

Лекція 20. Рядки

В мові Сі немає спеціального типу даних, який можна було б використовувати для опису рядків.

Замість цього рядки представляються у вигляді масиву елементів типу *char*.

Це означає, що символи рядка розташовуються в пам'яті в сусідніх комірках, по одному символу в комірці.

Ц	е		р	я	д	о	к	\0
---	---	--	---	---	---	---	---	----

В Сі *рядок символів* розглядають як звичайний масив символів, в якому останнім є символ *'\0'*. При цьому довжина масиву визначається на одиницю більше ніж довжина реального рядка.

В мові Сі використовується так звана концепція «*zero termination string*». Це означає, що для позначення закінчення кожного рядка (масиву типу char) використовується нуль-символ ("'\0'")

Оголошення `char str[10];`

передбачає, що рядок може містити максимум 9 символів.

Основні методи ініціалізації символічних рядків:

- `char str1[] = "ABCdef";`
 - `char str2[] = {'A', 'B', 'C', 'd', 'e', 'f', '0'};`
 - `char str3[10];`
 - `char str4[10] = "HELLO";`
 - `char * str5;`
-

Введення - виведення

Рядок можна обробляти як масив символів, використовуючи алгоритми обробки масивів або за допомогою функцій обробки.

Символ:

```
scanf("%c", &c);
```

Слово:

```
scanf("%s", s);
```

Рядок:

```
gets()
```

Для виведення рядків можна використовувати функції *puts()* і *printf()*

Синтаксис функції *puts()*: *int puts(char *string);*

```
char str1[] = "ABCdef";
printf("str1 - %s\n", str1);
char str2[] = { 'A', 'B', 'C', 'd', 'e', 'f', 0 };
printf("str2 - %s\n", str2);
```

ПРИКЛАД №1

```
char str3[100];
printf("\nрядок - ");
gets_s(str3);
puts(str3);
printf("%s\n", str3);
```

```
char str4[100];
printf("\nСлово - ");
scanf("%s", str4);
printf("%s\n", str4);
```

```
char c1;
printf("\nСимвол - ");
scanf("%c", &c1); scanf("%c", &c1);
printf("%c\n", c1);
```

```
str1 - ABCdef
str2 - ABCdef

рядок - qwerty 101
qwerty 101
qwerty 101

Слово - Qwerty
Qwerty

Символ - G
G
```

Необхідно мати також на увазі те, що рядок вигляду "x" - не те ж саме, що символ 'x'.

1: 'x' - об'єкт одного з основних типів даних мови Сі (*char*), в той час, як "x" - об'єкт похідного типу (масиву елементів типу *char*).

2: "x" насправді складається з двох символів - символу 'x' і нуль-символу.

'x'	x
"x"	x \0

Стандартна бібліотека мови програмування Сі містить клас функцій для роботи з рядками, і всі вони починаються з літер str. Для того, щоб використовувати одну або декілька функції необхідно підключити файл string.h.

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

Визначення довжини рядка

Для визначення довжини рядка використовується функція *strlen()*.

Її синтаксис : *size_t strlen(const char *s);*

Функція *strlen()* повертає довжину рядка *s*, при цьому завершуючий нульовий символ не враховується.

ПРИКЛАД №2

```
system("chcp 1251");  
system("cls");
```

```
char s[100] = "Чого Івась не навчиться, того й Іван не знатиме";  
puts(s);  
int len;  
len = strlen(s);  
  
printf("=%d\n", len);
```

```
Чого Івась не навчиться, того й Іван не знатиме  
=47
```

Копіювання рядків

Оператор присвоювання для рядків не визначений. Тому, якщо *s1* і *s2* - символічні масиви, то неможливо скопіювати один рядок в інший наступним чином:

```
char s1[100];
```

```
char s2[100];
```

```
s1 = s2; /* помилка */
```

Для копіювання необхідно скористатися функцією *strcpy()*.

```
strcpy(s1,s2);
```

Для копіювання рядків можна використовувати і функцію *strncpy()*, яка дозволяє обмежувати кількість символів, що копіюються.

```
strncpy(s1, s2, 5);
```

```
system("chcp 1251");  
system("cls");
```

ПРИКЛАД №3

```
char s1[100] = "Чого Івась не навчиться, того й Іван не  
знатиме";  
char s2[100] = "Пан з паном, а Іван з Іваном";  
char s3[100] = "Хто про Хому, а він про Ярему";  
printf("Вхідні рядки:\n");  
puts(s1);  
puts(s2);  
puts(s3);  
printf("Вихідні рядки:\n");  
strcpy(s1, s2);  
puts(s1);  
  
strncpy(s2, s3, 12);  
puts(s2);  
puts(s3);
```

```
Вхідні рядки:  
Чого Івась не навчиться, того й Іван не знатиме  
Пан з паном, а Іван з Іваном  
Хто про Хому, а він про Ярему  
Вихідні рядки:  
Пан з паном, а Іван з Іваном  
Хто про Хому а Іван з Іваном  
Хто про Хому, а він про Ярему
```

Конкатенація рядків

Конкатенація двох рядків означає їх **об'єднання**, при цьому створюється новий, більш довгий рядок.

strcat (s1, s2) об'єднує рядок s2 з рядком s1. Результат зберігається в s1.

strncat (s1, s2, n) об'єднує n символів рядка s2 з рядком s1. Результат зберігається в s1

```
system("chcp 1251");  
system("cls");
```

```
char s1[200] = "Чого Івась не навчиться, того й Іван не знатиме";  
char s2[100] = "Пан з паном, а Іван з Іваном";  
char s3[100] = "Хто про Хому, а він про Ярему";  
printf("Вхідні рядки:\n");  
puts(s1); puts(s2); puts(s3);  
printf("Вихідні рядки:\n");  
strcat(s2, s1);  
puts(s2);  
printf("-----\n");  
strcat(strncat(s1, s2, 3), s3);  
puts(s1);
```

Вхідні рядки:

Чого Івась не навчиться, того й Іван не знатиме

Пан з паном, а Іван з Іваном

Хто про Хому, а він про Ярему

Вихідні рядки:

Пан з паном, а Іван з ІваномЧого Івась не навчиться, того й Іван не знатиме

Чого Івась не навчиться, того й Іван не знатимеПанХто про Хому, а він про Ярему

Для продовження натисніть будь-яку клавішу . . .

Порівняння рядків

int strcmp (char *st1, char *st2); — порівнює рядки **st1** і **st2** та повертає цілу величину, що дорівнює:

<0 — якщо **st1 < st2;**

= 0 — якщо **st1 = st2;**

>0 — якщо **st1 > st2;**

```
char st1[] = "Слово";  
char st2[] = "слово";  
int k;  
k = strcmp(st1, st2);  
printf("k=%d\n", k);
```

k=-1

```
char st1[] = "Слово";  
char st2[] = "Слово";  
int k;  
k = strcmp(st1, st2);  
printf("k=%d\n", k);
```

k=0

```
char st1[] = "Слово";  
char st2[] = "слово";  
int k;  
k = strcmp(st1, st2);  
printf("k=%d\n", k);
```

k=1

Функції перетворення символів рядка

char *strlwr (char*st); — перетворює символи рядка **st** верхнього регістра в символи нижнього регістра, при цьому інші символи не враховуються.

char *strupr (char *st); — перетворює символи рядка **st** нижнього регістра в символи верхнього регістра, інші символи не враховуються.

char *strrev (char *st); — записує символи в рядку **st** у зворотному порядку.

Дано рядок, що складається із цифр, арифметичних операцій та деякої кількості відкриваючих та закриваючих дужок. Перевірити кількість відкриваючих та закриваючих дужок.



```
int count1=0, count2=0;
char str[] = "(1+2-125))-(45*2)-5)";
for (int i = 0; i < strlen(str); i++) {
    if (str[i] == '(') count1++;
    if (str[i] == ')') count2++;
}
printf("( =%d\n) =%d\n", count1, count2);
```

```
( =2
) =4
```

Речення складається із деякої кількості слів, розділені пробілами. Вивести на екран окремі слова, впорядкувавши їх у алфавітному порядку.



```
system("chcp 1251");  
system("cls");
```


```
char *s11;  
char str[] = "Паша Оля Коля Маша Анна";
```

```
s11 = strtok(str, " ");  
while (s11 != NULL) {  
    printf("[%s]\n", s11);  
    s11 = strtok(NULL, " ");  
}
```

```
[Паша]  
[Оля]  
[Коля]  
[Маша]  
[Анна]
```

```
char *s11, s12[50], *tmp, str2[100]; int f11 = 0, f12 = 0, k;  
char str[] = "Паша Оля Коля Маша Анна";
```

```
s11 = strtok(str, " ");  
while (s11 != NULL) {  
    printf("[%s]\n", s11);  
    if (f11 == 0) {strcpy(s12, s11); f11 = 1;}  
    else { k = strcmp(s11, s12);  
        if (k == -1 || k == 0) {  
            if (f12==0) { strcpy(str2, s11); strcat(str2, " "); f12 = 1;  
            }  
            else { strcat(str2, s11); strcat(str2, " ");  
            }  
        }  
        else {  
            if (f12==0){strcpy(str2, s12); strcat(str2, " "); f12 = 1;}  
            else {  
                strcat(str2, s12); strcat(str2, " "); strcpy(s12, s11);  
            }  
        }  
    }  
    s11 = strtok(NULL, " ");  
}  
strcat(str2, s12); puts(str2);
```



```
[Паша]  
[Оля]  
[Коля]  
[Маша]  
[Анна]  
Оля Коля Маша Анна Паша
```

```

#define n 2
void strTokS(char *str);
void main(){
    system("chcp 1251"); system("cls");
    char sentence[256], *Str;
    for (int i = 0; i < n; i++) {
        printf("Введіть речення:\n");
        gets_s(sentence);
        Str = sentence; strTokS(Str);
        printf("\n");
    }
}
void strTokS(char *str) {
    char *word; char *next = NULL;
    word = strtok_s(str, " ", &next);
    do {
        printf("%2d ", strlen(word));
        word = strtok_s(NULL, " ", &next);
    } while (word != NULL);
}

```

Введите предложение :

В мові Сі немає спеціального типу даних, який можна було б використовувати для опису рядків

1 4 2 5 12 4 6 4 5 4 1 15 3 5 6

Введите предложение :

Замість цього рядки представляються у вигляді масиву елементів типу char

7 5 5 15 1 7 6 9 4 4

```
#include<stdio.h>
#include <windows.h>
#include<malloc.h>
int main(){
system("chcp 1251"); system("cls");
char *ptr;
ptr = (char*)realloc(NULL, 20 * sizeof(char));
strcpy(ptr, "Це перша частина, "); printf("%s\n", ptr);
ptr = (char*)realloc(ptr, 100 * sizeof(char));
strcat(ptr, "це друга частина");
printf("%s\n", ptr);
realloc(ptr, 0);
//-----
ptr = (char*)malloc(20 * sizeof(char));
strcpy(ptr, "Це треття частина, ");
printf("%s\n", ptr);
ptr = (char*)realloc(ptr, 100 * sizeof(char));
strcat(ptr, "це четверта частина"); printf("%s\n", ptr);
realloc(ptr, 0);
return 0;
}
```

