

Тема: Рядкові величини. Структура опису величини рядкового типу

Комп'ютер використовують для обробки інформації різних видів, зокрема, і текстової. Текст можна розглядати як сукупність символічних рядків. Розглянемо засоби, які мова C++ надає для роботи з ними.

Оголошення символічних рядків у програмі

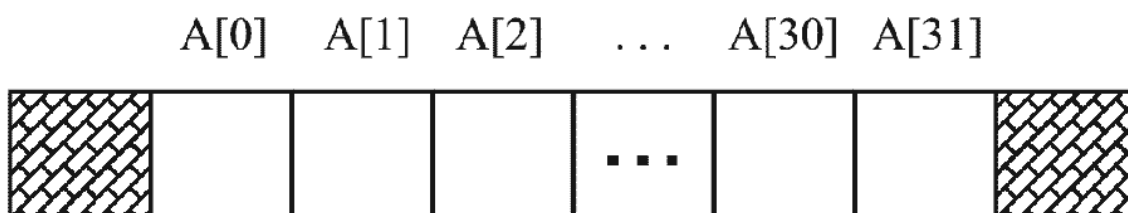
Символьні рядки у C++ являють собою звичайні масиви.

Головна відмінність між символічними рядками й іншими типами масивів полягає в тому, що символічний рядок можна виводити, наприклад, командою `cout << abc;`, а не організувати цикл для по елементного виведення даних.

Кінець символічного рядка позначається символом `NULL`, що у програмі зображується як спеціальний символ `'\0'`. Можна сказати, що саме `NULL` перетворює частину символічного масиву в символічний рядок.

Для оголошення символічного рядка усередині програми просто оголосіть масив типу `char` з кількістю елементів, достатньою для зберігання необхідних символів. Наприклад, оголосимо змінну з ім'ям `A`, здатну зберігати 32 символи (не забувайте, що символ `NULL` є одним із цих 32 символів): `char A[32];`

Як видно з мал. 1, це оголошення створює масив з елементами, пронумерованими від `A[0]` до `A[31]`.



C++ трактує символічний рядок як масив типу `char`

Розглянуті раніше програми використовують рядкові константи, уміщені у подвійні лапки:

"Це рядкова константа"

При створенні рядкової константи компілятор C++ автоматично додає символ `NULL`, як показано на малюнку 2.

Ц	Е		Р	Я	Д	К	О	В	А		К	О	Н	С	Т	А	Н	Т	А	\0
---	---	--	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	----

Компілятор C++ автоматично додає символ NULL

Коли при виконанні програми за допомогою вихідного потоку cout виводиться рядкова константа, cout використовує символ '\0' як ознаку кінця рядка.

Більшість функцій C++, призначених для роботи з рядками, використовують символ NULL для визначення останнього символу рядка.

У наступній програмі уводиться рядок з використанням циклу for, а потім виводиться на екран за допомогою cout:

```
#include<iostream.h> //Програма 6.1
#include<conio.h>
int main()
{
char abc [27]; // 26 символів плюс NULL
char let; int i;
//Заповнення рядкового масиву латинськими
літерами
for (let = 'A', i = 0; let <= 'Z'; let++, i++)
abc [i] = let;
abc [i] = '\0'; //Ознака кінця рядка
cout << abc; //Виведення рядкової константи
getch();
return 0;
}
```

У програмах на C++ зустрічаються окремі символи, взяті в одинарні лапки (наприклад, 'A'), і символи у подвійних лапках ("A").

Між цими записами є принципова різниця. Символ в одинарних лапках являє собою символну константу.

Компілятор C++ виділяє тільки один байт пам'яті для її зберігання. Однак символ у подвійних лапках являє собою рядкову константу – зазначений

символ і символ NULL (доданий компілятором). Таким чином, компілятор виділить два байти для символного рядка. Мал.3 ілюструє, як компілятор C++ зберігає рядкову константу "A" і символну константу 'A'.



Як "A" відрізняється від 'A'

Ініціалізація символного рядка

Як ви вже знаєте, C++ дозволяє ініціалізувати масиви при оголошенні.

Символьні рядки C++ не є винятком. Для ініціалізації символного рядка при оголошенні вкажіть необхідну послідовність символів усередині подвійних лапок, як показано нижче: `char tit [32] = "Вивчаємо мову C++";`

Якщо кількість символів, що присвоюється рядку, менша від розміру масиву, більшість компіляторів C++ присвоюють символи NULL елементам рядкового масиву, що залишаються. Як і у випадку з масивами інших типів, якщо ви не вказуєте розмір масиву, що ініціює при оголошенні, компілятор C++ виділить достатньо пам'яті для розміщення зазначених символів рядка і службового символу NULL: `char tit[] = "Вивчаємо мову C++";`

При компіляції наведеного фрагменту для рядка `tit` буде виділено 18 байтів (15 зображених символів + 2 пропуски + символ '\0').

У наступній програмі символні рядки ініціалізуються при оголошенні:

```
#include<iostream.h> //Програма 6.2
```

```
#include<conio.h>
```

```
int main()
```

```
{
```

```
char tit[32] = "Hello, C++";
```

```
char les[] = "Simvolni ryadki!";
```

```

cout<<"Book: "<<tit<<endl;
cout<<"Lesson: "<<les<<endl;
getch();
return 0;
}

```

Зверніть увагу, що для масиву `tit` виділено 32 байти, хоча рядок, занесений в нього при ініціалізації, – коротший. Тому пізніше, в програмі, у цей масив можна буде вмістити довший рядок, обов’язково помістивши в кінці символ ‘\0’.

Передавання рядків у функції

Передавання символьного рядка у функцію подібне до передавання будь-якого масиву як параметру. У середині функції потрібно просто вказати тип масиву (`char`) і квадратні дужки масиву. Не треба вказувати розмір рядка. Наприклад, наведена програма використовує функцію `show_ryad` для виведення символьного рядка:

```

#include<iostream.h> //Програма 6.3
#include<conio.h>
void show_ryad(char ryad[])
{
cout << ryad << endl;
}
int main()
{
show_ryad("Hello, C++!");
show_ryad("Good Bye, C++!");
getch();
return 0;
}

```

Як бачите, функція `show_ryad` трактує параметр символьного рядка як масив:

```
void show_ryad(char ryad[])
```

Оскільки символ NULL вказує кінець рядка, функція не вимагає параметра, що задає кількість елементів у масиві. Замість цього функція може визначити останній елемент, просто знайшовши в масиві символ NULL.

Як ви вже знаєте, функції C++ часто використовують символ NULL для визначення кінця рядка. Наступна програма містить функцію з ім'ям `str_len`, що шукає у рядку символ NULL для визначення кількості символів у ньому. Далі функція використовує оператор `return` для повернення значення довжини рядка до функції, що викликала. При виконанні програми кілька різних символічних рядків передаються у функцію, і на екран виводиться довжина кожного з них:

```
#include<iostream.h> //Програма 6.4
#include<conio.h>
int str_len(char ryad[])
{
int i;
for (i = 0; ryad[i] != '\0'; i++);
return(i);
}
int main()
{
char title[]="Vchimosya programuvannyu";
char lesson[]="Simvolni ryadki";
cout<<"Ryadok: "<<title<<" mae "<<
str_len(title)<<" simvoliv"<<endl;
cout<<"Ryadok: "<<lesson<<" mae "<<
str_len(lesson)<<" simvoliv"<<endl;
getch();
return 0;
}
```

Як бачите, у функції організовано перегляд рядка, починаючи з першого символу (елемент з індексом 0), і при виявленні NULL перевірка припиняється. На цей момент змінна і дорівнює кількості символів у рядку, тому саме її значення повертають оператором `return(i)`;
Далі ви побачите, що наведена функція моделює одну зі стандартних функцій для роботи з рядками.

Питання для самоконтролю:

1. Що являють собою символні рядки?
2. Чим символні рядки відрізняються від звичайних масивів?
3. Для чого використовується символ NULL?
4. Чим 'В' відрізняється від "В"?
5. Як здійснюється ініціалізація символного рядка?
6. Як здійснюється передавання рядків у функцію?

Вправи

- 1) Випробуйте програму 6.1. Уважно дослідіть роботу програми. Виведіть маленькі літери латинського алфавіту. (Не виводьте за допомогою цієї програми українських літер!).
 - 2) Випробуйте програми 6.2. та 6.3. Переробіть програму 6.3 так, щоб параметром, переданим функції, була змінна.
 - 3) Випробуйте програму 6.4. Змініть значення рядкових величин.
- Збережіть програми, створивши у власній папці нову папку wr6-1.