C++ STRING CLASS

Dalla pagina web: http://www.bgsu.edu/departments/compsci/docs/string.html

In order to use the C++ Standard Template Library (STL) string class you should include the following statements: `#include <string>`

The following examples assume these declarations and initial values for each:

```
string s = "abc def abc";
string s2 = "abcde uvwxyz";
char c;
char ch[] = "aba daba do";
char *cp = ch;
unsigned int i;
```

Stream input	`cin >> s;`	Changes the value of s to the value read in. The value stops at whitespace.
Stream output	`cout << s;`	Writes the string to the specified output stream.
Line input	`getline(cin, s);`	Reads everything up to the next newline character and puts the result into the specified string variable.

Assignment	`s = s2;s = "abc";`
	`s = ch; s = cp;`
		A string literal or a string variable or a character array can be assigned to a string variable. The last two assignments have the same effect.

Subscript	`s[1] = 'c';`
	`c = s[1];`
		Changes s to equal "acc def abc"
		Sets c to 'b'. The subscript operator returns a char value, not a string value.

Length	`i = s.length();`
	`i = s.size();`
		Either example sets i to the current length of the string s

Empty?	`if(s.empty()) i++;`
	`if(s == "") i++;`
		Both examples add 1 to i if string s is now empty

Relational operators	`if (s < s2) i++;`	Uses ASCII code to determine which string is smaller. Here the condition is true because a space comes before letter d

Concatenation	`s2 = s2 + "x";`
	`s2 += "x";`
		Both examples add x to the end of s2

Substring	`s = s2.substr(1,4);`
	`s = s2.substr(1,50);`
		The first example starts in position 1 of s2 and takes 4 characters, setting s to "bcde". In the second example, s is set to "bcde uvwxyz". If the length specified is longer than the remaining number of characters, the rest of the string is used. The first position in a string is position 0.

Substring replace	`s.replace(4,3,"x");`	Replaces the three characters of s beginning in position 4 with the character x. Variable s is set to "abc x abc".

Substring removal	`s.erase(4,5);`
	`s.erase(4);`
		Removes the five characters starting in position 4 of s. The new value of s is "abc bc".
		Remove from position 4 to end of string. The new value of s is "abc ".

Character array to string	`s = ch;`	Converts character array ch into string s.
String to character array	`cp = s.c_str();`	Pointer cp points to a character array with the same characters as s.

Pattern matching	`i = s.find("ab",4);`
	
	`if(s.rfind("ab",4)`
	`!= string::npos)`
	`   cout << "Yes" <<`
	`endl;`
		The first example returns the position of the substring "ab" starting the search in position 4. Sets i to 8. The find and rfind functions return the unsigned int **string::npos** if substring not found. The second example searches from right to left starting at position 4. Since the substring is found, the word Yes is printed.