3.15 More string operations

Finding in a string / Getting a substring

The string library provides numerous useful functions, including functions for finding a character or string within a string, or getting a substring of a string.

Table 3.15.1: find() and substr() functions, invoked as myString.find().

find()	***find***(item) returns index of first item occurrence, else returns string::npos (a constant defined in the string library). Item may be char, string variable, string literal (or char array).
	
	find(item, indx) starts at index indx.
		`// userText is "Help me!"`
		`userText.find('p')    // Returns 3`
		`userText.find('e')    // Returns 1 (first occurrence of e only)`
		`userText.find('z')    // Returns string::npos`
		`userText.find("me")   // Returns 5`
		`userText.find('e', 2) // Returns 6 (starts at index 2)`

substr()	***substr***(index, len) returns substring starting at index and having len characters.	`// userText is "http://google.com"`
		`userText.substr(0, 7)                    // Returns "http://"`
		`userText.substr(13, 4)                   // Returns ".com"`
		`userText.substr(userText.size() - 4, 4)   // Last 4: ".com"`

Feedback?

Figure 3.15.1: Example: Get username from email address.

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

int main() {
   string emailText;
   int atSymbolIndex;
   string emailUsername;

   cout << "Enter email address: ";
   cin  >> emailText;

   atSymbolIndex = emailText.find('@');
   if (atSymbolIndex == string::npos) {
      cout << "Address is missing @" << endl;
   }
   else {
      emailUsername = emailText.substr(0,
atSymbolIndex);
      cout << "Username: " << emailUsername <<
endl;
   }

   return 0;
}
```

```
Enter email address:
AbeLincoln@fakeemail.com
Username: AbeLincoln

...

Enter email address: swimming_is_fun
Address is missing @
```

Feedback?

PARTICIPATION ACTIVITY 3.15.1: find() and substr().

userText is "March 17, 2034".
Do not type quotes in answers.

1) What does userText.find(',') return?

8

Check Show answer

Correct

8

',' is the 9th character, so the index is 8.

2) What does userText.find("April") return?

string::npos

Correct

string::npos

Check Show answer

This strange-looking value is a constant integer defined in the string library. If int x = userText.find("April"), one might then type `if (x != string::npos) { ... }` to proceed only if the item was found.

3) What does userText.substr(0, 3) return?

Mar

Check Show answer

Correct

Mar

The substr function returns a string of length 3 starting with the character at index 0, or M.

4) What does userText.substr(userText.size() - 4, 4) return?

2034

Check Show answer

Correct

2034

That common form can be used to get the last N characters (in this case, the last 4 characters). The length is 14, so 14 - 4 is 10. Thus, the gotten substring is from index 10 to 13 (10, 11, 12, 13 is 4 characters), which are the characters 2034.

5) In the above email username example, what was the second argument passed to substr()?

atSymbolIndex

Check Show answer

Correct

atSymbolIndex

The find() function got the index of @. If found (i.e. not string::npos), then that index was used to indicate the size of the sub-string preceding the @. So for AbeLincoln@fakeemail.com, the @ is at index 10, which is the number of letters in AbeLincoln. Thus, substr() starts at index 0 and gets 10 characters, namely "AbeLincoln".

Feedback?

Combining / Replacing

The string library has more functions for modifying strings.

Table 3.15.2: String modify functions, invoked as myString.push_back(c). Each increases/decreases string's length appropriately.

push_back()	***push_back***(c) appends character c to the end of a string.	`// userText is "Hello"`
		`userText.push_back('?'); // Now "Hello?"`
		`userText.size();  // Returns 6`

insert()	***insert***(indx, subStr) Inserts string subStr starting at index indx.	`// userText is "Goodbye"`
		`userText.insert(0, "Well "); // Now "Well Goodbye"`
		
		`// userText is "Goodbye"`
		`userText.insert(4, "---");  // Now "Good---bye"`

replace()	***replace***(indx, num, subStr) replaces characters at indices indx to indx+num-1 with a copy of subStr.	`// userText is "You have many gifts"`
		`userText.replace(9, 4, "a plethora of");`
		`// Now "You have a plethora of gifts"`

str1 + str2	Returns a new string that is a copy of str1 with str2 appended.
	
	If one of str1, str2 is a string, the other may be a character (or character array).
		`// userText is "A B"`
		`myString = userText + " C D";`
		`// myString is "A B C D"`
		`myString = myString + '!';`
		`// myString now "A B C D!"`
		`myString = myString + userText;`
		`// myString now "A B C D!A B"`

Feedback?

Figure 3.15.2: String modify example: Greeting.

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

int main() {
   string userName;
   string greetingText;
   int itemIndex;

   itemIndex = 0;

   cout << "Enter name: ";
   getline(cin, userName);

   // Combine strings using +
   greetingText = "Hello " + userName;

   // Append a period (could have used +)
   greetingText.push_back('.'); // '' not ""
   cout << greetingText << endl;

   // Insert Mr/Ms before user's name
   greetingText.insert(6, "Mr/Ms ");
   cout << greetingText << endl;

   // Replace occurrence of "Darn" by "@$#"
   if (greetingText.find("Darn") != string::npos) { // Found
      itemIndex = static_cast<int>(greetingText.find("Darn"));
      greetingText.replace(itemIndex, 4, "@#$");
   }
   cout << greetingText << endl;

   return 0;
}
```

```
Enter name: Julia
Hello Julia.
Hello Mr/Ms Julia.
Hello Mr/Ms Julia.

...

Enter name: Darn Rabbit
Hello Darn Rabbit.
Hello Mr/Ms Darn Rabbit.
Hello Mr/Ms @#$ Rabbit.
```

Feedback?

PARTICIPATION ACTIVITY 3.15.2: String modification functions.

str1 is "Main" and str2 is " Street" (note the leading space).

1) Use + to combine str1 and str2, so newStr should be "Main Street".

```
newStr = str1
+ str2
;
```

Correct

+ str2

+ appends str2 to str1 in a new string.

Check Show answer

2) Use push_back to append period to str2, so str2 should be " Street."

Correct

push_back('.')

```
str2.push_back('.')
;
```

push_back appends a period to str2. Note the use of single quotes.

Check Show answer

3) Replace "ai" with "our" in str1, so str1 should be "Mourn". The first two arguments are just numbers.

```
str1.replace(1, 2, "our"
);
```

Correct

1, 2, "our"

1 is where "ai" starts in str1. 2 is how many chars to replace. "our" is then inserted there.

Check Show answer

Feedback?

Exploring further:

Numerous additional functions exist for strings.

- C++ string library

CHALLENGE ACTIVITY 3.15.1: Combining strings.

Assign secretID with firstName, a space, and lastName. Ex: If firstName is Barry and lastName is Allen, then output is:

```
Barry Allen
```

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6     string secretID;
7     string firstName;
8     string lastName;
9
10    cin >> firstName;
11    cin >> lastName;
12
```

```
   /* Your solution goes here  */
   secretID = firstName + " " + lastName;

   cout << secretID << endl;
   return 0;
}
```

Run ✓ All tests passed

✓ Testing: "Barry", "Allen"

Your output `Barry Allen`

✓ Testing: "Steve", "Austin"

Your output `Steve Austin`

✓ Testing: "Selina", "Kyle"

Your output `Selina Kyle`

Feedback?

CHALLENGE ACTIVITY 3.15.2: Name song.

Modify songVerse to play "The Name Game" (see OxfordDictionaries.com), by replacing "(Name)" with userName but without the first letter.

Ex: If userName = "Katie" and songVerse = "Banana-fana fo-f(Name)!", the program prints:

```
Banana-fana fo-fatie!
```

Ex: If userName = "Katie" and songVerse = "Fee fi mo-m(Name)", the program prints:

```
Fee fi mo-matie
```

Note: You may assume songVerse will always contain the substring "(Name)".

```
using namespace std;

int main() {
```

```
   string userName;
   string songVerse;

   getline(cin, userName);
   userName = userName.substr(1, userName.size() - 1);  // Remove first character

   getline(cin, songVerse);

   // Modify songVerse to replace (Name) with userName without first character

   /* Your solution goes here  */

      songVerse.replace(songVerse.find('('), 6, userName);

   cout << songVerse << endl;

   return 0;
}
```

Run ✓ All tests passed

✓ Testing Katie and Banana-fana fo-f(Name)!

Your output `Banana-fana fo-fatie!`

✓ Testing Walter and Banana-fana fo-f(Name)!

Your output `Banana-fana fo-falter!`

✓ Testing Katie and Fee fi mo-m(Name)

Your output `Fee fi mo-matie`

Feedback?

CHALLENGE ACTIVITY 3.15.3: Using find().

Print "Censored" if userInput contains the word "darn", else print userInput. End with newline. Ex: If userInput is "That darn cat.", then output is:

```
Censored
```

Ex: If userInput is "Dang, that was scary!", then output is:

```
Dang, that was scary!
```

Note: If the submitted code has an out-of-range access, the system will stop running the code after a few seconds, and report "Program end never reached." The system doesn't print the test case that caused the reported message.

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

int main() {
   string userInput;

   getline(cin, userInput);

   /* Your solution goes here  */
   //cout << substr(userInput.find('d'), 4);
   if(userInput.substr(userInput.find('d'), 4) == "darn" || userInput.substr(userInput.f
   {
      cout << "Censored";
   }else{
      cout << userInput;
   }

   cout << endl;
   return 0;
```

Run

✓ Testing for: That darn cat.

Your output `Censored`

✗ Test aborted

```
Exited with return code -6 (SIGIOT).
terminate called after throwing an instance of 'std::out_of_range
  what():  basic_string::substr: __pos (which is 1844674407370955
```

Feedback?