Start of the Write-up

Just showing the test of plusEquals, sadly I couldn't make it work.

It might be in my strcat function, tried different ways but couldn't make it work.

```
==26407==    by 0x401160: testPlusEquals() (MyString.cpp:448)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==
==26407== Invalid read of size 1
==26407==    at 0x4C30A56: __GI_mempcpy (vg_replace_strmem.c:1525)
==26407==    by 0x57447C4: _IO_file_xsputn@@GLIBC_2.2.5 (in /usr/lib64/libc-2.17.so)
==26407==    by 0x57395D1: fwrite (in /usr/lib64/libc-2.17.so)
==26407==    by 0x4F3EBED: sputn (streambuf:451)
==26407==    by 0x4F3EBED: __ostream_write<char, std::char_traits<char> > (ostream_insert.h:50)
==26407==    by 0x4F3EBED: std::basic_ostream<char, std::char_traits<char> >& std::__ostream_insert<char, std::cha
r_traits<char> >(std::basic_ostream<char, std::char_traits<char> >&, char const*, long) (ostream_insert.h:101)
==26407==    by 0x4F3EEF6: std::basic_ostream<char, std::char_traits<char> >& std::operator<< <std::char_traits<ch
ar> >(std::basic_ostream<char, std::char_traits<char> >&, char const*) (ostream:561)
==26407==    by 0x4024EF: MyString::print(std::ostream&) const (MyString.cpp:356)
==26407==    by 0x4025DF: operator<<(std::ostream&, MyString const&) (MyString.cpp:384)
==26407==    by 0x401181: testPlusEquals() (MyString.cpp:449)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==  Address 0x5aa8d22 is 2 bytes inside a block of size 6 free'd
==26407==    at 0x4C2B61D: operator delete[](void*) (vg_replace_malloc.c:621)
==26407==    by 0x401FC3: MyString::str2dup(char*, char*) const (MyString.cpp:127)
==26407==    by 0x40244C: MyString::operator+=(MyString const&) (MyString.cpp:337)
==26407==    by 0x401160: testPlusEquals() (MyString.cpp:448)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==  Block was alloc'd at
==26407==    at 0x4C2A888: operator new[](unsigned long) (vg_replace_malloc.c:423)
==26407==    by 0x401F19: MyString::str2dup(char*, char*) const (MyString.cpp:117)
==26407==    by 0x40244C: MyString::operator+=(MyString const&) (MyString.cpp:337)
==26407==    by 0x401160: testPlusEquals() (MyString.cpp:448)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==
result: Hello                      Plus equal
==26407== Invalid free() / delete / delete[] / realloc()
==26407==    at 0x4C2B61D: operator delete[](void*) (vg_replace_malloc.c:621)
==26407==    by 0x4025AA: MyString::~MyString() (MyString.cpp:376)
==26407==    by 0x4011B5: testPlusEquals() (MyString.cpp:443)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==  Address 0x5aa8d20 is 0 bytes inside a block of size 6 free'd
==26407==    at 0x4C2B61D: operator delete[](void*) (vg_replace_malloc.c:621)
==26407==    by 0x401FC3: MyString::str2dup(char*, char*) const (MyString.cpp:127)
==26407==    by 0x40244C: MyString::operator+=(MyString const&) (MyString.cpp:337)
==26407==    by 0x401160: testPlusEquals() (MyString.cpp:448)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==  Block was alloc'd at
==26407==    at 0x4C2A888: operator new[](unsigned long) (vg_replace_malloc.c:423)
==26407==    by 0x401F19: MyString::str2dup(char*, char*) const (MyString.cpp:117)
==26407==    by 0x40244C: MyString::operator+=(MyString const&) (MyString.cpp:337)
==26407==    by 0x401160: testPlusEquals() (MyString.cpp:448)
==26407==    by 0x4019D2: main (MyString.cpp:563)
==26407==
=====Testing Plus=====
MyString s('Hello')
MyString d('There')
Using s+d; expected: 'Hello There'
==26407== Invalid write of size 1
==26407==    at 0x402067: MyString::strcat(char*, char const*) const (MyString.cpp:144)
==26407==    by 0x401FAC: MyString::str2dup(char*, char*) const (MyString.cpp:124)
==26407==    by 0x402402: MyString::operator+(MyString const&) const (MyString.cpp:325)
==26407==    by 0x401028: testPlus() (MyString.cpp:437)
==26407==    by 0x4019F3: main (MyString.cpp:565)
==26407==  Address 0x5aa8e66 is 0 bytes after a block of size 6 alloc'd
==26407==    at 0x4C2A888: operator new[](unsigned long) (vg_replace_malloc.c:423)
==26407==    by 0x401F19: MyString::str2dup(char*, char*) const (MyString.cpp:117)
==26407==    by 0x402402: MyString::operator+(MyString const&) const (MyString.cpp:325)
==26407==    by 0x401028: testPlus() (MyString.cpp:437)
```

Here is the plus operator, doesn't work as well.

```
==26407==    by 0x401028: testPlus() (MyString.cpp:437)
==26407==    by 0x4019F3: main (MyString.cpp:565)
==26407==
result: Hello
```

plus

```
=====Testing Pattern=====
MyString s('My Name is buf')
MyString d('Name')
MyString h('Hello')
Using s.indexOf(d) expected output: 3
result: 3
Using s.indexOf(h) expected output: -1
result: -1

=====Testing Print=====
MyString s('hello')
cout<<s; expected: 'hello'
result: hello

=====Testing indexOf Character=====
MyString s('Hello')
using s.indexOf('o'); expected: 4
result: 4

=====Testing length=====
MyString s('Hello')
s.length(); expected: 5
result: 5

=====Testing reverse()=====
MyString s('hello there')
s.reverse() expected output: ereht olleh
result: ereht olleh

=====Testing compare()=====
MYString s('hello')
MyString h('hello')
MyString v('howdy')
Comparing between 'hello'(s) and 'hello'(h)
s==h; expected: True
result: True
Comparing between 'hello'(s) and 'howdy'(v)
s==v; expected: False
result: False

=====Testing Assignment=====
MyString s('Hello')
MyString h('Goodbye')
cout<<s; expected: 'Hello'
result: Hello
After assignment using s=h; expected: 'Goodbye'
result: Goodbye

=====Testing brackets=====
MyString s('hello')
using s[0]; expected: h
result: h
using s[5]; expected:
result:

=====Testing Input=====
MyString s;
cin>>s; expected user input
This was a hard first homework
result: This was a hard first homework
```

all good

All else tested well.

```
=====Testing indexOf Character=====
MyString s('Hello')
using s.indexOf('o'); expected: 4
result: 4

=====Testing length=====
MyString s('Hello')
s.length(); expected: 5
result: 5

=====Testing reverse()=====
MyString s('hello there')
s.reverse() expected output: ereht olleh
result: ereht olleh

=====Testing compare()=====
MYString s('hello')
MyString h('hello')
MyString v('howdy')
Comparing between 'hello'(s) and 'hello'(h)
s==h; expected: True
result: True
Comparing between 'hello'(s) and 'howdy'(v)
s==v; expected: False
result: False

=====Testing Assignment=====
MyString s('Hello')
MyString h('Goodbye')
cout<<s; expected: 'Hello'
result: Hello
After assignment using s=h; expected: 'Goodbye'
result: Goodbye

=====Testing brackets=====
MyString s('hello')
using s[0]; expected: h
result: h
using s[5]; expected:
result:

=====Testing Input=====
MyString s;
cin>>s; expected user input
This was a hard first homework
result: This was a hard first homework

Net memory allocated at program end: -23
(should be zero! positive = memory leak, negative = duplicate delete)
==26407==
==26407== HEAP SUMMARY:
==26407==     in use at exit: 72,704 bytes in 1 blocks
==26407==   total heap usage: 27 allocs, 27 frees, 72,904 bytes allocated
==26407==
==26407== LEAK SUMMARY:
==26407==    definitely lost: 0 bytes in 0 blocks
==26407==    indirectly lost: 0 bytes in 0 blocks
==26407==      possibly lost: 0 bytes in 0 blocks
==26407==    still reachable: 72,704 bytes in 1 blocks
==26407==         suppressed: 0 bytes in 0 blocks
==26407== Rerun with --leak-check=full to see details of leaked memory
==26407==
==26407== For counts of detected and suppressed errors, rerun with: -v
==26407== ERROR SUMMARY: 39 errors from 13 contexts (suppressed: 0 from 0)
-bash-4.2$
```

Did end up with errors but no memory leaks.

End of Write-up