

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
федеральное государственное автономное образовательное  
учреждение высшего образования  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники  
Отделение автоматизации и робототехники  
Направление мехатроника и робототехника

**Отчет**  
по лабораторной работе №4  
по дисциплине  
«Основы программирования и алгоритмизации»

**Строки в C++**

Выполнил:

Студент группы 8E21

\_\_\_\_\_

Н.С. Моисеев

Проверил:

Ассистент ОАР ИШИТР

\_\_\_\_\_

Я.О. Кургинов

Томск 2023

## Вариант 10

### Цель работы:

Поработать со строками в C++.

### Задание 1

Найдите количество вхождения 'aba' в строку.

### Блок схема (Рисунок 1)

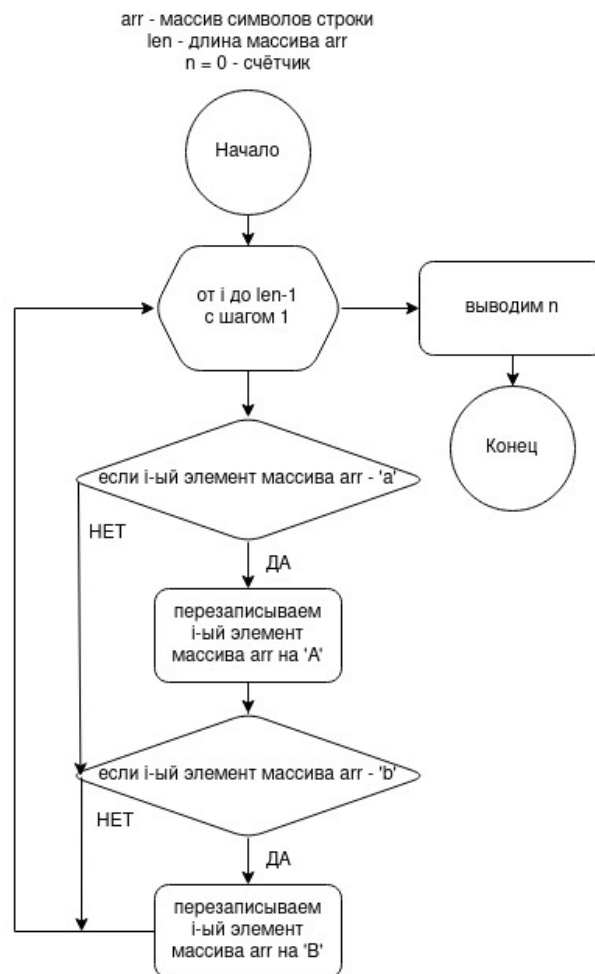


Рисунок 1 - Блок-схема к заданию 1.

Текст программы (Листинг 1, Приложение А)

## Результаты работы (Рисунок 2)

```
Input file name:
```

```
line.txt
```

```
2
```

```
** Process exited - Return Code: 0 **
```

Рисунок 2 - Скриншот результата работы программы к заданию 1 на консоли.

## Задание 2:

Дана строка. Заменить все символы 'a' и 'b' на 'A' и 'B' соответственно.

## Блок схема (Рисунок 3)

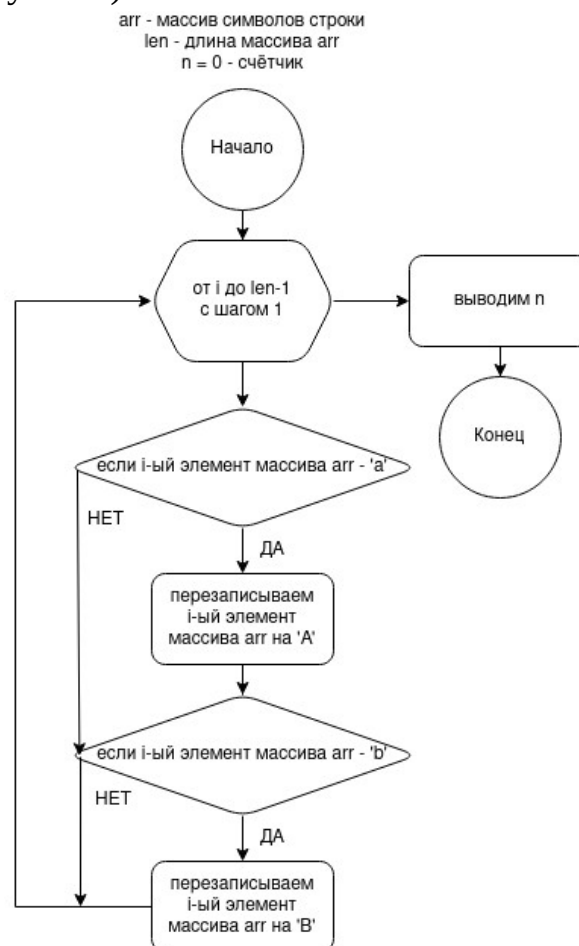


Рисунок 3 - Блок-схема к заданию 2.

Текст программы (Листинг 2, Приложение А)

### Результаты работы (Рисунок 4)

```
Input file name:
line.txt
AAA_BBA___BABA_B

** Process exited - Return Code: 0 **
```

Рисунок 4 - Скриншот результата работы программы к заданию 2 на консоли.

### Задание 3

В данной строке вставить после каждого символа 'a' символ 'b'.

### Блок схема (Рисунок 5)

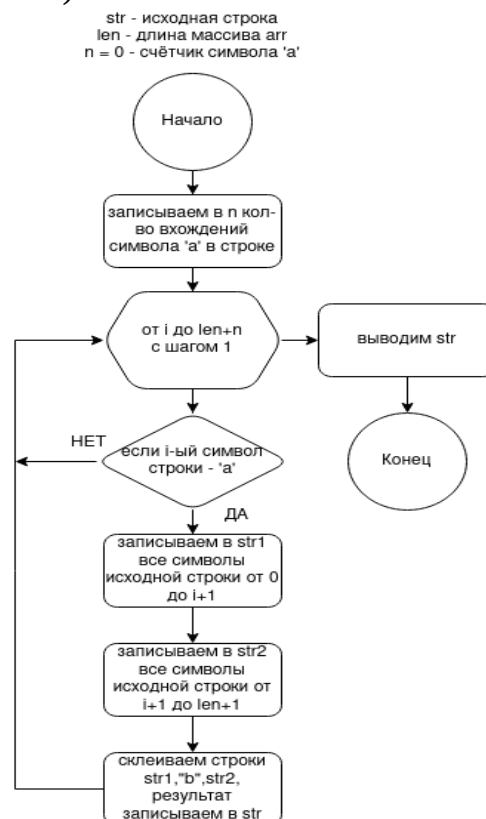


Рисунок 5 - Блок-схема к заданию 3.

### Результаты работы (Рисунок 6)

```
Input file name:
line.txt
ab_bb_ababab_AA-Aab

** Process exited - Return Code: 0 **
```

Рисунок 6 - Скриншот результата работы программы к заданию 3 на консоли.

### Задание 4

Вывести в алфавитном порядке все слова, содержащие наибольшее количество гласных букв; найти все слова, в которые буква «а» входит не менее двух раз.

### Блок-схема (Рисунок 7)

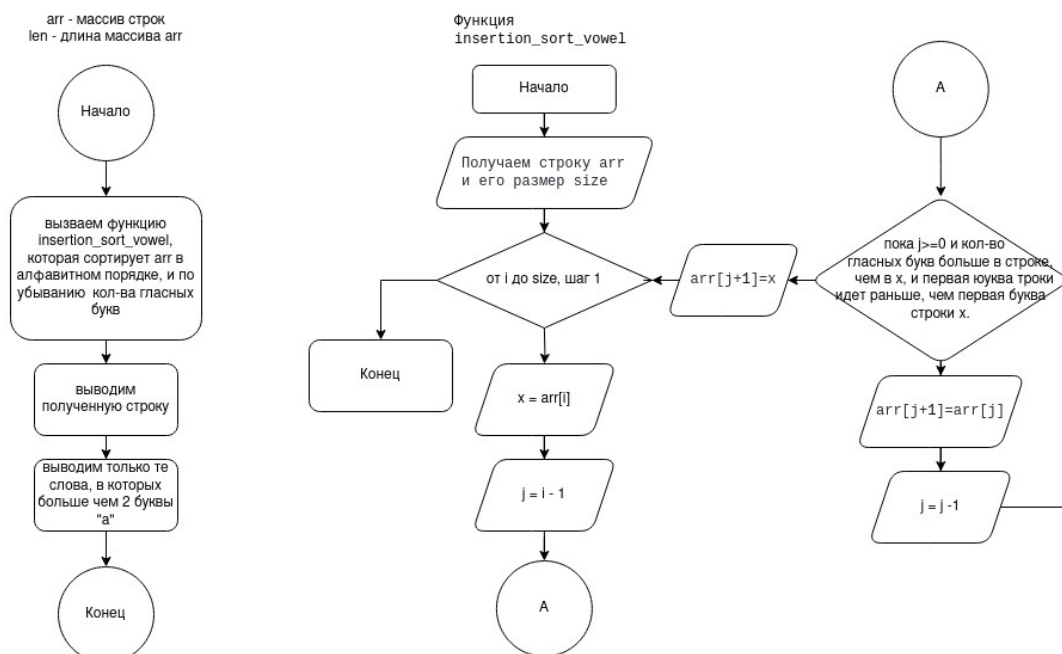


Рисунок 7 - Блок-схема к заданию 4.

Текст программы (Листинг 4, Приложение А)

**Результаты работы (Рисунок 8)**

```
Input file name:
line.txt
    abandon abroad dragon
World with count('a') >=2:
abandon abroad

** Process exited - Return Code: 0 **
```

Рисунок 8 - Скриншот результата работы программы к заданию 4 на консоли.

**Задание 5**

Строка состоит из слов, разделенных одним или несколькими пробелами. Переставьте слова по убыванию их длин.

## Блок схема (Рисунок 9)

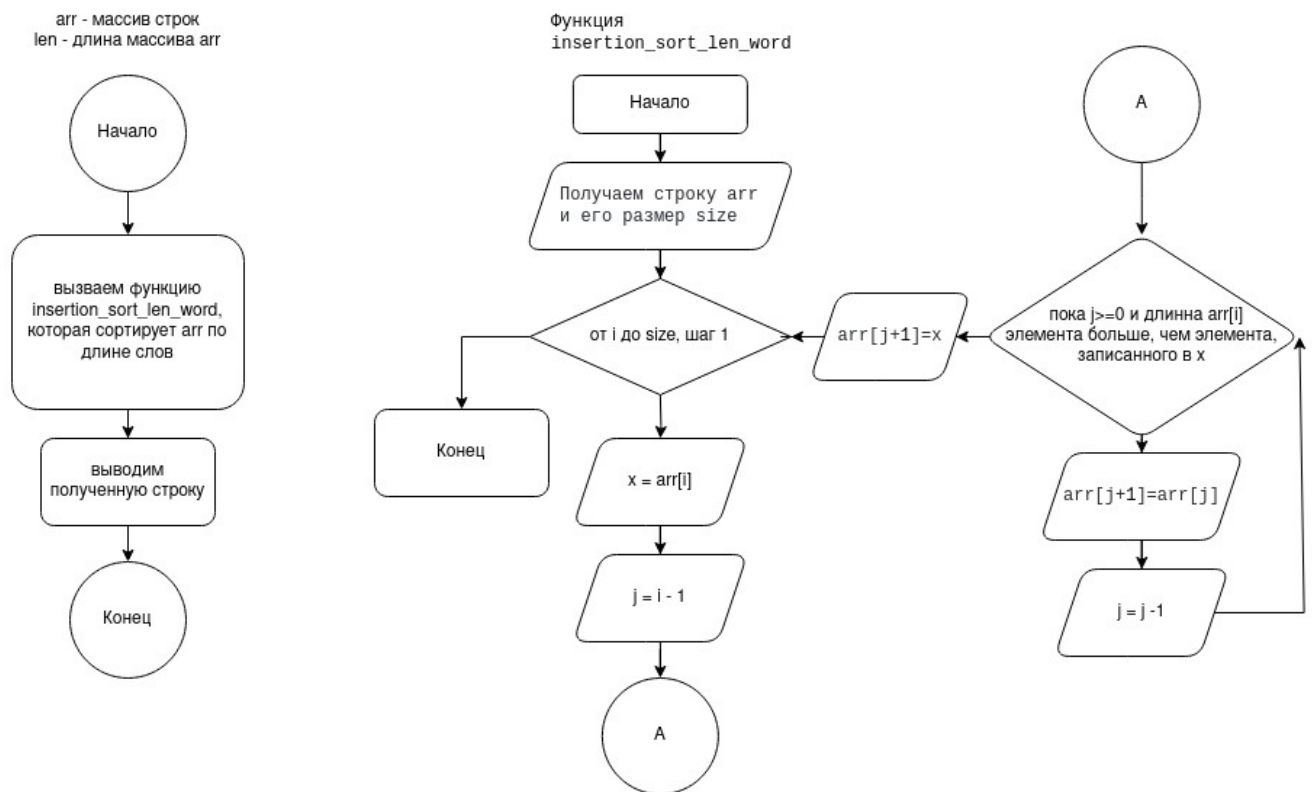


Рисунок 9 - Блок-схема к заданию 5.

Текст программы (Листинг 5, Приложение А)

**Результаты работы** (Рисунок 10)

```

Input file name:
line.txt
pineapple amazing family apple god

** Process exited - Return Code: 0 **
  
```

Рисунок 10 - Скриншот результата работы программы к заданию 5 на консоли.

## Задание 6

Дан текст. Некоторые его фрагменты выделены группами символов `##`.  
Заменить выделение группами символов `'<'` и `'>'`). Пример: 'Это  
`##`тестовый пример`##` для задачи `##`на`##` строки' преобразуется в 'Это  
<тестовый> пример для задачи <на> строки'.

### Блок схема (Рисунок 11)

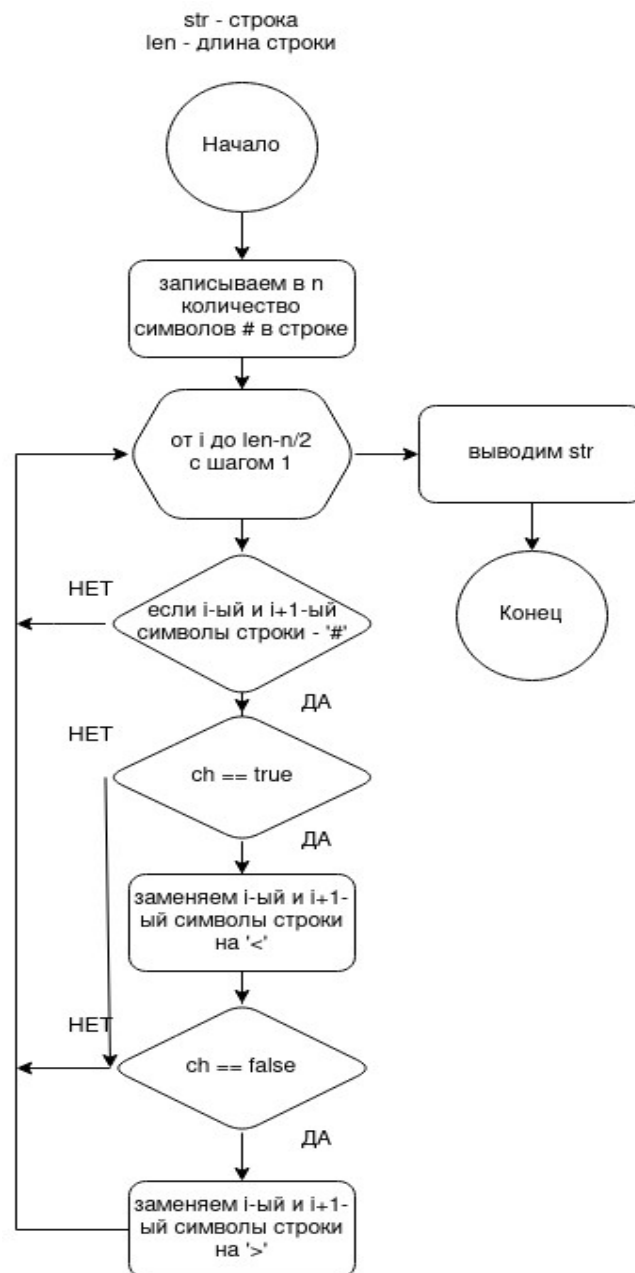


Рисунок 11 - Блок-схема к заданию 6.



Текст программы (Листинг 6, Приложение А)

**Результаты работы** (Рисунок 12)

```
Input file name:  
line.txt  
my <code> is <great> .<!>  
  
** Process exited - Return Code: 0 **
```

Рисунок 12 - Скриншот результата работы программы к заданию 6 на консоли.

### **Задание 7**

Даны две строки *s* и *w*, вывести строку *x* максимальной длины, состоящую из букв, таких, что существует перестановка *x*, являющаяся подстрокой перестановки *s* и одновременно являющаяся подстрокой перестановки *w*.

### Блок схема (Рисунок 13)

str1, str2-первая и вторая строка соответственно.  
str3 - пустая строка

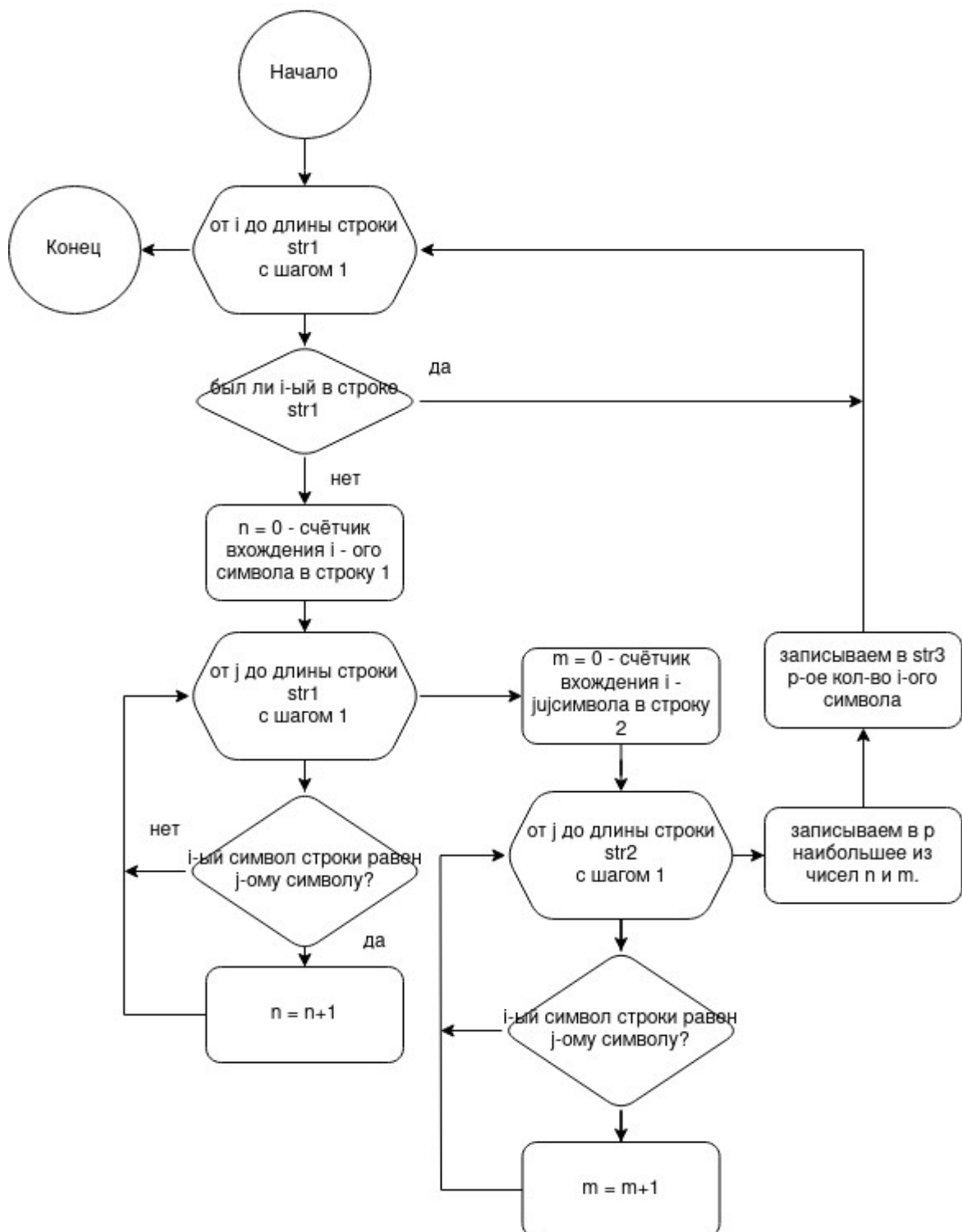


Рисунок 13 - Блок-схема к заданию 7.

Текст программы (Листинг 7, Приложение А)

**Результаты работы** (Рисунок 14)

```
Input file name:
line.txt
abccf pplets

** Process exited - Return Code: 0 **
```

Рисунок 14 - Скриншот результата работы программы к заданию 7 на консоли.

## Приложение А

Листинг 1 – Программа к заданию 1.

```
1. #include <iostream>
2. #include <fstream>
3.
4. int main()
5. {
6.     std::string file_name; std::cout << "Input file name: "; std::cin >>
        file_name;
7.
8.     std::ifstream file;
9.     file.open(file_name);
10.
11.    if(file.is_open()){
12.        std::string str;
13.        getline(file,str);
14.        const int len=str.length();
```

```

15.      char arr[len];
16.      for(int i=0;i<len;i++){
17.          arr[i]=str[i];
18.      }
19.      int n = 0;
20.      for (int i = 0; i <= len-2; i++) {
21.          if (arr[i] == 'a' && arr[i + 1] == 'b' && arr[i + 2] == 'a') n++;
22.      }
23.      std::cout << n;
24.  }else{
25.      std::cout << "ERROR" << std::endl;
26.  }
27.  return 0;
28.}

```

## Листинг 2 – Программа к заданию 2.

```

1. #include <iostream>
2. #include <fstream>
3.
4. int main()
5. {
6.     std::string file_name; std::cout << "Input file name: "; std::cin >>
        file_name;
7.
8.     std::ifstream file;
9.     file.open(file_name);
10.
11.     if(file.is_open()){
12.
13.         std::string str;

```

```

14.     getline(file,str);
15.     const int len=str.length();
16.     char arr[len];
17.     for(int i=0;i<len;i++){
18.         arr[i]=str[i];
19.     }
20.     int n = 0;
21.     for (int i = 0; i <= len; i++) {
22.         if (arr[i] == 'a') arr[i] = 'A';
23.         if (arr[i] == 'b') arr[i] = 'B';
24.     }
25.     std::cout << arr;
26. }else{
27.     std::cout << "ERROR"<<std::endl;
28. }
29.     return 0;
30.}

```

Листинг 3 – Программа к заданию 3.

```

1. #include <iostream>
2. #include <fstream>
3.
4. int main()
5. {
6.     std::string file_name; std::cout << "Input file name: "; std::cin >>
        file_name;
7.     std::ifstream file;
8.     file.open(file_name);
9.
10.    if(file.is_open()){

```

```

11.
12.     std::string str;
13.     getline(file,str);
14.     const int len = str.length();
15.     int n = 0;
16.
17.     for (int i = 0; i < len; i++) {
18.         if (str.at(i) == 'a') n++;
19.     }
20.
21.     for(int i = 0; i < len+n; i++) {
22.         if (str.at(i) == 'a') {
23.             std::string str1(str, 0, i+1);
24.             std::string str2(str, i+1, len+1);
25.             str.clear();
26.             str.append(str1).append("b").append(str2);
27.         }
28.     }
29.
30.     std::cout << str;
31. }
32. file.close();
33.
34.     return 0;
35. }

```

Листинг 4 – Программа к заданию 4.

```

1. #include <iostream>
2. #include <fstream>
3. #include <string>

```

```
4.
5. int word_count(std::string file_name){
6.     std::ifstream file;
7.     file.open(file_name);
8.
9.     if(file.is_open()){
10.         int count = 1;
11.         std::string str;
12.         getline(file,str);
13.
14.         for (int i = 0; i < str.size(); i++)
15.             if (str[i] == ' ') count++;
16.
17.         file.close();
18.         return count;
19.     }
20.     else{
21.         file.close();
22.     }
23.     return 0;
24. }
25.
26. int vowel_count(std::string str){
27.     int n=0;
28.     for(int i=0; i<=str.length();i++){
29.         if((int(str[i])==97||int(str[i])==101||int(str[i])==105||int(str[i])==111||
            int(str[i])==117||int(str[i])==121) n++;
30.     }
```

```

31.  return n;
32.}
33.
34.int a_count(std::string str){
35.  int n=0;
36.  for(int i=0; i<=str.length();i++){
37.      if(int(str[i])==97) n++;
38.  }
39.  return n;
40.}
41.
42.void insertion_sort_vowel(std::string *arr, int size){
43.    for(int i=0;i<size;i++){
44.        std::string x=arr[i];
45.        int j=i-1;
46.        while(j>=0 && vowel_count(arr[j])>=vowel_count(x)
&& int(arr[j][0])<=int(x[0])){
47.            arr[j+1]=arr[j];
48.            j--;
49.        }
50.        arr[j+1]=x;
51.    }
52.}
53.
54.
55.
56.int main()
57.{

```



```
58.  std::string file_name; std::cout << "Input file name: "; std::cin >> file_name;
59.  int words = word_count(file_name);
60.
61.  std::string arr[words];
62.
63.  std::ifstream file;
64.  file.open(file_name);
65.
66.  if(file.is_open()){
67.
68.      for(int i=0; i<=words;i++){
69.          file >> arr[i];
70.      }
71.
72.      insertion_sort_vowel(arr, words);
73.
74.      for(int i=words; i>=0;i--){
75.          std::cout << arr[i] << " ";
76.      }
77.      std::cout << std::endl;
78.      std::cout << "World with count('a') >=2: " << std::endl;
79.      for(int i=words; i>=0;i--){
80.          if(a_count(arr[i])>=2) std::cout<<arr[i]<<" ";
81.      }
82.  }else{
83.      std::cout << "ERROR";
84.  }
```

```
85.  
86.  return 0;  
87.}
```

Листинг 5 – Программа к заданию 5.

```
1. #include <iostream>  
2. #include <fstream>  
3. #include <string>  
4.  
5. int word_count(std::string file_name){  
6.     std::ifstream file;  
7.     file.open(file_name);  
8.  
9.     if(file.is_open()){  
10.         int count = 1;  
11.         std::string str;  
12.         getline(file,str);  
13.  
14.         for (int i = 0; i < str.size(); i++)  
15.             if (str[i] == ' ') count++;  
16.  
17.         file.close();  
18.         return count;  
19.     }  
20.     else{  
21.         file.close();  
22.     }
```

```

23. return 0;
24.}
25.
26.void insertion_sort_len_word(std::string *arr, int size){
27.    for(int i=0;i<size;i++){
28.        std::string x=arr[i];
29.        int j=i-1;
30.        while(j>=0 && arr[j].length()>=x.length()){
31.            arr[j+1]=arr[j];
32.            j--;
33.        }
34.        arr[j+1]=x;
35.    }
36.}
37.
38.int main()
39.{
40.    std::string file_name; std::cout << "Input file name: "; std::cin >> file_name;
41.    int words = word_count(file_name);
42.
43.    std::string arr[words];
44.
45.    std::ifstream file;
46.    file.open(file_name);
47.
48.    if(file.is_open()){
49.

```

```

50.     for(int i=0; i<words;i++){
51.         file >> arr[i];
52.     }
53.
54.     insertion_sort_len_word(arr, words);
55.
56.     for(int i=words; i>=0;i--){
57.         std::cout << arr[i] << " ";
58.     }
59.
60.     return 0;
61. }else{
62.     file.close();
63. }
64. return 0;
65.}

```

Листинг 6 – Программа к заданию 6.

```

1. #include <iostream>
2. #include <fstream>
3. #include <string>
4.
5. int main()
6. {
7.     std::string file_name; std::cout << "Input file name: "; std::cin >> file_name;
8.
9.     std::ifstream file;

```

```
10. file.open(file_name);
11.
12. if(file.is_open()){
13.
14.     std::string str;
15.     getline(file,str);
16.     const int length = str.length();
17.
18.     int n = 0;
19.     for (int i = 0; i < length; i++) {
20.         if (str.at(i) == '#') n++;
21.     }
22.     bool ch = false;
23.     for(int i = 0; i < length-n/2; i++) {
24.         if (str.at(i) == '#' && str.at(i+1) == '#') {
25.             if(!ch){
26.                 std::string str1(str, 0, i);
27.                 std::string str2(str, i+2, length+1);
28.                 str.clear();
29.                 str.append(str1).append("<").append(str2);
30.                 ch = true;
31.                 continue;
32.             }
33.             if(ch){
34.                 std::string str1(str, 0, i);
35.                 std::string str2(str, i+2, length+1);
36.                 str.clear();
```

```

37.         str.append(str1).append(">").append(str2);
38.         ch = false;
39.         continue;
40.     }
41. }
42. }
43.
44.     std::cout << str;
45. }else{
46.     std::cout<<"ERROR"<<std::endl;
47. }
48. return 0;
49.}

```

Листинг 7 – Программа к заданию 7.

```

1. #include <iostream>
2. #include <fstream>
3. #include <string>
4.
5. std::string str4 = ""; // символы которые уже были.
6. bool check_symbol(char symbol, std::string str);
7.
8. int main()
9. {
10.
11.     std::string file_name; std::cout << "Input file name: "; std::cin >> file_name;
12.     std::ifstream file;

```

```
13. file.open(file_name);
14.
15. if(file.is_open()){
16.
17.     std::string str1;
18.     std::string str2;
19.     getline(file,str1);
20.     getline(file,str2);
21.
22.     const int len1 = str1.length();
23.     const int len2 = str2.length();
24.
25.     std::string str3 = ""; // конечная строка.
26.
27.
28.     for(int i=0;i<len1;i++){
29.         std::string s(str1, i, 1);
30.         if(check_symbol(str1.at(i), str1)){//проверяем не был ли уже символ
31.             str4.append(s);
32.             int n = 0; // счётчик вхождения символа в строку 1.
33.             for(int j=0;j<str1.length();j++){
34.                 if(str1.at(i) == str1.at(j)) n++;
35.             }
36.             int m = 0; // счётчик вхождения символа в строку 2.
37.             for(int j=0;j<str2.length();j++){
38.                 if(str2.at(j) == str1.at(i)) m++;
39.             }
```

```
40.         int p = (n>m)? m:n;
41.         for(int j=1;j<=p;j++) str3.append(s);
42.     }
43. }
44.
45.     std::cout << str3;
46. }else{
47.     std::cout << "ERROR" << std::endl;
48. }
49.
50.     return 0;
51.}
52.
53.bool check_symbol(char symbol, std::string str){
54.    for(int i=0;i<str4.length();i++){
55.        if(str4.at(i) == symbol) return false;
56.    }
57.    return true;
58.}
```