6.14 Char arrays / C strings

6.15 C-String library functions

C++ provides functions for working with C strings, presented in the **cstring** library. To use those functions, the programmer starts with: `#include <cstring>`.

Some C string functions for modifying strings are summarized below.

Table 6.15.1: Some C string modification functions.

Given:

```
char orgName[100] = "United Nations";
char newText[20] = "UNICEF";
char targetText[10];
```

strcpy()	strcpy(destStr, sourceStr)
	
	Copies sourceStr (up to and including the null character) to destStr. destStr must have enough space for sourceStr, including the null character.
		`strcpy(targetText, usertext);`
		`// Copies "Hello!" + null char to targetText`
		`strcpy(targetText, orgName);`
		`// Error: "United Nations" has > 10 chars`
		`targetText = orgName;`
		`// Error: Strings can't be copied this way`

strncpy()	strncpy(destStr, sourceStr, numChars) Copies up to numChars characters from sourceStr to destStr must have enough space for the copied characters, including the null character. If sourceStr has less than numChars characters, strncpy will fill the remaining space with the null character.	`strncpy(orgName, usertext, 6);` `// orgName is "UNICEF Nations"`
strcat()	strcat(destStr, sourceStr) Copies sourceStr to end of destStr (starting at destStr's null character) and then appends a null character. destStr must have enough space for the addition of sourceStr plus the null character.	`strcat(orgName, usertext);` `// orgName is "United NationsHELLO"`

strncat()	strncat(destStr, sourceStr, numChars)
	
	Copies up to numChars characters from sourceStr to end of destStr (starting at destStr's null character) and then appends a null character. destStr must have enough space for the addition of sourceStr plus the null character. Copying of sourceStr characters continues until either the end of sourceStr or numChars is reached.
		`strcpy(targetText, "abc");`
		`// targetText = "abc"`
		`strcat(targetText, "123456789", 5);`
		`// targetText is "abc12"`

Feedback?

For strcpy(), a common error is to copy a source string that is too large, causing an out-of-range access in the destination string. Another common error is to call strcpy with the source string first rather than the destination string, which copies in the wrong direction.

Note that string assignment, as in targetText = orgName, does not copy the string and should not be used. The exception is during initialization, as in `char usertext[20] = "UNICEF";`, for which the compiler copies the string literal's characters into the array.

PARTICIPATION ACTIVITY 6.15.1: String modification functions.

Given: char usertext[50]
Do not type quotes in your answers.
If the function call is incorrect, causes an out-of-range access, or the resulting string does not end with a null character, type Error.

1) What is userStr after: strcpy(userStr, "Bye");

Check Show answer

2) If userStr is initially "Hi", what is userStr after: strcpy(userStr, "Bye");

Check Show answer

3) What is userStr after: strcpy(userStr, "Goodbye");

Check Show answer

4) If userStr is initially "Hi!!", what is userStr after: strncpy(userStr, "Bye", 3);

Check Show answer

5) What is userStr after: strncpy(userStr, "Goodbye", 5);

Check Show answer

6) If userStr is initially "Hi", what is userStr after: strcat(userStr, "!");

Check Show answer

7) If userStr is initially "Hi", what is userStr after: strncat(userStr, "!");

Check Show answer

8) If userStr is initially "Hi", what is userStr after: strncat(userStr, "56789", 2);

Check Show answer

Feedback?

Several C string functions that get information about strings are summarized below.

Table 6.15.2: Some C string information functions.

Given:

```
char orgName[100] = "United Nations";
char usertext[20] = "UNICEF";
char targetText[10];
```

strchr()	strchr(sourceStr, searchChar)
	
	Returns NULL if searchChar does not exist in sourceStr. Else, returns address of first occurrence, discussed elsewhere). NULL is defined in the cstring library.
		`if (strchr(orgName, 'U') != NULL) { // "U" exists in orgName}`
		`// "U" exists in "United Nations", branch taken`
		
		`if (strchr(orgName, 'x') != NULL) { // "x" exists in orgName}`
		`// "x" doesn't exist case sensitive, branch not taken`

strlen()	size_t strlen(sourceStr)
	
	Returns number of characters in sourceStr up to, but not including, first null character. size_t is integer type.
		`a = strlen(orgName);  // Assigns 14 to a`
		`a = strlen(usertext);  // Assigns 6 to a`
		`a = strlen(targetText);  // Error: targetText may lack null char`

strcmp()	int strcmp(str1, str2)
	
	Returns 0 if str1 and str2 are equal, non-zero if they differ.
		`if (strcmp(orgName, "United Nations") == 0) {`
		`// Equal, branch taken}`
		
		`if (strcmp(orgName, usertext) != 0) {`
		`// Not equal, branch taken}`

Feedback?

strcmp() is usually used to compare for equality, returning 0 if the strings are the same length and have identical characters. A common error is to use == when comparing C strings, which does not work: str1 == str2 compares the strings' addresses, not their contents. Because those addresses will usually be different, str1 == str2 will evaluate to 0. This is not a syntax error, but clearly a logic error. Another common error is to forget to compare the result of strcmp with 0, as in `if (strcmp(str1, str2)) {...}`. The code is not a syntax error, but is a logic error because the if condition will be false (0) when the strings are equal. The correct condition would instead be `if (strcmp(str1, str2) == 0) {...}`. Although strcmp returns 0, a good practice is to avoid using `if (!strcmp(str1,str2)) {...}` because that 0 does not represent "false" but rather is encoding a particular situation.

strcmp(str1, str2) returns a negative number if str1 is less than str2, and a positive number if str1 is greater than str2. Evaluation first compares the character pair at element 0, then at element 1, etc., returning as soon as a pair differs.

PARTICIPATION ACTIVITY 6.15.2: String comparison.

Feedback?

strlen is often used to iterate through each string character in a loop:

Figure 6.15.1: Iterating through a C string using strlen.

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

int main() {
  char usernameStr[15] = "Alan Turing";
  int i = 0;

  cout << "Before: " << usernameStr << endl;

  for (i = 0; i < strlen(usernameStr); ++i) {
    if (usernameStr[i]==' ')
      usernameStr[i] = '_';
    else
      ;
  }

  cout << "After: " << usernameStr << endl;

  return 0;
}
```

```
Before: Alan Turing
After: Alan_Turing
```

Feedback?

PARTICIPATION ACTIVITY 6.15.3: Some C string library functions.

Feedback?

PARTICIPATION ACTIVITY 6.15.4: String information functions.

Given:

```
char str1[13] = "Earth";
char str2[25] = "Earth's moon";
char str3[13] = "Mars";
```

Answer the following questions. If appropriate, type: Error

1) What does strlen(str1) return?

Check Show answer

2) Is the branch taken? (Yes/No/Error)

```
if (strchr(str1, 'E') != NULL) {
  // Print "Found E"
}
```

Check Show answer

3) Is the branch taken? (Yes/No/Error)

```
if (strchr(str1, 'E') != NULL) {
  // Print "Found E"
}
```

Check Show answer

4) Is the branch taken? (Yes/No/Error)

```
if (strchr(str2, "Earth") != NULL) {
  // Print "Found Earth"
}
```

Check Show answer

5) Is the branch taken? (Yes/No/Error)

```
if (strcmp(str1, str2) == 0) {
  // Print "Strings are equal"
}
```

Check Show answer

6) Is the branch taken? (Yes/No/Error)

```
if (str1 == str3) {
  // Print "Strings are equal"
}
```

Check Show answer

7) Finish the code to take the branch if str1 and str3 are equal.

```
if (strcmp(str1, str3)  [ ] [ ]
  // Strings are equal
}
```

Check Show answer

Feedback?

CHALLENGE ACTIVITY 6.15.1: C-string library functions.

coin-toss-v-cstyle

Start

Variables stateStr, symbolStr, and numChars are assigned with a string for a state name, a string of symbols, and an integer read from input. numChars represents the number of characters to copy. Copy numChars characters from stateStr to the beginning of symbolStr.

Ex. if the input is:

```
Wisconsin
=========
5
```

then the output is:

```
Wisco====
```

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

int main() {
  char stateStr[80];
  char symbolStr[80];
  int numChars;

  cin >> stateStr;
  cin >> symbolStr;
  cin >> numChars;

  /* Your code goes here */

  cout << symbolStr << endl;

  return 0;
}
```

1 2 3 4

Check Next level

Feedback?

Exploring further:

- More C string functions from cplusplus.com

How was this section? Provide section feedback

6.16 Char library functions: cctype