

The CharList class

In this assignment, your task is to implement a doubly-linked list of characters. You are not allowed to use library functions for this task. The class that stores the list should be called **CharList**, and it should have the following API:

```
bool empty();           // is the list empty?
int size();             // the number of elements in the list
void insert(char c);     // inserts the character c at the front of the list
void append(char c);     // inserts the character c at the back of the list
void insert(char c, char d); // inserts c immediately before the first occurrence of d, or
                           // at the end of the list if d does not occur.
void append(char c, char d); // inserts c immediately after the first occurrence of d, or
                           // at the end of the list if d does not occur.
void insert(char c, char d, int n); // inserts c immediately before the nth occurrence of d, or
                           // at the end of the list if d does not occur n times.
void append(char c, char d, int n); // inserts c immediately after the nth occurrence of d, or
                           // at the end of the list if d does not occur n times.
void remove(char c);     // delete the first occurrence of character c. Do nothing if
                           // c does not occur.
void remove(char c, int n); // delete the nth occurrence of c. Do nothing if c does not
                           // occur n times.
string toString();      // return the list as a string.
CharList(string s);      // constructor. Initializes the list to the contents of the string.
~CharList();             // destructor. Returns all memory in the list.
```

The constructor takes a string, such as “abc”, and makes a list with the characters of the string in order, such as ‘a’, then ‘b’, then ‘c’. The toString() method does the same in reverse—it constructs a string consisting of the characters of the list, in order.

You may add private member functions and member variables to the API. You may use default arguments if that makes implementation easier for you. You should have a CharList.h file and a CharList.cpp, and you may have any other classes and files that you like.

Testing

We will test your code using a main.cpp that we will not provide you with. To be sure that your code is correct, you should test it yourself. Include with your submission a main.cpp that creates a CharList, applies operations to it, and compares the result of these operations (via toString) with what you expect the result to be. If it is what you expect, print “passed”, and otherwise print “failed” along with the expected result string and the actual result string.

For instance, the following main creates a CharList and tests one function, printing the resulting string form of the CharList:

```
int main(int argc, char** argv) {
    CharList* c = new CharList("abd");
    c->insert('c', 'd');
    cout << "result: " << c.toString() << endl;
}
```

This code should print:

abcd

But you should test all functions, not just one. And test to be sure that functions are inserting at the back of the list when they should. Also, in your submission, only print the line with “passed” or “failed” on it. (During your debugging, you may want to print more.)

Submission

Use CourSys (<http://coursys.sfu.ca>) for submission of your assignment. You’ll need to create a zip file with your .cpp and .h files in it. Do not include compiled files (any .o files, a.out, or a.exe).