

Глава 8. Использование символов и строк.

В языке программирования C функции для работы со строками объявляются в заголовочном файле *string.h*, а функции работы с символами в файле *ctype.h*. Нужно не забывать подключать их.

Следует обратить внимание на то, что строкой в C считается любая последовательность символов, оканчивающаяся нулевым кодом. Поэтому функциям можно передавать адрес любого символа строки, и они будут считать строкой оставшуюся часть. Это свойство можно эффективно использовать. Для примера вставим одну строку в середину другой:

```
char s1[20] = "one three", s2[20] = "two";
strcpy(s2+3, s1+3);
strcpy(s1+4, s2);
puts(s1);
```

Здесь сначала во вторую строку копируется конец первой, получается "two three". Затем в первую строку, минуя ее начало, копируется вторая.

Еще одной интересной функцией является *strtok()*. С ее помощью можно разбить строку на отдельные части (лексемы). Однако, эта функция портит исходную строку, вставляя в неё терминальные нули. Поэтому, если исходная строка нужна в последующем коде, то необходимо сделать её копию.

Рассмотрим пример:

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[200],str2[200];
    fgets(str,200,stdin); // ввод строки из консоли
    strcpy(str,str2); // делаем копию для работы с strtok
    // Объявляем с запасом массив из 30 слов длиной не более 19 символов
    char words[30][20];
    char * pch = strtok (str, " ,.-"); // получаем первое слово
    // во втором параметре указаны разделители слов (пробел, запятая, точка, тире)
    int n=0;
    strcpy(words[n++],pch); // копируем первое слово

    // пока есть очередное слово и место в массиве words:
    while( (pch = strtok (NULL, " ,.-")) != NULL && n<30)
        strcpy(words[n++],pch); // копируем очередное слово
    for(int i=0;i<n;i++)
        printf("%s %d\n",words[i],strlen(words[i])); // выводим слова и их длины
    return 0;
}
```

При первом вызове *strtok()* в функцию передается указатель на первый символ массива и строка-разделитель. После этого вызова массив *str* изменяется, в нем остается только первое слово, также функция возвращает указатель на это слово, который присваивается *pch*. Остаток массива потерялся, но внутри *strtok()* сохраняется указатель на этот остаток. Когда в качестве адреса передается NULL, то функция работает с этим остатком.

Для подсчета количества одинаковых символов строки можно использовать их коды для индексирования элементов массива, в которых будет храниться количество соответствующих символов. Обратите внимание, что тип *char* – это знаковый целочисленный тип в диапазоне от -128 до +127. Поэтому, для индексации массива надо использовать *unsigned char* (диапазон от 0 до 255).

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void CountChars(unsigned char* str,int* freq)
{
    int i = 0;
    for(i=0;i<256;i++)
        freq[i]=0; // обнуляем массив
    // подсчет количества символов равных str[i]:
    for(i=0;i<str[i]!=0;i++)
        freq[str[i]]++;
}
void Print(int* freq)
{
    int i;
    for(i=0;i<256;i++)
```

```

        if(freq[i]!=0) // если символ встречался в строке
// вывод кода символа и сколько раз встретился:
        printf("%c %d\n",i,freq[i]);
}
int main()
{
    unsigned char myString[100];
    int f[256];

    system("chcp 1251"); // установка русской кодировки в консоли

    printf("Введите строку:\n");
    fgets((char*)myString,100,stdin);
    CountChars(myString,f);
    Print(f);
    getchar();
    return 0;
}

```

Здесь функция `CountChars()` выполняет подсчет встречаемости символов, а функция `Print()` выводит результат. Обратите внимание на выражение `freq[str[i]]++`, в котором код символа `str[i]` является индексом для массива `freq`. Это позволяет подсчитать количество символов, имеющих данный код.

Варианты задач для решения

В задании необходимо написать функцию соответствующего варианта и в функции `main()` протестировать ее, вводя необходимые строки из файла, а дополнительные данные из консоли, выводя результат на экран.

1. Вариант Написать функцию, которая удаляет из строки все знаки пунктуации. Для проверки знаков использовать библиотечную функцию <code>ispunct()</code> .
2. Вариант Написать функцию, которая проверяет, является ли строка целым десятичным числом с опциональным знаком. Для проверки цифр использовать библиотечную функцию <code>isdigit()</code> .
3. Вариант Написать функцию, которая подсчитывает количество гласных букв в строке используя массив гласных букв и функцию <code>strcspn()</code> .
4. Вариант Написать функцию, которая подсчитывает, сколько раз в строке встречается заданная буква используя функцию <code>strchr()</code> и после этого удалить ее из строки.
5. Вариант Написать функцию, которая заменяет в строке <code>s</code> все вхождения подстроки <code>s1</code> на подстроку <code>s2</code> используя функцию <code>strstr()</code> .
6. Вариант Написать функцию, которая находит все слова строки, отличные от последнего, используя функции <code>strtok()</code> и <code>strcmp()</code> .
7. Вариант Написать функцию, которая удаляет из строки слова начинающиеся на указанную букву используя функцию <code>strtok()</code> .
8. Вариант Написать функцию, которая определяет количество различных слов строки используя функции <code>strtok()</code> и <code>strcmp()</code> .
9. Вариант Написать функцию, которая вычисляет количество вхождений переданного символа в строке используя функцию <code>strchr()</code> .
10. Вариант Написать функцию, проверяющую, что строка является палиндромом используя функции <code>strrev()</code> и <code>strcmp()</code> .
11. Вариант Написать функцию, вычисляющую количество цифр в строке используя функцию <code>isdigit()</code> .

12. Вариант Написать функцию, вычисляющую длину слов в строке используя функцию <i>strtok()</i> .
13. Вариант Написать функцию, находящую позицию самого последнего вхождения указанного символа в строке используя функцию <i>strchr()</i> и сравнить с результатом функции <i>strrchr()</i> .
14. Вариант Написать функцию сравнения двух строк без учета регистра используя функцию <i>_stricmp()</i> . Функция возвращает 0, если строки одинаковые и 1, если разные. Не забудьте задать русскую локаль вызовом функции <i>setlocale()</i> , если строки содержат кириллицу.
15. Вариант Написать функцию, инвертирующую все слова переданной строки.
16. Вариант Написать функцию, которая принимает на вход строку и два символа. Функция заменяет все вхождения первого символа в строке на второй символ используя функцию <i>strchr()</i> .
17. Вариант Написать функцию, проверяющую, что переданную строку можно разбить на две одинаковые подстроки используя функцию <i>strncmp()</i> .
18. Вариант Написать функцию, которая записывает строковое представление числа в массив. Число и массив передаются ей в качестве параметров. Сравнить ее с функцией <i>itoa()</i> .
19. Вариант Написать функцию, которая находит самое короткое слово в строке используя функцию <i>strtok()</i> . Слова отделяются пробелами.
20. Вариант Написать функцию, которая выполняет выравнивание строки, введенной пользователем, до заданной ширины используя функцию <i>strtok()</i> и добавляя необходимое количество пробелов.
21. Вариант Написать функцию, выводящую на экран слова, которые одновременно содержатся в каждой из двух заданных строк используя функции <i>strtok()</i> и <i>strcmp()</i> .
22. Вариант Написать функцию, которая используя функции <i>strtok()</i> и <i>strcmp()</i> выводит на экран слова, которые являются общими для трех строк, а также для каждой пары строк.
23. Вариант Написать функцию, которая используя функции <i>strtok()</i> и <i>strcmp()</i> находит слова, которые повторяются два или более раз и выводит их с указанием числа повторений. В случае отсутствия таких слов выводится соответствующее сообщение.
24. Вариант Написать функцию, которая используя функции <i>strtok()</i> и <i>strchr()</i> определяет количество слов в заданной строке, содержащих заданную букву. Вывести число слов в строке, содержащих русские буквы 'а' и 'р'. В случае отсутствия таких слов вывести соответствующее сообщение.
25. Вариант Написать функцию, которая используя функции <i>strtok()</i> и <i>strchr()</i> выводит все слова заданной строки, в которые заданная буква входит не менее K раз. В случае отсутствия таких слов выводится соответствующее сообщение.
26. Вариант Написать функцию, которая используя функции <i>strtok()</i> и <i>strcmp()</i> вычисляет количество разных слов в заданной строке.
27. Вариант Написать функцию, которая преобразует целое число(int) в строку, представляющую число в указанной системе счисления (от 2 до 10).
28. Вариант Написать функцию возвращающую максимальное количество подряд идущих произвольных символов в строке. Например, для строки "asdaajhys as asaaaafg r" нужно вернуть 4.

29. Вариант

Написать функцию, которая возвращает символ, встречающийся в строке чаще всего.

30. Вариант

Написать программу, которая выводит на экран слово "yes", если из букв введенной строки X можно составить введенную строку Y, при условии, что каждую букву строки X можно использовать один раз. Если это невозможно сделать, вывести "no". Подсказка: можно сравнить частотные таблицы первой и второй строки.