

## Лабораторная работа №5

### Строки

**Пример 1.** Сформулировать результат выполнения функции, написать вызов функции.

```
void f(char c[]){
    int j=0, comm=0; // ненулевое значение переменной comm -
                     // признак нахождения внутри закомментированного блока
    for(int i=0; c[i]!='\0'; i++){
        if (c[i]=='*' && c[i+1]=='/'){comm--; i++; continue;}
        if (c[i]=='/' && c[i+1]=='*'){comm++; i++; continue;}
        if (comm == 0) c[j++]=c[i];
    }
    c[j]= '\0';
}
```

Функция осуществляет изменение обрабатываемой строки, удаляя из неё все символы, находящиеся между ограничителями /\* и \*/.

```
int main(){
    char c[]="djkkj", c1[]="j/*kjk*/j", c2[]="djkkj/*kj";
    printf("%s\n",c);      f(c);      printf("%s\n\n",c);
    printf("%s\n",c1);     f(c1);     printf("%s\n\n",c1);
    printf("%s\n",c2);     f(c2);     printf("%s\n",c2);
    return 0;
}
```

**Пример 2.** Дана строка, в которой между словами может быть более одного пробела. Написать функцию, копирующую в другую строку все символы первой строки, кроме лишних пробелов.

Первую строку будем просматривать посимвольно, равномерно двигаясь по ней с помощью индекса *i*. Когда текущий символ первой строки – пробел, во вторую строку ничего не вносится. Во вторую строку копируется символ, когда он не пробел. При этом, если перед текущим символом был пробел, тогда во вторую строку сначала добавляется пробел, потом текущий символ. По завершению цикла вторую строку оканчиваем символом конца строки. Заметим, что индекс *j*, с помощью которого движемся по второй строке, меняется не равномерно от итерации к итерации, а только в моменты добавления очередного символа, поэтому длина второй строки будет меньше длины первой.

```
void f (char c1[], char c2[]){
    int j=0;
    for(int i=0; c1[i] != '\0'; i++){
        if (c1[i] != ' '){
            if (i != 0 && c1[i-1] == ' '){
                c2[j++] = ' ';
            }
            c2[j++] = c1[i];
        }
    }
    c2[j] = '\0';
}
```

**Пример 3.** Дана строка. Написать функцию, которая ищет первое вхождение заданного фрагмента строки.

Функция должна получать указатель на начало строки, продвигать его к началу обнаруженного фрагмента и возвращать его значение в качестве результата. Для проверки на наличие фрагмента нужно при совпадении элемента строки с первым символом фрагмента далее попарно сравнивать (до обнаружения первого расхождения) все элементы строки с элементами фрагмента (информация о котором в функцию передается с помощью указателя, настроенного на начало). Если фрагмент кончится, то это означает, что все его символы стоят в нашей строке и нужно вернуть указатель в точку вызова функции.

```
char* f (char *p, char *q){
    int i;
    for (; *p != '\0'; p++){
        for (i=0; (q[i] != '\0')&&(p[i] == q[i]); i++);
        if (q[i] == '\0')
            return p;
    }
    return 0;
}
```

### Задание 5.

1. Исходный текст находится в текстовом файле, имя которого вводится с клавиатуры.
2. Вывод результата осуществляется одновременно в текстовый файл, имя которого вводится с клавиатуры, и на экран монитора.

- 5.1.** Дан текст из строчных русских букв, за которым следует точка. Напечатать этот текст заглавными буквами.
- 5.2.** Дан непустой текст из заглавных русских букв, за которым следует точка. Определить упорядочены ли эти буквы по алфавиту. В случае неупорядоченности указать позицию первого символа, нарушающего алфавитный порядок.
- 5.3.** Напечатать в алфавитном порядке все различные строчные русские буквы, входящие в заданный текст.
- 5.4.** Дана строка латинских символов. Напечатать эту строку, предварительно заменив все вхождения «abc» на «def».
- 5.5.** Дана строка латинских символов. Напечатать эту строку, предварительно удалив первое вхождение «w», если такое есть (образовавшуюся «дыру» заполнить последующими буквами, а в конец добавить пробел).
- 5.6.** Дана строка латинских символов. Напечатать эту строку, предварительно заменив на «ks» первое вхождение «x», если оно есть.
- 5.7.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать эту же последовательность слов, но в обратном порядке.
- 5.8.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать эту же последовательность слов, удалив из нее повторно входящие слова.
- 5.9.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все слова, которые встречаются в последовательности по одному разу.
- 5.10.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все различные слова, указав для каждого из них число его вхождений в последовательность.
- 5.11.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – запятая, за последним словом точка. Напечатать все слова в алфавитном порядке разделенные пробелами без запятых.
- 5.12.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова и являются симметричными.
- 5.13.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова и первая буква слова входит в него еще ровно один раз.
- 5.14.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова и буквы слова упорядочены по алфавиту.
- 5.15.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова и в слове нет повторяющихся букв.
- 5.16.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно перенеся первую букву в конец.
- 5.17.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно перенеся последнюю букву в начало.
- 5.18.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно удалив из слова первую букву.
- 5.19.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно удалив из слова последнюю букву.
- 5.20.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова, предварительно удалив из слова все последующие вхождения первой буквы.
- 5.21.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно удалив из слова все предыдущие вхождения последней буквы.
- 5.22.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно оставив в слове только первые вхождения каждой буквы.
- 5.23.** Дана последовательность, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 строчных латинских букв; между соседними словами – не менее одного пробела, за последним словом точка. Напечатать слова, которые отличны от последнего слова предварительно удалив из слова среднюю букву, если оно нечетной длины.
- 5.24.** Дан текст из заглавных латинских букв, за которым следует пробел. Определить, является ли этот текст правильной записью римскими цифрами целого числа от 1 до 999, и, если является, распечатать это число арабскими цифрами.
- 5.25.** Задано шестнадцатеричное число. Напечатать таблицу умножения в шестнадцатеричной системе счисления от 1 до данного числа.