

## Тема : знакомство с С (Си)

### Стандартные строковые функции – упражнения

Содержание: [строковые функции](#) [упражнение 1](#) [упражнение 2](#)

Для работы с символьными строками в языке С имеется набор стандартных функций. Описания строковых функций находятся в заголовочном файле **string.h**. Этот файл надо включить в программу директивой **#include**.

Рассмотрим некоторые строковые функции.

Функция **strcat** добавляет одну строку к другой. Функция объявлена следующим образом:

```
char *strcat(char *dest, const char *src);
```

Функция добавляет копию строки *src* в конец *dest* и возвращает указатель на результирующую строку. Надо предусмотреть, чтобы новая строка поместилась бы в массив.

Функция **strchr** ищет в строке первое вхождение заданного символа. Функция объявлена следующим образом:

```
char *strchr(const char *s, int c);
```

Функция ищет первое вхождение символа *c* в строку *s*. Она возвращает указатель на первое вхождение символа *c* в *s*; если *c* не обнаружен в *s*, то *strchr* возвращает **NULL**.

Функция **strcmp** сравнивает одну строку с другой. Функция объявлена следующим образом:

```
int strcmp(const char *s1, const char *s2);
```

Функция *strcmp* осуществляет сравнение строк *s1* и *s2*, начиная с первого символа каждой строки, до тех пор, пока очередные соответствующие символы в строках не будут различны или пока не будут достигнуты концы строк. Функция возвращает отрицательное значение, если *s1* меньше чем *s2*, ноль, если *s1* равна *s2*, и положительное значение, если *s1* больше чем *s2*.

Функция **strcpy** копирует одну строку в другую. Функция объявлена следующим образом:

```
char *strcpy(char *dest, const char *src);
```

Функция копирует строку src в dest и возвращает dest.

Функция **strlen** вычисляет длину строки. Функция объявлена следующим образом:

```
size_t strlen(const char *s);
```

Функция вычисляет длину строки s и возвращает полученное значение (символ конца строки не учитывается).

Функция **strstr** ищет в строке вхождение заданной подстроки. Функция объявлена следующим образом:

```
char *strstr(const char *s1, const char *s2);
```

Функция осуществляет поиск в s2 первого вхождения подстроки s1. Функция возвращает указатель на элемент в строке s1, с которого начинается s2. Если s2 не обнаружена в s1, функция возвращает NULL.

**Пример 1:**

```
#include <stdio.h>  
#include <string.h>  
#define SIZE 80  
  
int main (void)  
{  
char flower[SIZE];  
char addon[]="  прекрасно пахнет";  
puts ("Какой ваш любимый цветок?");  
gets (flower);  
strcat(flower, addon);  
puts(flower);  
puts (addon);  
  
return 0;  
}
```

Вывод программы:

```
Какой ваш любимый цветок?  
Роза  
Роза  прекрасно пахнет  
      прекрасно пахнет
```

## Пример 2:

Предположим, что мы хотим сравнить ответ со строкой. Будет ли работать данная программа правильно?

```
#include <stdio.h>

#define ANSWER "Grant"

int main (void)
{
    char try[40];

    puts("Кто похоронен в могиле Гранта?");
    gets (try);

    while( try != ANSWER)
    {
        puts("Неправильно! Попробуйте еще раз");
        gets (try);
    }
    puts("Теперь правильно! ");

    return 0;
}
```

К сожалению, нет! Надо сравнивать сами строки, а не их адреса.

Правильный вариант программы:

```
#include <stdio.h>
#include <string.h>

#define ANSWER "Grant"
#define MAX 40

int main (void)
{
    char try[MAX];

    puts("Кто похоронен в могиле Гранта?");
    gets (try);

    while(strcmp( try ,ANSWER)!=0)
    {

        puts("Неправильно! Попробуйте еще раз");
        gets (try);
    }
    puts("Теперь правильно! ");
    return 0;
}
```

- **Упражнение 1**

Написать собственную версию функции **strncpy(s1,s2,n)**.

Библиотечная функция **strncpy(s1,s2,n)** копирует ровно **n** символов из строки **s2** в **s1** и возвращает строку. Если **n** меньше или равно длине строки **s2**, то нулевой символ не добавляется автоматически в строку **s1**. Если **n** больше, чем длина **s2**, то в строку **s1** добавляется нулевой символ(ы) до длины **n**. Поведение библиотечной функции не определено, если строки **s1** и **s2** перекрываются. В своей функции предусмотреть проверку того помещаются ли копируемые символы в строку **s1**, проверку значения **n**.

**Проверить полученную функцию в программе, которая в цикле передает входные значения данной функции. Вывести на экран символы, скопированные в строку s1,**

вместо нулевых символов вывести на экран строку «нулевой символ» (определить через `#define`). Написать функцию для вывода.

- **Упражнение 2**

Используя функцию `strlen`, написать функцию, которая укорачивает строку. Функция принимает в качестве параметров строку и новую длину строки. Функция должна укоротить строку, если длина строки больше передаваемого функции параметра `size`, в противном случае вывести на экран предложение, что строка осталась неизменной. **Протестировать** полученную функцию в программе, которая в цикле передает входные значения данной функции.

[http://www.gnu.org/software/libc/manual/html\\_mono/libc.html#Copying-and-Concatenation](http://www.gnu.org/software/libc/manual/html_mono/libc.html#Copying-and-Concatenation)

[http://www.gnu.org/software/libc/manual/html\\_mono/libc.html#Search-Functions](http://www.gnu.org/software/libc/manual/html_mono/libc.html#Search-Functions)

Марина Брик

*Составлено: 23.11.2010*