mystring.append(str); Appends a copies of 'z' to mystring. mystring.append(str); Appends at to mystring.str can be a string object or character array. mystring.append(str, n); The first n characters from str, starting at position x, are appended to mystring. mystring.append(str, x, n); n number of characters from str, starting at position x, are appended to mystring. If mystring is too small, the function will copy as many characters as possible. mystring.assign(str); Assigns n copies of 'z' to mystring. mystring.assign(str, n); The first n characters from str, starting at position x, are assigned to mystring. mystring.assign(str, x, n); n number of characters from str, starting at position x, are assigned to mystring. mystring.assign(str, x, n); n number of characters from str, starting at position x, are assigned to mystring. If mystring is too small, the function will copy as many characters as possible. mystring.ask(); Returns the character in the string. (This member function was introduced in C++ 11.) mystring.begin(); Returns the character in the string. (This member function was introduced in C++ 11.) mystring.compare(str); Compares mystring to the first character in the string. (For more information on iterators, see Chapter 16.) mystring.compare(str); Returns the size of the storage allocated for the string. mystring.compare(str); Clears the string by deleting all the characters stored in it. mystring.compare(str); Compares mystring and str, starting at position x, and continuing for a character. The return value is like stremp, str can be a string object or a character array. mystring.enpety(); Returns the first position at or beyond position x, and continuing for a characters. If mystring is too small, the function will copy as many characters as possible. mystring.find('z', x); Returns the first position at or beyond position x where the string str is found in mystring is too small, the function will copy as many characters as possible. mystring.find('z', x); Returns the first position at o	Member Function Example	Description
character array. The first n characters of the character array str are appended to mystring. The first n characters from str, starting at position x, are appended to mystring. If mystring is too small, the function will copy as many characters as possible. Mystring.assign(str); Mystring.assign(str); Mystring.assign(str, n); Mystring.assign(str, n); Mystring.assign(str, x, n); Mystring	<pre>mystring.append(n, 'z');</pre>	
to mystring. mystring.append(str, x, n); number of characters from str, starting at position x, are appended to mystring. If mystring is too small, the function will copy as many characters as possible. mystring.assign(str); Assigns no copies of 'z' to mystring. mystring.assign(str, n); The first n characters of the character array str are assigned to mystring. mystring.assign(str, x, n); n number of characters from str, starting at position x, are assigned to mystring. If mystring is too small, the function will copy as many characters as possible. mystring.dack(); Returns the character at position x in the string. mystring.back(); Returns the character in the string. (This member function was introduced in C++ 11.) mystring.cstr(); Converts the contents of mystring to a C-string, mystring.capacity(); Returns an iterator pointing to the first character in the string.(For more information on iterators, see Chapter 16.) mystring.compare(str); Clears the string by deleting all the characters stored in it. Performs a comparison like the strcmp function (see Chapter 4), with the same return values as trean be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like strcmp str can be a string object or character array. mystring.empty(); Returns to if mystring is too small, the function will copy as many characters as possible. mystring.find(str, x); Returns true if mystring is too small, the function will copy as many characters as possible. Returns true if mystring is too small, the function will copy as many characters are position x mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.insert(x, n, 'z'); Inserts a copy of str into mystring, beginning at position x before the string in mystring to a character array. mystring.r	<pre>mystring.append(str); .</pre>	
appended to mystring is too small, the function will copy as many characters as possible. mystring.assign(str); Assigns n copies of 'z' to mystring. mystring.assign(str); Assigns str to mystring. str can be a string object or character array. mystring.assign(str, x, n); The first n characters of the character array str are assigned to mystring. mystring.assign(str, x, n); n number of characters from str, starting at position x, are assigned to mystring. If mystring is too small, the function will copy as many characters as possible. Mystring.at(x); Returns the last character in the string. This member function was introduced in C++ 11.) mystring.begin(); Returns the interactor pointing to the first character in the string. (For more information on iterators, see Chapter 16.) mystring.capacity(); Returns the size of the storage allocated for the string. mystring.compare(str); Clears the string by deleting all the characters stored in it. mystring.compare(str); Clears the string by deleting all the characters stored in it. Performs a comparison like the streng function (see Chapter 4), with the same return value. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp, str can be a string object or character array. mystring.empty(); Returns the mine the first character of the string in mystring.empty(); Returns the instruction will copy as many characters as possible. mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring, beginning at position x. mystring.find(str, x); Returns the first position at or beyond position x where the string at a first position at or beyond position x where the string at a first position at or begond position x. mystring.insert(x, n, 'z'); Inserts 'z' in times into mystring, beginning at position x. mystring.neplace(x, n, str); Returns the first position at or beyond position x.	<pre>mystring.append(str, n);</pre>	
Assigns str to mystring. str can be a string object or character array. The first n characters of the character array str are assigned to mystring. mystring.assign(str, x, n); mystring.begin(); mystring.begin(); mystring.begin(); mystring.cstr(); mystring.cstr(); mystring.capacity(); mystring.capacity(); mystring.compare(str); mystring.compare(str); mystring.compare(str); mystring.compare(str); mystring.compare(x, n, str); mystring.compare(x, n, str); mystring.compare(x, n, str); mystring.compare(x, n, str); mystring.empty(); mystring.empty(); mystring.empty(); mystring.empty(); mystring.empty(); mystring.empty(); mystring.erase(x, n); mystring.find(str, x); mystring.find(str, x); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.replace(x, n, str); mystring.replace(x, n, str); mystring.replace(x, n, str); mystring.replace(x, n, str); mystring.resize(n, 'z'); mystring.resize(n, 'z	<pre>mystring.append(str, x, n);</pre>	n number of characters from str, starting at position x, are appended to mystring. If mystring is too small, the function
character array. The first n characters of the character array str are assigned to mystring. mystring.assign(str, x, n); n number of characters from str, starting at position x, are assigned to mystring. If mystring is too small, the function will copy as many characters as possible. mystring.back(); Returns the character at position x in the string. mystring.begin(); Returns the last character in the string. (This member function was introduced in C++ 11.) mystring.c_str(); Returns an iterator pointing to the first character in the string. (For more information on iterators, see Chapter 16.) mystring.capacity(); Returns the size of the storage allocated for the string. mystring.compare(str); Clears the string by deleting all the characters stored in it. Performs a comparison like the stremp function (see Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp. str can be a string object or character array. mystring.copy(str, x, n); Copies the character array at to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. mystring.empty(); Returns the first position at or beyond position x where the string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where the string object or a character array. mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.insert(x, n, str); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.replace(x, n, str); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object or a character array. mystring.replace(x, n, str); Replaces the n characters	mystring.assign(n, 'z');	
mystring.assign(str, x, n); mystring.assign(str, x, n); mystring.at(x); mystring.back(); mystring.back(); mystring.begin(); mystring.cstr(); mystring.cstr(); mystring.cstr(); mystring.compare(str); mystring.compare(str); mystring.compare(x, n, str); mystring.compare(x, n, str); mystring.compare(x, n, str); mystring.empty(); mystring.empty(); mystring.empty(); mystring.empty(); mystring.compare(x, n, str); mystring.empty(); mystring.empty(); mystring.compare(x, n, str); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, n, str); mystring.replace(x, n, str);	<pre>mystring.assign(str);</pre>	_
assigned to mystring. If mystring is too small, the function will copy as many characters as possible. Returns the character at position x in the string. Returns the last character in the string. (This member function was introduced in C++ 11.) mystring.begin(); Returns an iterator pointing to the first character in the string. (For more information on iterators, see Chapter 16.) mystring.capacity(); Returns the contents of mystring to a C-string, and returns a pointer to the C-string. mystring.capacity(); Returns the size of the storage allocated for the string. Mystring.compare(str); Performs a comparison like the stremp function (see Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp. str can be a string object or character array. Copies the character array str to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. mystring.empty(); mystring.empty(); mystring.enae(x, n); mystring.find(str, x); mystring.find(str, x); mystring.find(str, x); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, n, str); mystring.insert(x, n, str); mystring.neplace(x, n, str); mystring.replace(x, n, str); Changes the site and the end enough times to fill the new spaces.	<pre>mystring.assign(str, n);</pre>	
mystring.back(); mystring.begin(); Returns the last character in the string. (This member function was introduced in C++ 11.) Returns an iterator pointing to the first character in the string. (For more information on iterators, see Chapter 16.) mystring.capacity(); Returns the contents of mystring to a C-string, and returns a pointer to the C-string. mystring.capacity(); Returns the size of the storage allocated for the string. Clears the string by deleting all the characters stored in it. mystring.compare(str); Performs a comparison like the stromp function (see Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stromp. str can be a string object or character array. mystring.copy(str, x, n); Copies the character array str to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. mystring.end(); Returns true if mystring is empty. Returns an iterator pointing to the last character of the string in mystring. (For more information on iterators, see Chapter 16.) Erases a characters from mystring, beginning at position x. Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. Returns the first position at or beyond position x where 'z' is found in mystring. mystring.insert(x, n, 'z'); Inserts a copy of str into mystring, beginning at position x str may be either a string object or a character array. Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, str); mystring.replace(x, n, str); Returns the length of the string in mystring to n. If n is less than the current size of the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fil	mystring.assign(str, x, n);	assigned to mystring. If mystring is too small, the function
function was introduced in C++ 11.) Returns an iterator pointing to the first character in the string. (For more information on iterators, see Chapter 16.) Mystring.c_str(); Mystring.clear(); Mystring.clear(); Mystring.compare(str); Mystring.compare(str); Mystring.compare(x, n, str); Mystring.end(); Mystring.end(); Mystring.end(); Mystring.find(str, x); Mystring.find(str, x); Mystring.find('z', x); Mystring.find('z', x); Mystring.find('z', x); Mystring.insert(x, n, 'z'); Mystring.insert(x, n, 'z'); Mystring.insert(x, n, str); Mystring.insert(x, n, str); Mystring.length(); Mystring.replace(x, n, str); Mystring.replace(x, n, str); Mystring.resize(n, 'z');	<pre>mystring.at(x);</pre>	Returns the character at position x in the string.
mystring.c_str(); mystring.capacity(); mystring.clear(); mystring.compare(str); mystring.compare(str); mystring.compare(str); mystring.compare(x, n, str); mystring.end(); mystring.end(); mystring.end(); mystring.erase(x, n); mystring.find(str, x); mystring.find(str, x); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.find('z', x); mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.length(); mystring.replace(x, n, str); mystring.replace(x, n, str); mystring.resize(n, 'z'); Converts the contents of mystring beginning at position x with the character in mystring beginning at position x with the character in the string in mystring. mystring.find('z', x); mystring.find('z	mystring.back();	
returns a pointer to the C-string. Returns the size of the storage allocated for the string. Clears the string by deleting all the characters stored in it. Performs a comparison like the stremp function (see Chapter 4), with the same return values. str can be a string object or a character array. Mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp str can be a string object or character array. Mystring.copy(str, x, n); Copies the character array str to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. Mystring.end(); Returns true if mystring is empty. Returns an iterator pointing to the last character of the string in mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. Mystring.front(); Returns the first position at or beyond position x where '2' is found in mystring. Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. Mystring.length(); Mystring.replace(x, n, str); Returns the length of the string in mystring. Returns the length of the string beginning at position x with the characters in string object or a character array. Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.begin();</pre>	
mystring.clear(); mystring.compare(str); Performs a comparison like the stremp function (see Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp. str can be a string object or character array. mystring.copy(str, x, n); Compares mystring object or character array. mystring.empty(); Returns the first position in mystring is too small, the function will copy as many characters as possible. mystring.empty(); Returns true if mystring is empty. mystring.erase(x, n); Returns an iterator pointing to the last character of the string in mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where the string str is found in mystring. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in mystring to n. If n is less than the current size of the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.c_str();</pre>	
Performs a comparison like the stremp function (see Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp. str can be a string object or character array. mystring.copy(str, x, n); Copies the character array str to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. mystring.empty(); Returns true if mystring is empty. mystring.erase(x, n); Returns an iterator pointing to the last character of the string in mystring. (For more information on iterators, see Chapter 16.) mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first position at or beyond position x where 'z' is found in mystring. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object or a character array. Mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	mystring.capacity();	Returns the size of the storage allocated for the string.
Chapter 4), with the same return values. str can be a string object or a character array. mystring.compare(x, n, str); Compares mystring and str, starting at position x, and continuing for n characters. The return value is like stremp. str can be a string object or character array. Mystring.copy(str, x, n); Copies the character array str to mystring, beginning at position x, for n characters. If mystring, beginning at position will copy as many characters as possible. Mystring.empty(); Returns true if mystring is empty. Mystring.erase(x, n); Returns an iterator pointing to the last character of the string in mystring. (For more information on iterators, see Chapter 16.) Mystring.find(str, x); Returns the first position at or beyond position x where the string object or a character array. Mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. Mystring.front(); Returns the first character in the string. (This member function was introduced in C++ 11.) Mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. Mystring.length(); Returns the length of the string in mystring. Mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. Mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	mystring.clear();	Clears the string by deleting all the characters stored in it.
continuing for n characters. The return value is like stromp. str can be a string object or character array. Copies the character array str to mystring, beginning at position x, for n characters. If mystring is too small, the function will copy as many characters as possible. Mystring.empty(); Mystring.end(); Mystring.erase(x, n); Mystring.find(str, x); Mystring.find(str, x); Mystring.find('z', x); Mystring.find('z', x); Mystring.front(); Mystring.insert(x, n, 'z'); Mystring.insert(x, n, 'z'); Mystring.length(); Mystring.replace(x, n, str); Mystring.resize(n, 'z'); Mystring.resize(n, 'z'); Changes the size of the atring in return value is like stromp. Str can be a string object or character array. Mystring.serting.serting beginning at position x. Mystring.length(); Mystring.resize(n, 'z'); Changes the characters. The return value is like stromp. Str can be a string object or mystring is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.compare(str);</pre>	Chapter 4), with the same return values. str can be a
position x, for n characters. If mystring is too small, the function will copy as many characters as possible. mystring.empty(); mystring.end(); Returns true if mystring is empty. Returns an iterator pointing to the last character of the string in mystring. (For more information on iterators, see Chapter 16.) mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.insert(x, n, 'z'); mystring.insert(x, n, 'z'); mystring.insert(x, str); mystring.length(); mystring.length(); mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.compare(x, n, str);</pre>	continuing for n characters. The return value is like stromp.
Returns an iterator pointing to the last character of the string in mystring.erase(x, n); mystring.erase(x, n); mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.length(); mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.copy(str, x, n);</pre>	position x, for n characters. If mystring is too small, the
mystring.(For more information on iterators, see Chapter 16.) mystring.erase(x, n); Erases n characters from mystring, beginning at position x. Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.empty();</pre>	Returns true if mystring is empty.
mystring.find(str, x); Returns the first position at or beyond position x where the string str is found in mystring. str may be either a string object or a character array. Mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. Mystring.front(); Returns the first character in the string. (This member function was introduced in C++ 11.) Mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. Mystring.insert(x, str); Inserts a copy of str into mystring, beginning at position x. str may be either a string object or a character array. Mystring.length(); Returns the length of the string in mystring. Mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. Mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.end();</pre>	
the string str is found in mystring. str may be either a string object or a character array. mystring.find('z', x); Returns the first position at or beyond position x where 'z' is found in mystring. mystring.front(); Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.insert(x, str); Inserts a copy of str into mystring, beginning at position x. str may be either a string object or a character array. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.erase(x, n);</pre>	Erases n characters from mystring, beginning at position x.
is found in mystring. Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.insert(x, str); Inserts a copy of str into mystring, beginning at position x. str may be either a string object or a character array. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.find(str, x);</pre>	the string str is found in mystring. str may be either a
Returns the first character in the string. (This member function was introduced in C++ 11.) mystring.insert(x, n, 'z'); Inserts 'z' n times into mystring at position x. mystring.insert(x, str); Inserts a copy of str into mystring, beginning at position x. str may be either a string object or a character array. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.find('z', x);</pre>	
mystring.insert(x, str); Inserts a copy of str into mystring, beginning at position x. str may be either a string object or a character array. mystring.length(); Returns the length of the string in mystring. mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.front(); .</pre>	Returns the first character in the string. (This member
<pre>x. str may be either a string object or a character array. mystring.length(); mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position</pre>	<pre>mystring.insert(x, n, 'z');</pre>	Inserts 'z' n times into mystring at position x.
mystring.replace(x, n, str); Replaces the n characters in mystring beginning at position x with the characters in string object str. mystring.resize(n, 'z'); Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.insert(x, str);</pre>	
x with the characters in string object str. Changes the size of the allocation in mystring to n. If n is less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	mystring.length();	Returns the length of the string in mystring.
less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z' is appended at the end enough times to fill the new spaces.	<pre>mystring.replace(x, n, str);</pre>	
the second of the second secon	<pre>mystring.resize(n, 'z');</pre>	less than the current size of the string, the string is truncated to n characters. If n is greater, the string is expanded and 'z'
mystring.size(); Returns the length of the string in mystring.	mystring.size();	equities leader leaf ena glet upibbld alul /4
mystring.substr(x, n); Returns a copy of a substring. The substring is n characters long and begins at position x of mystring.		Returns a copy of a substring. The substring is n characters
mystring.swap(str); Swaps the contents of mystring with str.	mystring.swap(str);	include ≤string> confidence in the confidence i