользователю
278
279         printf("Введите \n 1 - Продолжить \n 2 - Закончить"); //сообщения пользователю
280         scanf("%d",&vibor);
281         system("cls");
282     }while(vibor!=2); //условие выполнения цикла
283     return 0;
284 }
285

```

Рис. 10. Исходный код программы

9. Результаты выполнения программы

Результаты, полученные при работе программы соответствуют примерам из таблицы 1.

Результат работы программы представлен на рисунке 11.

```
Введённый текст:  
abra_cadabra_tundra  
a roza+upala na-lapu)azora  
мама mila+ramu
```

```
Обработанный текст:  
_cadabra_tundra  
+
```

```
Введите  
1 - Продолжить  
2 - Закончить_
```

```
Введённый текст:  
  
There are wild and domestic  
animals. Wild animals live in  
the forest or in the zoo.  
Some animals are  
dangerous. A fox, a wolf, a  
bear, a hare, a tiger, an  
elephant are wild animals.
```

```
Обработанный текст:  
There wild domestic  
animals. Wild animals live  
Some animals
```

```
Введите  
1 - Продолжить  
2 - Закончить
```

Рис. 11. Результаты выполнения программы

Заключение

В результате выполнения работы были закреплены знания по обработке массивов, строк, текстов в языке си.