**Strings and null-terminated character sequences.**

Plain arrays with null-terminated sequences of characters are the typical types used in the C language to represent strings (that is why they are also known as *C-strings*). In C++, even though the standard library defines a specific type for strings (class [string](https://www32.cplusplus.com/string)), still, plain arrays with null-terminated sequences of characters (C-strings) are a natural way of representing strings in the language; in fact, string literals still always produce null-terminated character sequences, and not string objects.

In the standard library, both representations for strings (C-strings and library strings) coexist, and most functions requiring strings are overloaded to support both.

For example, *cin* and *cout* support null-terminated sequences directly, allowing them to be directly extracted from *cin* or inserted into *cout*, just like strings. For example:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 | // strings and NTCS:  #include <iostream>  #include <string>  using namespace std;  int main ()  {  char question1[] = "What is your name? ";  string question2 = "Where do you live? ";  char answer1 [80];  string answer2;  cout << question1;  cin >> answer1;  cout << question2;  cin >> answer2;  cout << "Hello, " << answer1;  cout << " from " << answer2 << "!\n";  return 0;  } | What is your name? Homer  Where do you live? Greece  Hello, Homer from Greece! | [Edit & Run](https://www32.cplusplus.com/doc/tutorial/ntcs/) |

In this example, both arrays of characters using null-terminated sequences and strings are used. They are quite interchangeable in their use together with *cin* and *cout*, but there is a notable difference in their declarations: arrays have a fixed size that needs to be specified either implicit or explicitly when declared; question1 has a size of exactly 20 characters (including the terminating null-characters) and answer1 has a size of 80 characters; while strings are simply strings, no size is specified. This is due to the fact that strings have a dynamic size determined during runtime, while the size of arrays is determined on compilation, before the program runs.

In any case, null-terminated character sequences and strings are easily transformed from one another:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 | char myntcs[] = "some text";  string mystring = myntcs; // convert c-string to string  cout << mystring; // printed as a library string  cout << mystring.c\_str(); // printed as a c-string | char myntcs[] = "some text";  string mystring = myntcs; // convert c-string to string  cout << mystring; // printed as a library string  cout << mystring.data(); // printed as a c-string |  |

Null-terminated character sequences can be transformed into strings implicitly, and strings can be transformed into null-terminated character sequences by using either of string's member functions *c\_str* or *data*:

(note: both *c\_str* and *data* members of string are equivalent)