

## Задание № 5

# Программа обработки строк с использованием функций

### 1. Формулировка задания

Разработать программу для обработки строковой информации, вводимой с клавиатуры. Длина исходной строки ограничена 255 символами. Строка состоит из **слов**, разделителем слов является один или несколько пробелов. В начале и в конце строки может содержаться ноль или более пробелов.

### 2. Общие требования к разрабатываемой программе:

**Разработанная программа должна обеспечить:**

- вывод на экран информации об авторе программы;
- ввод исходной строки пользователем с клавиатуры в диалоговом режиме;
- разбиение исходной строки на слова (словом считать последовательности символов, начинающиеся и заканчивающиеся не символом "пробел" и не содержащие внутри символов "пробел"), и помещение выделенных слов в массив;
- вывод на экран исходного массива слов, выделенных в результате разбора исходной строки, а также результатов реализации алгоритма, предусмотренного вариантом задания;

Все основные этапы реализации программы, – ввод исходной строки с клавиатуры, разбиение исходной строки на слова, необходимые преобразования в соответствии с вариантом задания, вывод результатов работы программы, – оформить **функциями языка C++**, обеспечив передачу необходимых данных через аппарат параметров.

### **3. Варианты заданий разрабатываемой программы:**

#### **Вариант 1**

Разработать программу, выводящую на экран слова из исходной строки, содержащие определенное количество символов. "Длина" слова вводится с клавиатуры.

#### **Вариант 2**

Проверить, являются ли слова, из которых состоит исходная строка, правильными идентификаторами языка C++. Правильные идентификаторы вывести на экран.

#### **Вариант 3**

В исходной строке определить самое короткое и самое длинное слова и вывести их на экран.

#### **Вариант 4**

Разработать программу, которая проверяет, являются ли слова, из которых состоит исходная строка целыми десятичными константами C++. Правильные десятичные константы вывести на экран.

#### **Вариант 5**

Выбрать из исходной строки слова, начинающиеся и оканчивающиеся одним и тем же символом. Вывести на экран выбранные слова, а затем – все остальные.

#### **Вариант 6**

В исходной строке определить слова, начинающиеся с определенного символа, вывести их на экран (символ вводится с клавиатуры).

#### **Вариант 7**

Вывести на экран из исходной строки слова, которые содержат не менее  $N$  вхождений некоторого символа. Количество вхождений  $N$  и изображение искомого символа вводятся с клавиатуры.

#### **Вариант 8**

Подсчитать общее количество символов, которое содержат нечетные слова исходной строки, вывести найденные слова и количество символов в них на экран.

#### **Вариант 9**

Составить программу, которая проверяет, являются ли слова, входящие в состав исходной строки, вещественными константами с фиксированной точкой. Правильные константы вывести на экран.

#### **Вариант 10**

В исходной строке определить самое длинное слово и выполнить его реверс (первая буква меняется местами с последней, вторая – с предпоследней и т.д.). Полученное слово вывести на экран.

### **Вариант 11**

Упорядочить слова исходной строки по возрастанию их длины. На экран вывести строку в упорядоченном виде.

### **Вариант 12**

Разработать программу, которая выводит на экран слова, являющиеся положительными шестнадцатеричными константами C++.

### **Вариант 13**

Разработать программу, которая выведет на экран слова, начинающиеся с символов букв по убыванию их длины.

### **Вариант 14**

Разработать программу, преобразующую слово, номер которого введён с клавиатуры (нумерация слов в строке начинается с единицы). Алгоритм преобразования: поменять в выбранном слове регистр символов. Исходное и преобразованное слова вывести на экран.

### **Вариант 15**

Разработать программу, удаляющую из исходной строки слова, длина которых не превышает некоторого значения  $L$ , вводимого с клавиатуры. Полученную в результате строку вывести на экран.

### **Вариант 16**

Отсортировать слова исходной строки по убыванию их длины, используя для этого алгоритм сортировки вставками. Отсортированную последовательность слов вывести на экран.

### **Вариант 17**

Разработать программу, проверяющую имеются ли в исходной строке отдельные слова, заключенные в скобки ( ), [ ], { }. Вывести на экран слово или слова, содержащиеся в скобках с ненарушенным балансом парности. Вложенность скобок отсутствует.

### **Вариант 18**

Разработать программу, которая подсчитывает количество слов, в которых встречается введенная с клавиатуры подстрока. Полученное число и слова, в которых встречается данная подстрока вывести на экран.

### **Вариант 19**

Разработать программу, рассчитывающую среднее количество символов в словах исходной строки. На экран вывести слова, длина которых превышает полученное среднее значение.

### **Вариант 20**

Разработать программу, отыскивающую в исходной строке слова, являющиеся целыми положительными десятичными числами. Слова, являющиеся целыми десятичными числами преобразовать в числа и вывести на экран их сумму.

### **Вариант 21**

Составить программу, переворачивающую слова в предложении. Например, "Мама мыла раму" – "амаМ алым умар". Полученную строку вывести на экран.

### **Вариант 22**

Разработать программу, формирующую аббревиатуру исходной строки. Например, "Российский технологический университет" – "РТУ". Исходную строку (пословно) и соответствующую ей аббревиатуру вывести на экран.

### **Вариант 23**

Составить программу, которая объединяет в единое слово те слова из исходной строки, номера которых введены с клавиатуры (нумерация слов в строке начинается с единицы). Полученное слово вывести на экран.

### **Вариант 24**

В исходной строке определить слова, заканчивающиеся символом цифры, вывести их на экран.

### **Вариант 25**

Разработать программу, которая выведет на экран слово (или слова) исходной строки наименьшей длины.

### **Вариант 26**

Разработать программу, которая расставляет слова во введенном предложении по алгоритму: последняя буква предыдущего слова совпадает с первой следующего. Полученную строку вывести на экран.

### **Вариант 27**

Разработать программу, которая будет редактировать введенное предложение, удаляя из него повторяющиеся слова. Строку без повторов вывести на экран.

### **Вариант 28**

Разработать программу, которая определяет количество повторений каждого слова в предложении. Информацию о количестве повторений вывести на экран.

### **Вариант 29**

Разработать программу, которая будет редактировать введенное предложение, удаляя из него слова с одинаковым количеством букв. Полученную строку вывести на экран.

### **Вариант 30**

Найти в строке самое короткое и самое длинное слова. В результирующей строке поменять их местами. Полученную строку вывести на экран.