

6.10 Functions with C string parameters

Functions commonly modify C strings. The following function modifies a string by replacing spaces with hyphens.

Figure 6.10.1: Modifying a C string parameter.

```
#include <iostream>
#include <cstring>
using namespace std;

// Function replaces spaces with hyphens
void StrSpaceToHyphen(char modString[]) {
    int i;          // Loop index

    for (i = 0; i < strlen(modString); ++i) {
        if (modString[i] == ' ') {
            modString[i] = '-';
        }
    }
}

int main() {
    const int INPUT_STR_SIZE = 50; // Input C string
    size
    char userStr[INPUT_STR_SIZE]; // Input C string
    from user

    // Prompt user for input
    cout << "Enter string with spaces: " << endl;
    cin.getline(userStr, INPUT_STR_SIZE);

    // Call function to modify user defined C string
    StrSpaceToHyphen(userStr);

    cout << "String with hyphens: " << userStr <<
    endl;

    return 0;
}
```

Enter string with spaces:
Hello there everyone.
String with hyphens: Hello-there-
everyone.

...

Enter string with spaces:
Good bye now !!!
String with hyphens: Good-bye--now-
--!!!

[Feedback?](#)

The parameter definition (yellow highlighted) uses `[]` to indicate an array parameter. The function call's argument (orange highlighted) does not use `[]`. The compiler *automatically passes the C string as a pointer*. Hence, the above function modifies the original string argument (`userStr`) and not a copy.

The `strlen()` function can be used to determine the length of the string argument passed to the function. So, unlike functions with array parameters of other types, a function with a C string

parameter does not require a second parameter to specify the string size.

zyDE 6.10.1: Modifying a C string parameter: Spaces to hyphens.

1. Run the program, noting correct output.
2. Modify the function to also replace each '!' by a '?'.

Load default template...

```

13     }
14 }
15 }
16
17 int main() {
18     const int INPUT_STR_SIZE = 50; // Input
19     char userStr[INPUT_STR_SIZE]; // Input
20
21     // Prompt user for input
22     cout << "Enter string with spaces: " <<
23     cin.getline(userStr, INPUT_STR_SIZE);
24
25     // Call function to modify user defined
26     StrSpaceToHyphen(userStr);
27
28     cout << "String with hyphens: " << userS
29
30     return 0;
31 }
32

```

Run

Hello there everyone!!!

Enter string with spaces: Hello there everyone!!!
String with hyphens: Hello there-everyone!!!

Feedback?

PARTICIPATION ACTIVITY

6.10.1: Functions with string parameters.

- 1) A parameter declared as `char movieTitle[]` is a string.

☐ True
☐ False
- 2) For a function with a string parameter, the function must include a second parameter for the string size.

☐ True
☐ False

- 3) To pass a string to a function, the argument must include [], as in `GetMovieRating(favMovie[])`.

- ☐ True
- ☐ False

[Feedback?](#)

A programmer can explicitly define an array parameter as a pointer. The following uses `char* modString` instead of the earlier `char modString[]`. Such pointer parameters are common for C string parameters, such as in the C string library functions.

Figure 6.10.2: Modifying a C string using a pointer parameter.

```
#include <iostream>
#include <cstring>
using namespace std;

// Function replaces spaces with hyphens
void StrSpaceToHyphen(char* modString) {
    int i;        // Loop index

    for (i = 0; i < strlen(modString); ++i) {
        if (modString[i] == ' ') {
            modString[i] = '-';
        }
    }
}

int main() {
    const int INPUT_STR_SIZE = 50; // Input string
    size
    char userStr[INPUT_STR_SIZE]; // Input C string
    from user

    // Prompt user for input
    cout << "Enter string with spaces: " << endl;
    cin.getline(userStr, INPUT_STR_SIZE);

    // Call function to modify user defined C string
    StrSpaceToHyphen(userStr);

    cout << "String with hyphens: " << userStr <<
    endl;

    return 0;
}
```

```
Enter string with spaces:
Hello there everyone!
String with hyphens: Hello-there-
everyone!

...

Enter string with spaces:
Good bye now !!!
String with hyphens: Good-bye--now-
--!!!
```

[Feedback?](#)

ACTIVITY

6.10.2: Functions with C string parameters.

1) Passing a C string to a function creates a copy of that string within the function.

- ☐ True
☐ False

2) A C string is automatically passed by pointer.

- ☐ True
☐ False

[Feedback?](#)CHALLENGE
ACTIVITY

6.10.1: Modify a C string parameter.

Complete the function to replace any period by an exclamation point. Ex: "Hello. I'm Miley. Nice to meet you." becomes:

"Hello! I'm Miley! Nice to meet you!"

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 void MakeSentenceExcited(char* sentenceText) {
6
7     /* Your solution goes here */
8     for (int i = 0; i < strlen(sentenceText); ++i) {
9         if (sentenceText[i] == '.') {
10             sentenceText[i] = '!';
11         }
12     }
13 }
14
15 int main() {
16     const int TEST_STR_SIZE = 50;
17     char testStr[TEST_STR_SIZE];
18
19     cin.getline(testStr, TEST_STR_SIZE);
20     MakeSentenceExcited(testStr);
21 }
```

Run

✓ All tests passed

✓ Testing with: Alright. I understand.

Your output

Alright! I understand!

✓ Testing with: Howdy friend. Until later...

Your output

Howdy friend! Until later!!!

✓ Testing with: .Hello.

Your output

!Hello!

✓ Testing with:

Your output

Your program correctly produced no output

[Feedback?](#)