Упражнение №9

СИМВОЛНИ НИЗОВЕ. ОБРАБОТКА НА НИЗОВЕ

Символен низ е масив от тип **char**, който завършва с нулевия символ **'\0'**. ***Низова константа*** е последователност от символи, заградена с кавички (").

```
char s[11];                        // s може да съхрани до 10 символа
#define MSG "Тестване на програма"              // низова константа
```

1. Деклариране на низ

а) наименована КОНСТАНТА за дължината на низа

```
#define NAME_LENGTH 30
```

б) дължината на низа е КОНСТАНТА+1 (\0 за край на низа)

в) инициализиране на низа с нула, за да се избегнат безкрайни низове

- инициализиране на низа с 0 при деклариране

```
char name[NAME_LENGTH+1] = {0};
```

- отделяне на памет за динамичен низ и инициализиране с 0 на всеки елемент чрез функцията **calloc()** вместо **malloc()**

```
char *otherName = (char *)calloc(NAME_LENGTH+1, sizeof(char));
```

г) разлика между указател към низ и масив от символи

```
char *stringPtr;
stringPtr = "Тестване на програма";
```

"Тестване на програма" е указател към низова константа и операцията присвояване (**=**) насочва указателя **stringPtr** към низовата константа **"Тестване на програма"**, но не копира съдържанието му в **stringPtr**. ***Препоръчва се използване на масиви от символи вместо указатели.***

Пример 1: Инициализиране на всички символи в низ с даден символ.

```
#include <stdio.h>
#define NAME_LENGTH 30
// Инициализира елементите на низа s със символа c.
void initialize(char *s, char c);

int main()
{
  char name[NAME_LENGTH+1] = {0};
  // Инизиализира всичките символи в низа name със символа 'A'
  initialize(name, 'A');
  printf("|%s|\n", name);
  return 0;
}

void initialize(char *s, char c)
{
  for(int i=0; i<NAME_LENGTH; i++)
    s[i] = c;
}
```

2. Операции с низове – извършват се чрез функциите **strlen()**, **strcpy()**, **strncpy()**, **strcmp()**, **strncmp()** и т.н. ***Препоръчва се използване на сигурните версии***

strncpy() и strncmp() вместо strcpy() и strcmp(), за да се избегнат безкрайни низове.

```
// Копира низа otherName в name с максимална дължина за копиране
// NAME_LENGTH чрез strncpy.
strncpy (name, otherName, NAME_LENGTH);
// Сравнява най-много NAME_LENGTH символа от otherName в name и връща
// стойност 0 при съвпадение и различна от нула при несъвпадение.
if(!strncmp(name, otherName, NAME_LENGTH))
    printf("Двата низа съвпадат\n");
  else
    printf("Двата низа не съвпадат\n");
```

Пример 2: Дължина на низ

Алгоритъм 1:

брой ← 0
докато не е достигнат краят на низа повтаряй
  брой ← брой + **1**
  преместване към следващия символ в низа

```
int length (char *s)
{
   int n = 0;
   while (*s != '\0')
   {
      n++;
      s++;
   }
   return n;
}
```

Алгоритъм 2:

указател **p ← s**
докато **p** не е достигнал края на низа повтаряй
  преместване на **p** към следващия символ в низа
брой ← p - s

```
int length (char *s)
{
   char *p = s;
   while (*p != '\0')
      p++;
   return p - s;
}
```

Пример 3: Копиране на низ

Алгоритъм:

копира символ от **s2** в **s1** и докато копираният символ не е достигнал края на **s2** повтаряй
  преместване на **s1** към следващия символ в низа **s1**
  преместване на **s2** към следващия символ в низа **s2**

```
// Вариант 1
void copy (char *s1, char *s2)
{
   while ((*s1 = *s2) != '\0')
   {
      s1++;
      s2++;
   }
}
```

```
// Вариант 2
void copy (char *s1, char *s2)
{
   while((*s1++=*s2++) != '\0')
      ;
}
```

Пример 4: Сравняване на низове

Алгоритъм:

докато съответстващите символ от **s1** и **s2** съвпадат повтаряй
  ако съвпадащият символ е край на низ
    s1 и **s2** съвпадат

преместване на **s1** към следващия символ в низа **s1**
преместване на **s2** към следващия символ в низа **s2**
разлика между несъвпадащите символи

```
//         < 0, ако s1 < s2;
// Връща = 0, ако s1==s2;
//         > 0, ако s1>s2.
int compare (char *s1, char *s2)
{
   while (*s1 == *s2)
   {
      if (*s1 == '\0')
         return 0;
      s1++;
      s2++;
   }
   return *s1 - *s2;
}
```

Пример 5: Да се напише функция, която намира колко пъти даден символ се съдържа в низ.

Алгоритъм:

брой ← 0
докато не е достигнат краят на низа повтаряй
  ако символът от **s** = дадения символ
    брой ← брой + **1**
  преместване към следващия символ в низа

```
int contain(char *s, char c)
{
  int n = 0;
  while (*s != '\0')
  {
    if(*s == c)
      n++;
    s++;
  }
  return n;
}
```

Задача: Да се напише функция, която намира броя на главните букви и броя на препинателните знаци в символен низ. Да се напише програма, която демонстрира използването на функциите.