

Лабораторная работа №7

Символьные строки

Цель работы: Освоить методику работы с символьными строками. Научиться использовать функции для работы со строками.

Теоретические сведения

В языке Си отсутствует специальный строковый тип. Строки в языке Си представляются как массив элементов типа `char`, в конце которого помещен символ `'\0'` – признак конца строки. Такой массив называется строкой. При объявлении строки необходимо предусмотреть место для `'\0'`, которым должна заканчиваться строка.

Если объявить `char str[10]`; и инициализировать следующим образом:

```
for(int i=0; i < 10; i++)
{
    printf("\nВведите символ");
    scanf("%c",&str[i]);
}
```

то в выделенную область запишутся символы, и это будет массив символов. Если инициализировать так: `gets(str)`;, то функция `gets()` в конец строки дописывает `'\0'` – признак конца строки и это будет строка. Например:

```
char *ss;
int n;
printf("\nВведите длину строки");
scanf("%d", &n);                // ввод длины строки
ss=(char *)calloc(n,sizeof(char)); // выделить память под строку
// ss=new char[n];
gets(ss);                        // ввод строки
```

Функции для работы со строками

Ввод строк

Прототип функции `gets()`;
`gets(char *)`;

Аргументом функции является указатель на строку, читает строку до тех пор, пока не встретит символ ввода `'\n'`. После считывания символа `'\n'` превращает его в символ конца строки `'\0'` и добавляет его в конец строки, поэтому в строке необходимо оставить место для символа `'\0'`.

Функция `scanf()` по формату `%s` читает до первого пустого символа в строке.

```
char str[10];
scanf("%s",str);
снег_снег    // будет прочитано все
снег снег    // будет прочитано только слово снег
```

Вывод строк

Функция puts(). Аргументом функции является указатель на строку. Функция puts() прекращает работу, когда встречается признак конца строки '\0'.

```
char str[10]="Зима";  
puts(str);
```

Приведем прототипы некоторых, часто используемых, функций для работы со строками. Прототипы этих функций находятся в файле <string.h>

```
unsigned strlen(char *str); // вычисляет длину строки не включая признак конца  
// строки '\0'.
```

```
int strcmp(char *str1, char *str2);
```

Сравнивает строки str1 и str2. Результат отрицательный, если str1 < str2, равен нулю, если str1==str2, и положителен, если str1 > str2. Функция возвращает разницу между кодами ASCII первой пары несовпадающих символов.

```
char * strcpy(char *str1, char *str2); // копирует строку str2 в место памяти, на  
которое указывает str1;
```

```
char * strcat(char *str1, char *str2); // присоединяет строку str2 в конец строки  
str1;
```

```
char * strstr(char *str1, const char *str2); // отыскивает первое вхождение строки  
str2 в строку str1.
```

Порядок выполнения работы

1. Изучить краткие теоретические сведения.
2. Составить блок-схему алгоритма.
3. По разработанной блок-схеме алгоритма написать программу. Память под строки выделять динамически.
4. Отладить и выполнить программу.

Варианты заданий

1. Даны натуральное число n , символы s_1, \dots, s_n . Преобразовать последовательность s_1, \dots, s_n , удалив каждый символ * и повторив каждый символ, отличный от *.

2. Даны натуральное число n и символы s_1, \dots, s_n среди которых есть двоеточие. Получить все символы, расположенные между первым и вторым двоеточием. Если второго двоеточия нет, то получить все символы, расположенные после единственного имеющегося двоеточия.

3. Даны натуральное число n и символы s_1, \dots, s_n . Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Найти длину самого короткого слова.

4. Даны натуральное число n и группы символов s_1, \dots, s_n и a_1, \dots, a_n . Удалить из строки s_1, \dots, s_n все символы, которые принадлежат строке a_1, \dots, a_n .

5. Даны натуральное число n и символы s_1, \dots, s_n . Известно, что среди данных символов есть хотя бы один, отличный от пробела. Требуется

преобразовать последовательность s_1, \dots, s_n следующим образом. Удалить группы пробелов, которыми начинается и которыми заканчивается последовательность, а также заменить каждую внутреннюю группу пробелов одним пробелом. Если указанных групп нет в данной последовательности, то оставить последовательность без изменения.

6. Преобразовать текст, состоящий только из прописных букв, в текст, состоящий из прописных и строчных букв. Первая буква после точки – прописная, остальные – строчные.

7. Даны натуральное число n и символы s_1, \dots, s_n . Удалить из каждой группы идущих подряд цифр, в которой более двух цифр и которой предшествует точка, все цифры, начиная с третьей (например, $ab+0.1973-1.1$ преобразуется в $ab+0.10-1.1$).

8. Даны натуральное число n и символы s_1, \dots, s_n . Группы символов, разделенные пробелами (одним или несколькими) и не содержащими пробелов внутри себя, будем называть словами:

а) подсчитать количество слов в данной последовательности;

б) преобразовать данную последовательность, заменяя всякое вхождение слова $str1$ длиной n , на слова $str2$ длиной m .

9. Даны натуральное число n и символы s_1, \dots, s_n . Строку s_1, \dots, s_n обработать так, чтобы все слова, состоящие только из цифр, были удалены, их сумма стояла последним словом в строке.

10. Даны натуральное число n и символы s_1, \dots, s_n . Среди символов этой строки особую роль играет знак $\#$, появление которого означает отмену предыдущего символа строки; k знаков $\#$ отменяет k предыдущих символов. Преобразовать строку с учетом роли знака $\#$. Например, строка «VR#Y##HELO#LO» должна быть «HELLO». Результирующую строку вывести на экран.

11. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Заменить в строке одно заданное слово (если оно есть) другим.

12. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Вывести все слова, длина которых больше заданного числа.

13. Дана строка символов. Слова в строке отделяются друг от друга любым количеством пробелов. Преобразовать строку таким образом, чтобы слова отделялись строго одним пробелом

14. Дана строка символов. Слова в строке отделяются одним пробелом. Поменять местами самое длинное и самое короткое слово.

15. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Удалить из строки самое длинное слово.