

Тема 9 (продължение)

Символни низове в C/C++

1. Масиви от символи. Символни низове

За разлика от други програмни езици, в език C не е предвидена възможност за дефиниране на променливи от тип *низ (string)*. За целта, като низове се използват масиви от символи т.е. масиви от тип *char*.

Когато се дефинира символен низ, трябва да се предвиди следната особеност: ако даден низ съдържа *n* символа, размерът на масива от символи е *n+1*. Последен символ е *управляваща константа за край на низ: '\0' (NULL)*, който автоматично се добавят при въвеждане на низа чрез *cin* или *scanf()*. Пример:

```
char str[20];
```

В случая, дефиниран е символен низ, който може да съдържа 19 символа максимално, последният символ в низа е управляващата константа за край на низ.

Пример за програма, която работи с масиви от символи е следната задача: Да се намери колко пъти се съдържа даден символ в низ.

```
#include <stdio.h>

main()
{
    char str[256];
    char ch;

    printf("Enter a character:");
    scanf("%c",&ch);
    printf("Enter a string:");
    scanf("%s",str);

    int counter = 0, i = 0;
    while(str[i]!='\0')
    {
        if(str[i]==ch) counter++;
        i++;
    }
    printf("\nCharacter %c contains %d times\n",ch,counter);
    return 0;
}
```

Масив от символи може да бъде инициализиран по два начина:

- При дефинирането на масив от символи се присвоява като стойност низова константа (поредица от символи, затворени в апострофи). В случая, добавянето на управляваща константа '\0' за край на низ е автоматично. Пример:

```
char str[] = "abcd";
```

Масивът `str` съдържа 5 елемента; последният е със стойност `'\0'`. На всеки елемент се присвоява определена стойност. В този случай константата `'\0'` се присвоява автоматично на последния елемент на масива.

- Друг начин за инициализиране на символен низ е този, по който обикновено се задават начални стойности на елементите на масива:

```
char str[] = {'a', 'b', 'c', 'd', '\0'};
```

В този случай, на последният елемент задължително се задава стойност `'\0'`. Разбира се, вторият начин е твърде неудобен и затова не се практикува.

За работа с низове в стандарта ANSI C са предвидени различни функции, чиито декларации са в заглавен файл ***string.h***. Най-често използваните функции за работа с низове са:

- `strcpy()`

Функцията има следния прототип (заглавие):

char *strcpy(char *to, const char *from);

Функцията копира символите от низ `from` в низ `to`, включително и символът за край на низ. Трябва да се отбележи, че функцията не прави проверка дали двата низа не са един и същ и е възможно препокриване на двата низа.

- `strncpy()`

Функцията има следния прототип (заглавие):

char *strncpy(char *to, const char *from, int count);

Функцията копира най-много `count` на брой символа от низ `from` в низ `to`, като добавя и символът за край на низ. Функцията връща резултантния низ.

- `strlen()`

Функцията има следния прототип (заглавие):

int strlen(char *str);

Функцията намира дължината на низа и връща като стойност броя на символите в низа, без символът за край на низ.

- `strcmp()`

Функцията има следния прототип (заглавие):

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

Функцията сравнява два низа и връща следните стойности:

по-малка от 0 (нула), ако `str1` е по-малък от `str2`;

0 (нула) – ако низовете са еднакви;

по-голяма от 0 (нула), ако `str1` е по-голям от `str2`;

Трябва да се отбележи, че използването на тези функции изисква включване на заглавен файл ***string.h***:

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

2. Тип **string** в програмен език C++

Стандартната C++ библиотека поддържа тип ***string***. Дефинирането на символен низ може да бъде по следния начин:

```
string str1 = "Hello";  
string str2 = "World";  
string str3;
```

В случая, дефинирани са три променливи от тип низ като първите две са инициализирани с посочните низови константи. Третата променлива не е инициализирана.

Допустимо е присвояване както на стойност низова константа, така и присвояването на стойност от друга низова променлива. Например, следващият програмен ред ще копира стойността "Hello" на променливата `str3`.

```
str3=str1;
```

Оператор + (събиране) може да се използва с операнди низове като резултатът е слепване на двата низа. В случая, този оператор се нарича **конкатенация**.

Пример:

```
str3=str1+str2;  
cout<<str3;
```

В случая, на екрана ще се изведе съобщението "HelloWorld".

Низовете могат да бъдат операнди и в операторите за сравнение <, >, <=, >=, ==, !=. Чрез тях се сравнява съдържанието на два низа. Както и в език C, допустимо е да се извлече символ от низ като се използва индексирание.

Пример:

```
string city = "Gabrovo";  
char firstSymbol=city[0];  
cout<<firstSymbol;
```

В посочения пример променливата `firstSymbol` получава стойност G и този символ се извежда на екрана.

Относно въпроси по темата на адрес: ln_zh_st@yahoo.com