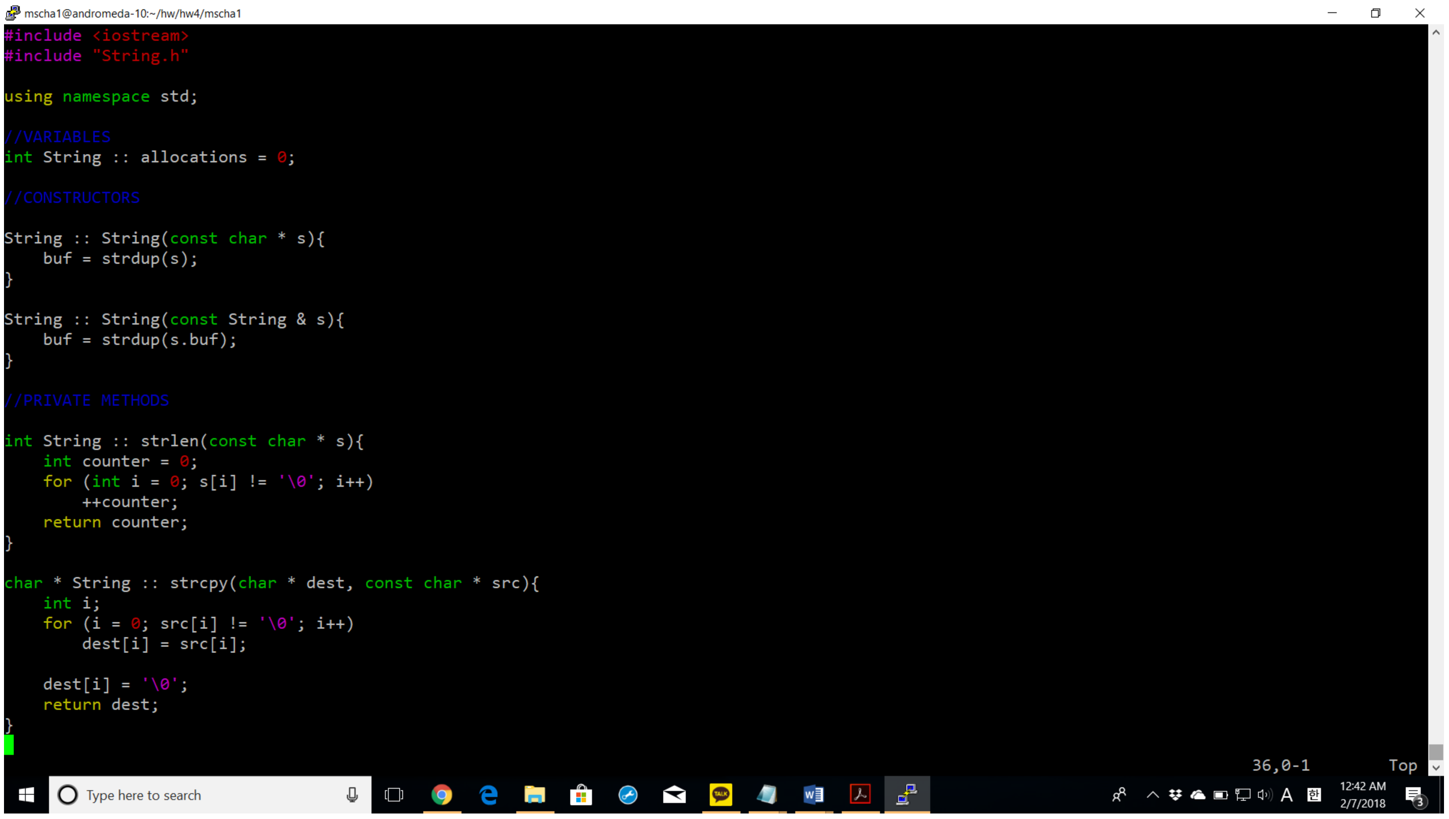
Name: Min Sung Cha Student ID: 85408485

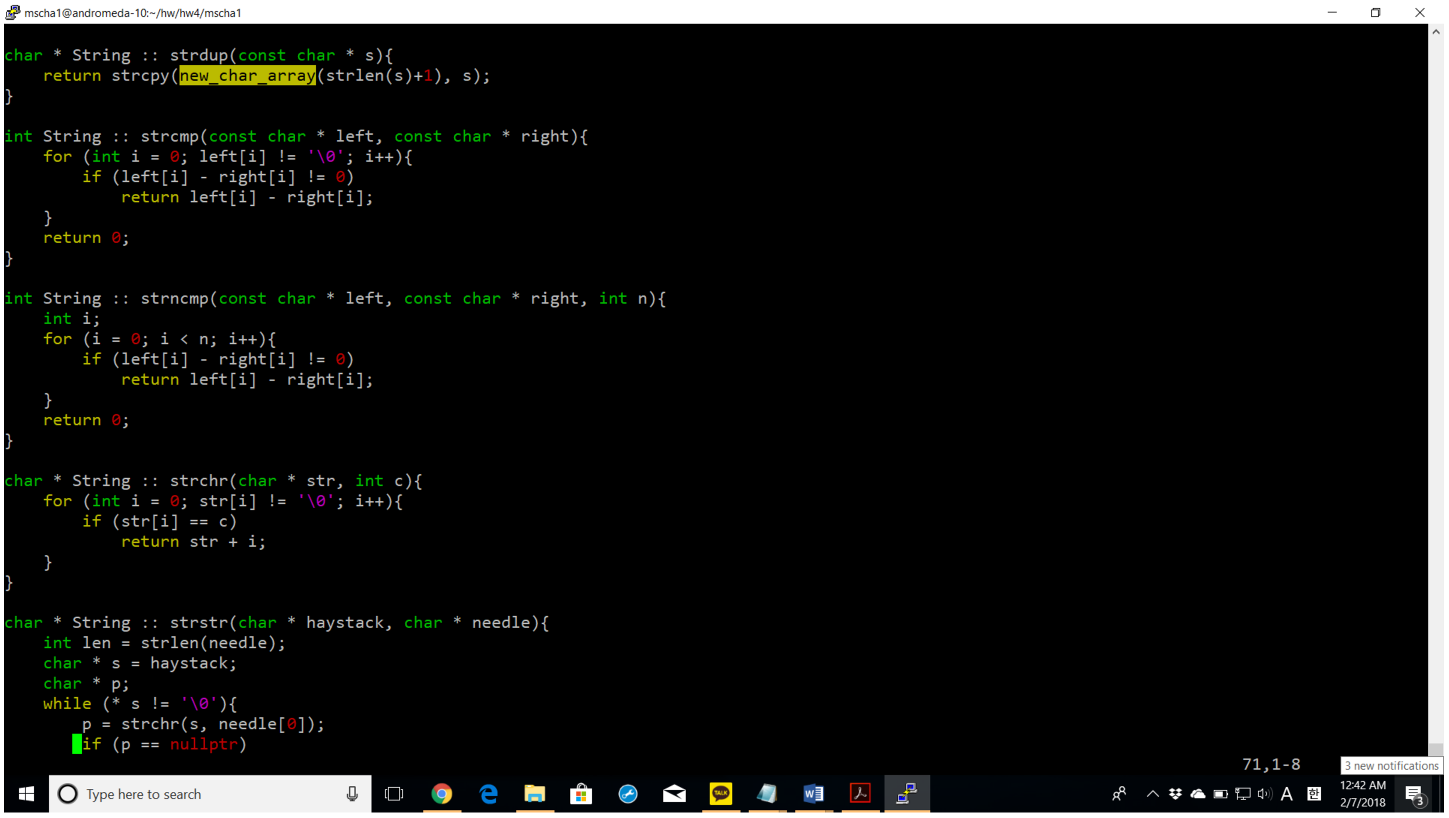
Analysis of a few functions and any embedded functions

String.cpp



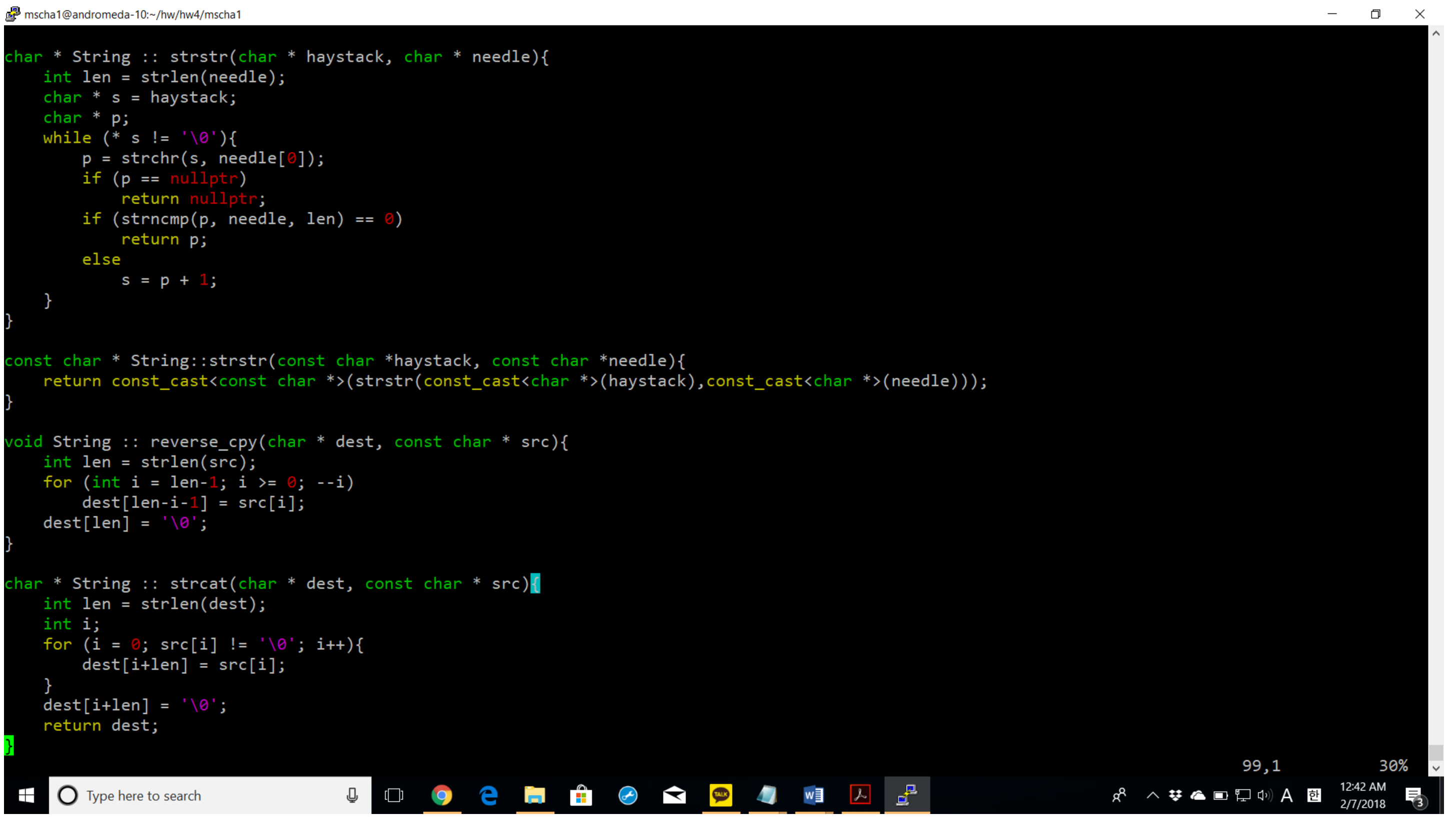
mscha1@andromeda-10:~/hw/hw4/mscha1

```
#include <iostream>
#include "String.h"

using namespace std;

//VARIABLES
int String :: allocations = 0;

//CONSTRUCTORS

String :: String(const char * s){
    buf = strdup(s);
}

String :: String(const String & s){
    buf = strdup(s.buf);
}

//PRIVATE METHODS

int String :: strlen(const char * s){
    int counter = 0;
    for (int i = 0; s[i] != '\0'; i++)
        ++counter;
    return counter;
}

char * String :: strcpy(char * dest, const char * src){
    int i;
    for (i = 0; src[i] != '\0'; i++)
        dest[i] = src[i];

    dest[i] = '\0';
    return dest;
}

                                                                                                            36,0-1        Top
```



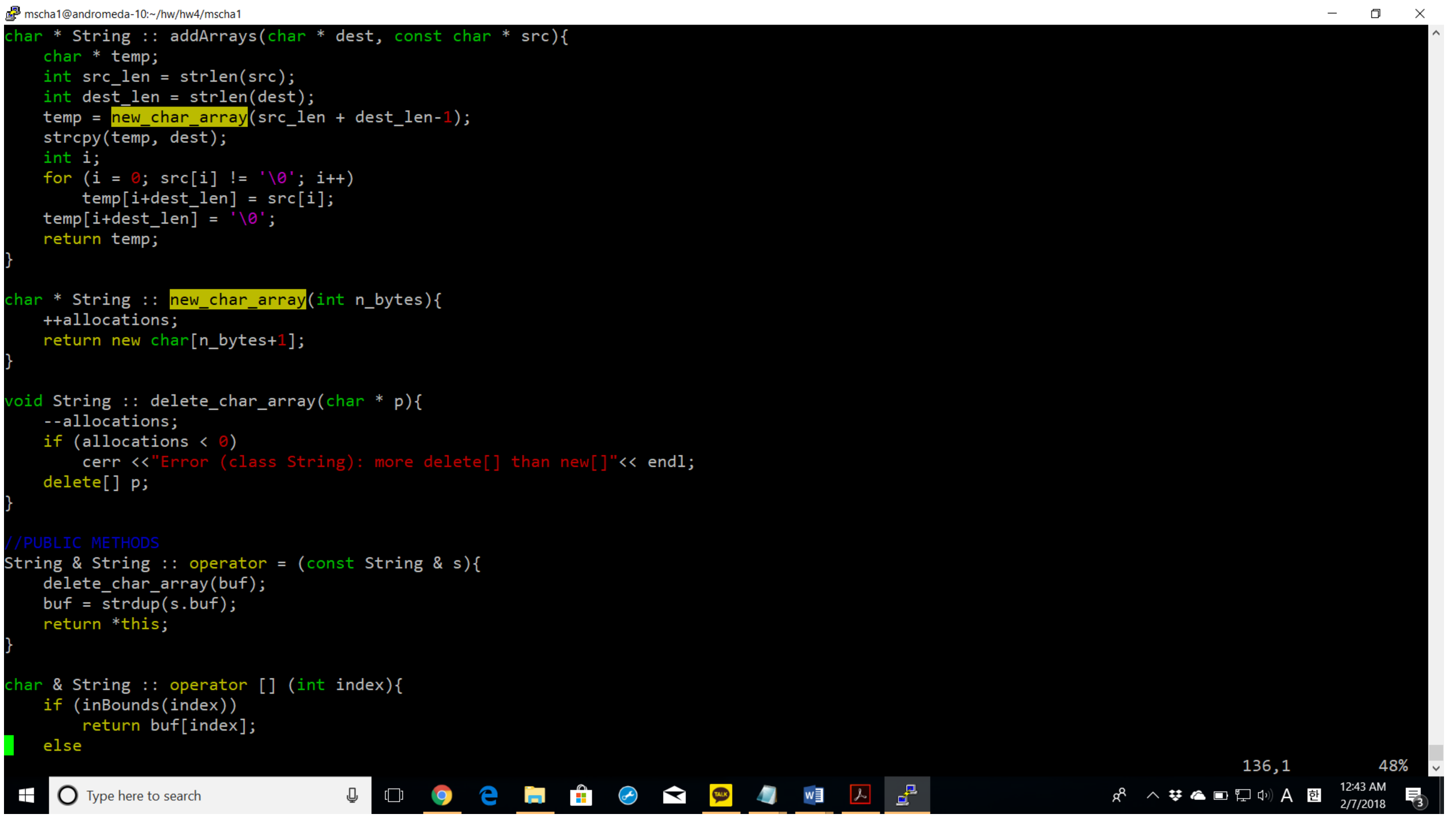
mscha1@andromeda-10:~/hw/hw4/mscha1

```
char * String :: strdup(const char * s){
    return strcpy(new_char_array(strlen(s)+1), s);
}

int String :: strcmp(const char * left, const char * right){
    for (int i = 0; left[i] != '\0'; i++){
        if (left[i] - right[i] != 0)
            return left[i] - right[i];
    }
    return 0;
}

int String :: strncmp(const char * left, const char * right, int n){
    int i;
    for (i = 0; i < n; i++){
        if (left[i] - right[i] != 0)
            return left[i] - right[i];
    }
    return 0;
}

char * String :: strchr(char * str, int c){
    for (int i = 0; str[i] != '\0'; i++){
        if (str[i] == c)
            return str + i;
    }
}

char * String :: strstr(char * haystack, char * needle){
    int len = strlen(needle);
    char * s = haystack;
    char * p;
    while (* s != '\0'){
        p = strchr(s, needle[0]);
        if (p == nullptr)
                                                                                                            71,1-8
```



mscha1@andromeda-10:~/hw/hw4/mscha1

```
char * String :: strstr(char * haystack, char * needle){
    int len = strlen(needle);
    char * s = haystack;
    char * p;
    while (* s != '\0'){
        p = strchr(s, needle[0]);
        if (p == nullptr)
            return nullptr;
        if (strncmp(p, needle, len) == 0)
            return p;
        else
            s = p + 1;
    }
}

const char * String::strstr(const char *haystack, const char *needle){
    return const_cast<const char *>(strstr(const_cast<char *>(haystack),const_cast<char *>(needle)));
}

void String :: reverse_cpy(char * dest, const char * src){
    int len = strlen(src);
    for (int i = len-1; i >= 0; --i)
        dest[len-i-1] = src[i];
    dest[len] = '\0';
}

char * String :: strcat(char * dest, const char * src){
    int len = strlen(dest);
    int i;
    for (i = 0; src[i] != '\0'; i++){
        dest[i+len] = src[i];
    }
    dest[i+len] = '\0';
    return dest;
}
                                                                  99,1          30%
```



mscha1@andromeda-10:~/hw/hw4/mscha1

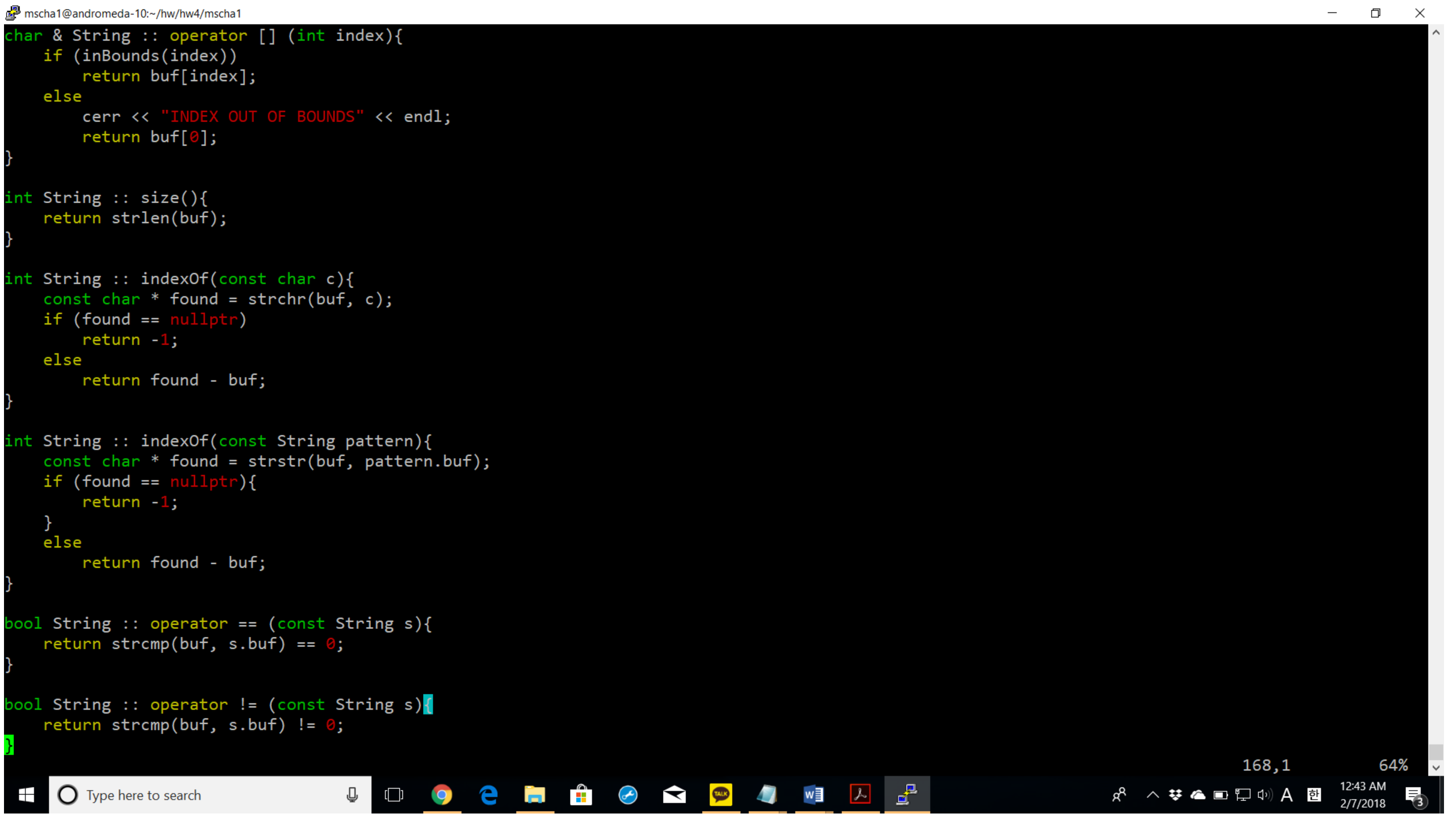
```
char * String :: addArrays(char * dest, const char * src){
    char * temp;
    int src_len = strlen(src);
    int dest_len = strlen(dest);
    temp = new_char_array(src_len + dest_len-1);
    strcpy(temp, dest);
    int i;
    for (i = 0; src[i] != '\0'; i++)
        temp[i+dest_len] = src[i];
    temp[i+dest_len] = '\0';
    return temp;
}

char * String :: new_char_array(int n_bytes){
    ++allocations;
    return new char[n_bytes+1];
}

void String :: delete_char_array(char * p){
    --allocations;
    if (allocations < 0)
        cerr <<"Error (class String): more delete[] than new[]"<< endl;
    delete[] p;
}

//PUBLIC METHODS
String & String :: operator = (const String & s){
    delete_char_array(buf);
    buf = strdup(s.buf);
    return *this;
}

char & String :: operator [] (int index){
    if (inBounds(index))
        return buf[index];
    else
                                                                  136,1         48%
```

addArrays: creates a temporary pointer that points to a new array of the size of both dest and src arrays and dest and src are copied inside the temporary pointer.

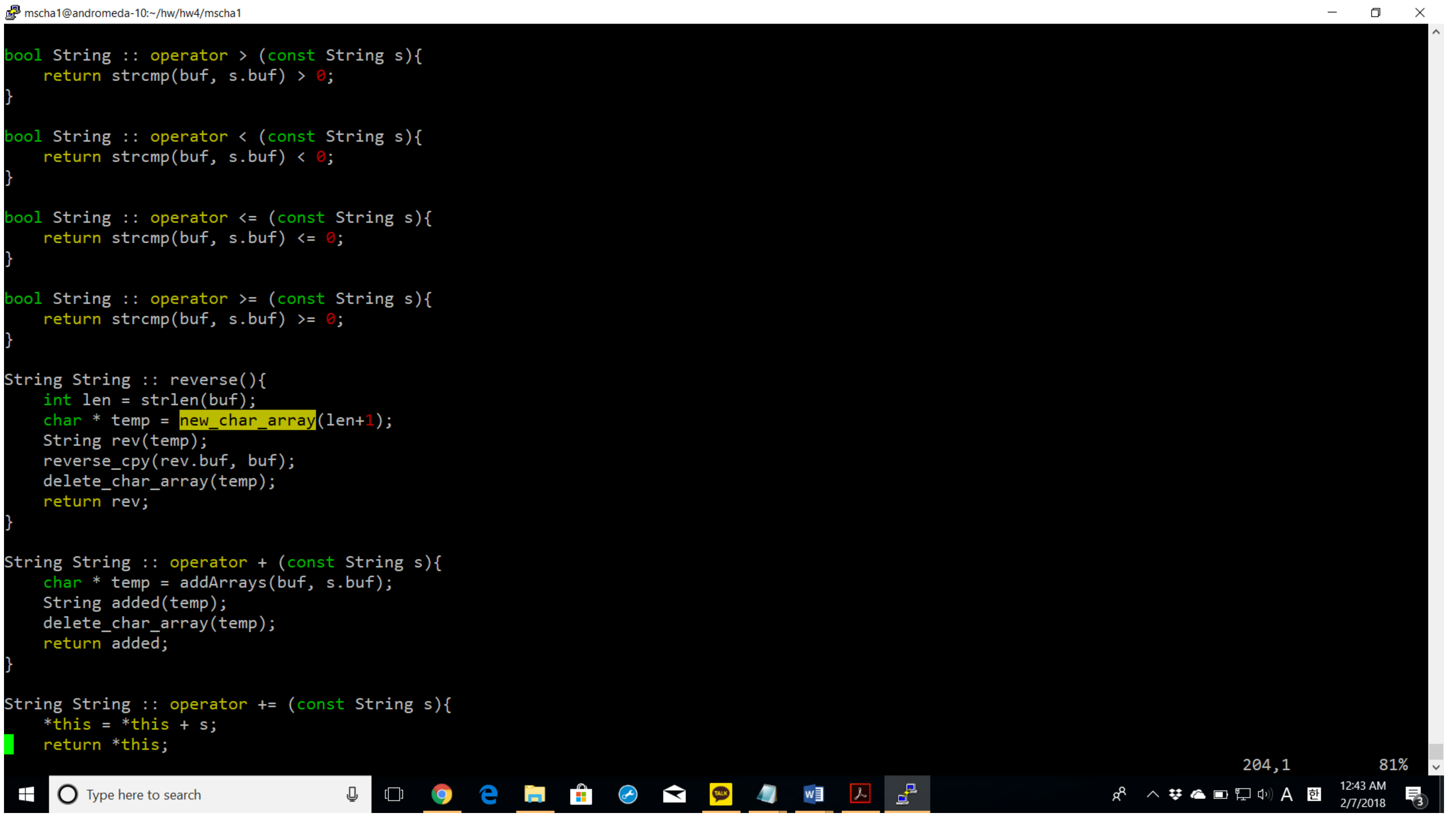
new_char_array: creates a new array of characters of the size passed in the parameter.

delete_char_array: deletes the array specified in the parameter.

operator = : deletes the current array and assigns it to the array of the String passed in the parameter.

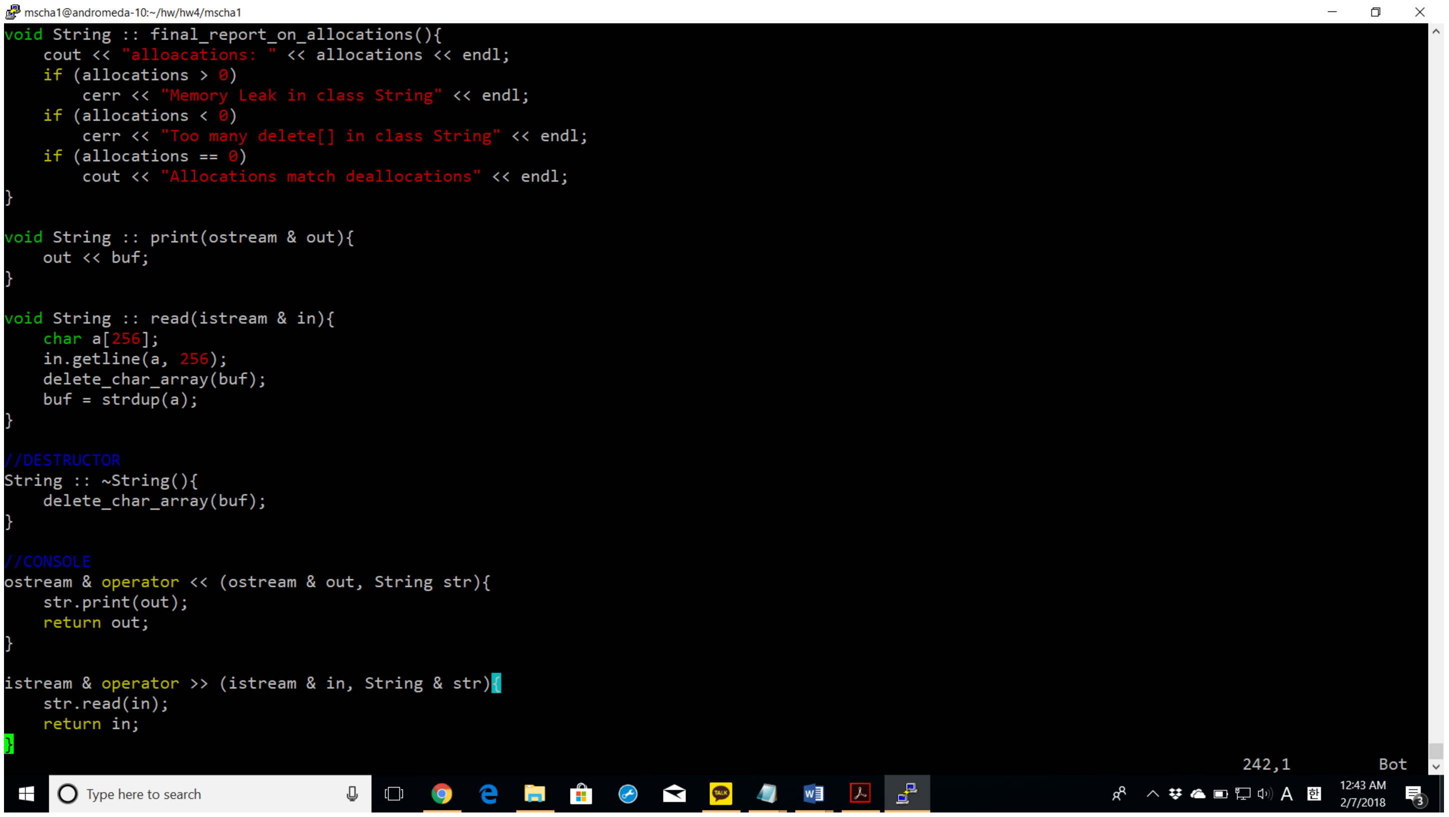

mscha1@andromeda-10:~/hw/hw4/mscha1

```
char & String :: operator [] (int index){
    if (inBounds(index))
        return buf[index];
    else
        cerr << "INDEX OUT OF BOUNDS" << endl;
        return buf[0];
}

int String :: size(){
    return strlen(buf);
}

int String :: indexOf(const char c){
    const char * found = strchr(buf, c);
    if (found == nullptr)
        return -1;
    else
        return found - buf;
}

int String :: indexOf(const String pattern){
    const char * found = strstr(buf, pattern.buf);
    if (found == nullptr){
        return -1;
    }
    else
        return found - buf;
}

bool String :: operator == (const String s){
    return strcmp(buf, s.buf) == 0;
}

bool String :: operator != (const String s){
    return strcmp(buf, s.buf) != 0;
}
                                                            168,1         64%
```


mscha1@andromeda-10:~/hw/hw4/mscha1

```

bool String :: operator > (const String s){
    return strcmp(buf, s.buf) > 0;
}

bool String :: operator < (const String s){
    return strcmp(buf, s.buf) < 0;
}

bool String :: operator <= (const String s){
    return strcmp(buf, s.buf) <= 0;
}

bool String :: operator >= (const String s){
    return strcmp(buf, s.buf) >= 0;
}

String String :: reverse(){
    int len = strlen(buf);
    char * temp = new_char_array(len+1);
    String rev(temp);
    reverse_cpy(rev.buf, buf);
    delete_char_array(temp);
    return rev;
}

String String :: operator + (const String s){
    char * temp = addArrays(buf, s.buf);
    String added(temp);
    delete_char_array(temp);
    return added;
}

String String :: operator += (const String s){
    *this = *this + s;
    return *this;
                                                            204,1         81%
```

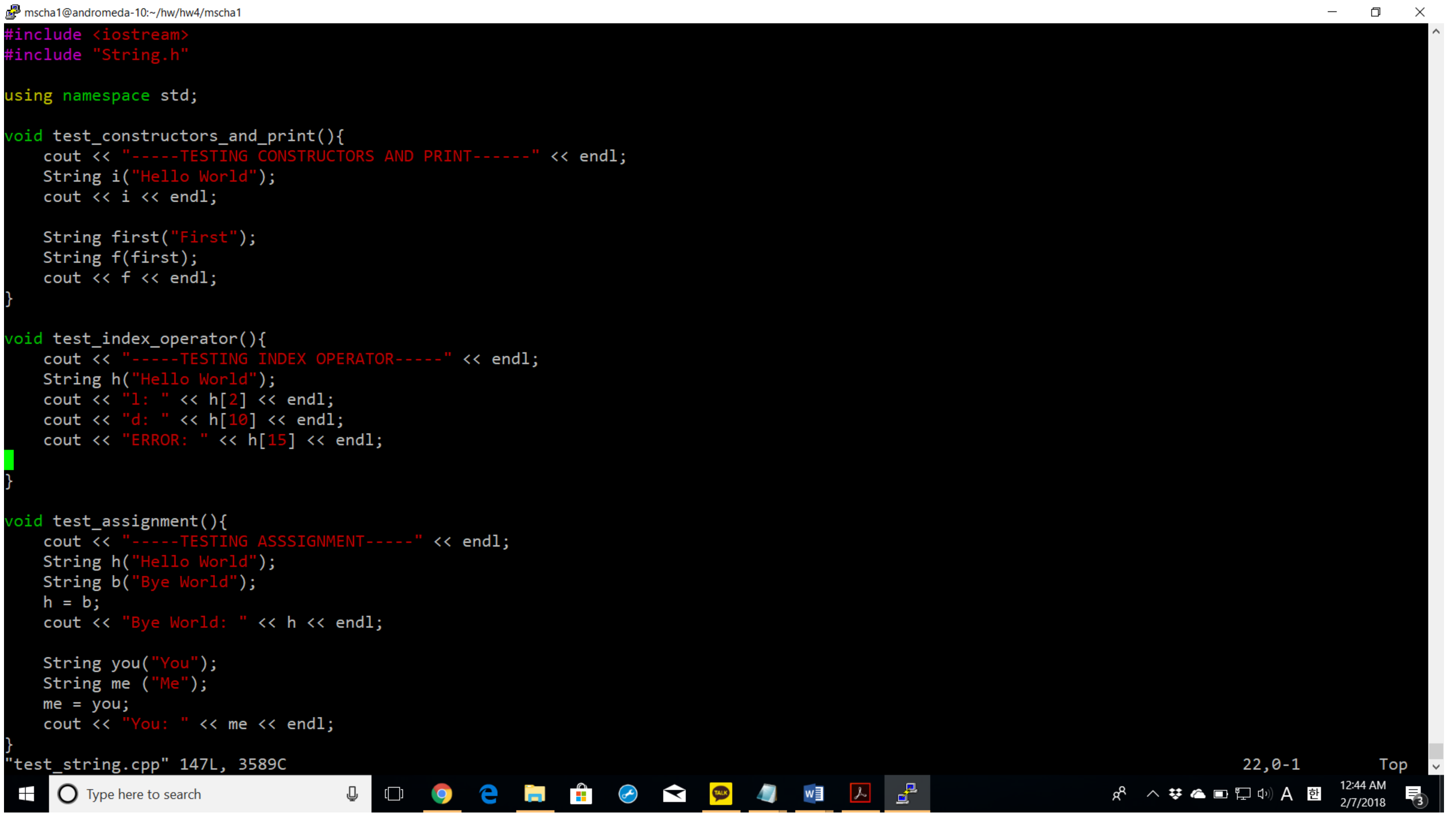
reverse(): creates a new String object passed with a temporary array of characters and the reverse of the current string is copied to the new String object.



```
void String :: final_report_on_allocations(){
    cout << "alloacations: " << allocations << endl;
    if (allocations > 0)
        cerr << "Memory Leak in class String" << endl;
    if (allocations < 0)
        cerr << "Too many delete[] in class String" << endl;
    if (allocations == 0)
        cout << "Allocations match deallocations" << endl;
}

void String :: print(ostream & out){
    out << buf;
}

void String :: read(istream & in){
    char a[256];
    in.getline(a, 256);
    delete_char_array(buf);
    buf = strdup(a);
}

//DESTRUCTOR
String :: ~String(){
    delete_char_array(buf);
}

//CONSOLE
ostream & operator << (ostream & out, String str){
    str.print(out);
    return out;
}

istream & operator >> (istream & in, String & str){
    str.read(in);
    return in;
}
                                                                242,1         Bot
```

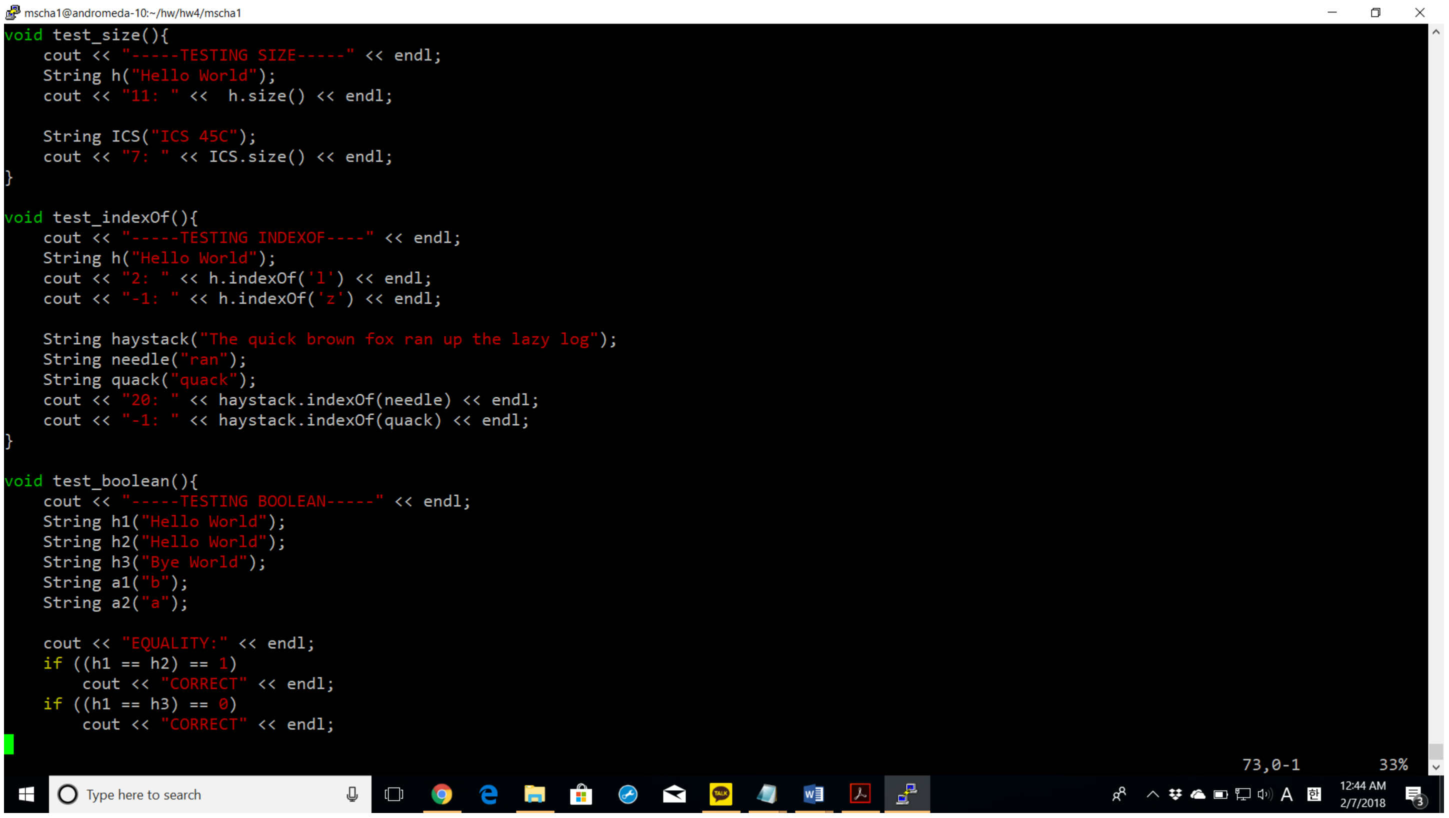
~String(): the buf of this String is deleted.

test_string.cpp



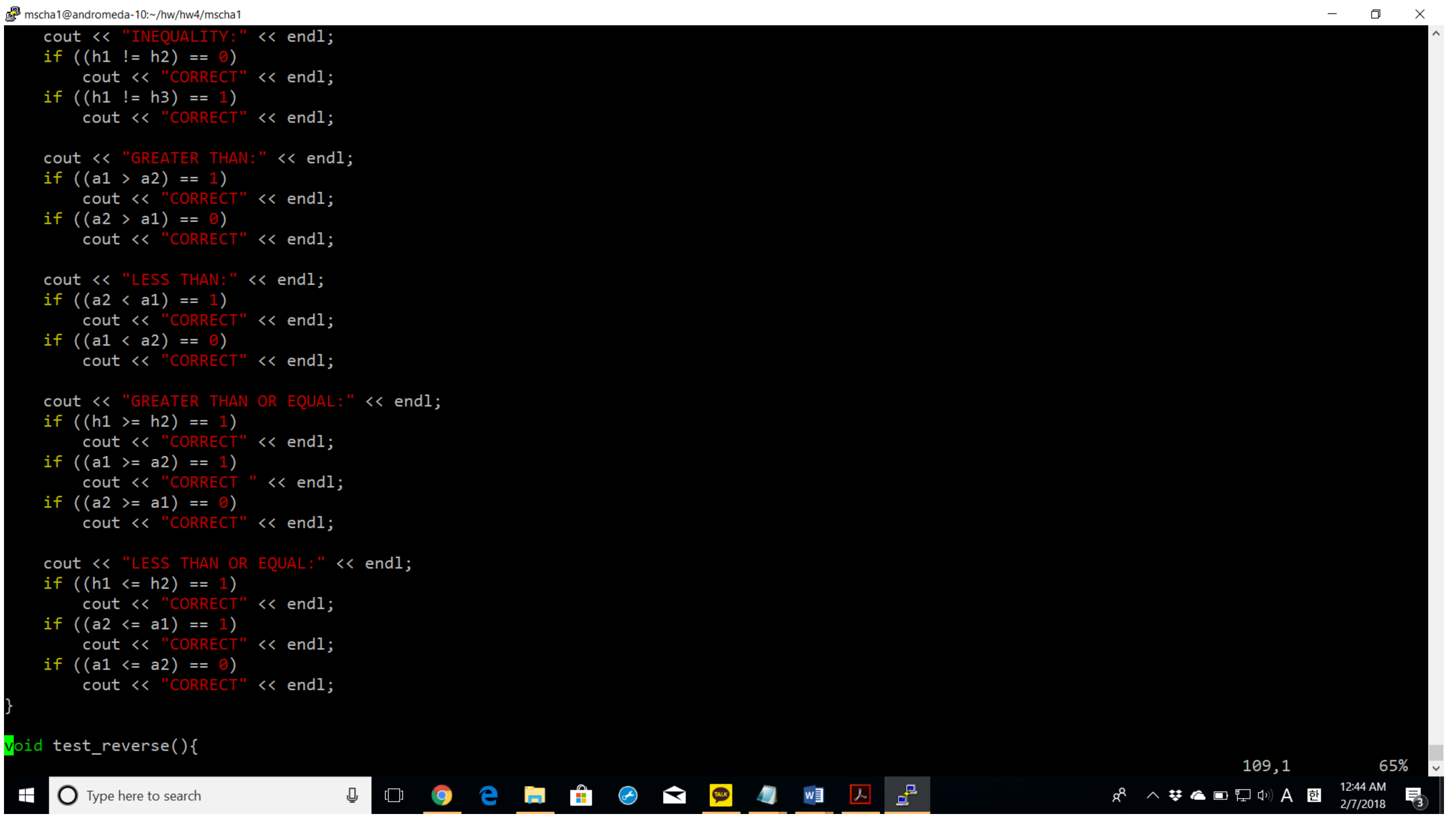
mscha1@andromeda-10:~/hw/hw4/mscha1

```
#include <iostream>
#include "String.h"

using namespace std;

void test_constructors_and_print(){
    cout << "-----TESTING CONSTRUCTORS AND PRINT------" << endl;
    String i("Hello World");
    cout << i << endl;

    String first("First");
    String f(first);
    cout << f << endl;
}

void test_index_operator(){
    cout << "-----TESTING INDEX OPERATOR-----" << endl;
    String h("Hello World");
    cout << "l: " << h[2] << endl;
    cout << "d: " << h[10] << endl;
    cout << "ERROR: " << h[15] << endl;

}

void test_assignment(){
    cout << "-----TESTING ASSIGNMENT-----" << endl;
    String h("Hello World");
    String b("Bye World");
    h = b;
    cout << "Bye World: " << h << endl;

    String you("You");
    String me ("Me");
    me = you;
    cout << "You: " << me << endl;
}
"test_string.cpp" 147L, 3589C                                                                              22,0-1        Top
```

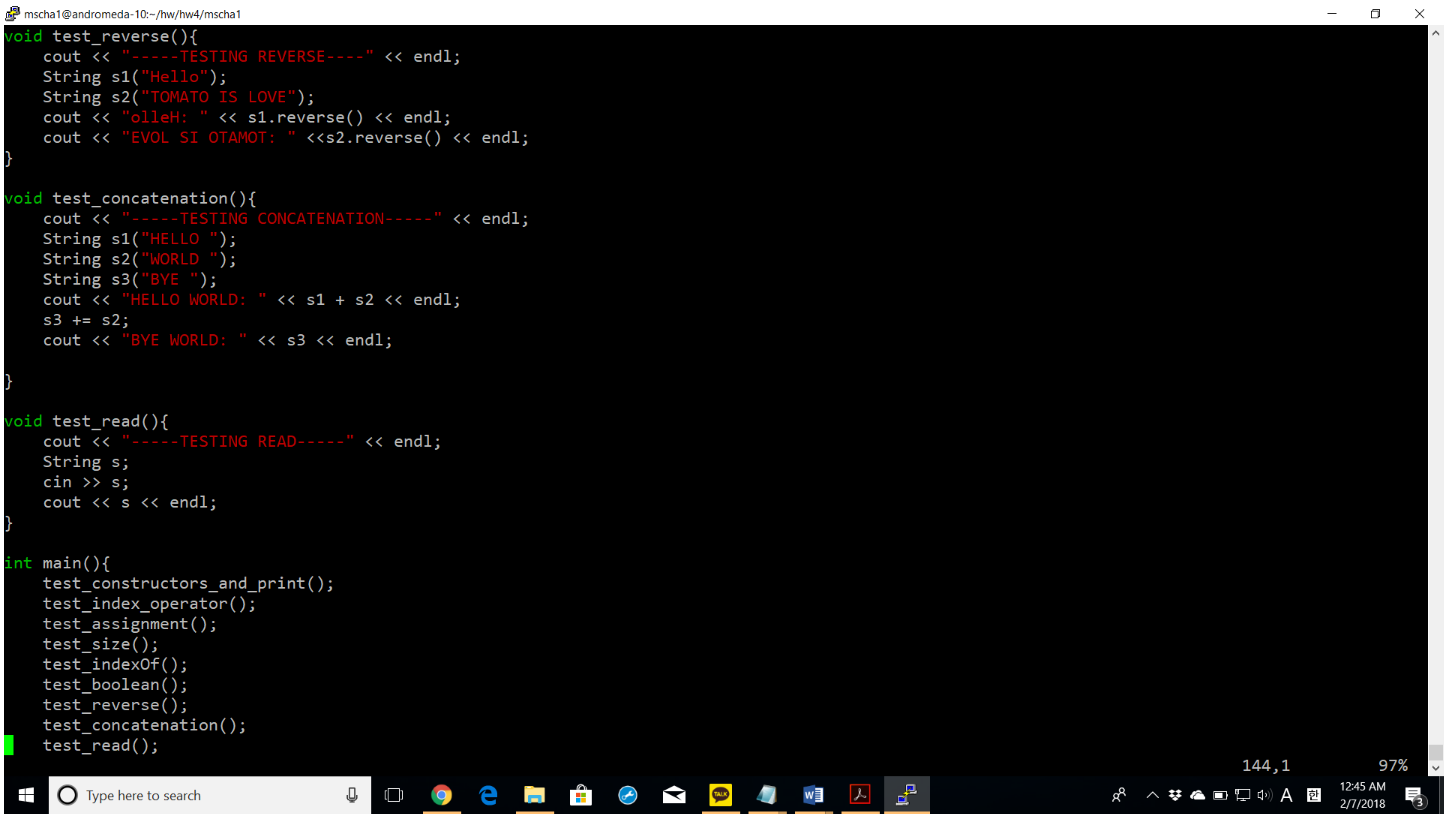


mscha1@andromeda-10:~/hw/hw4/mscha1

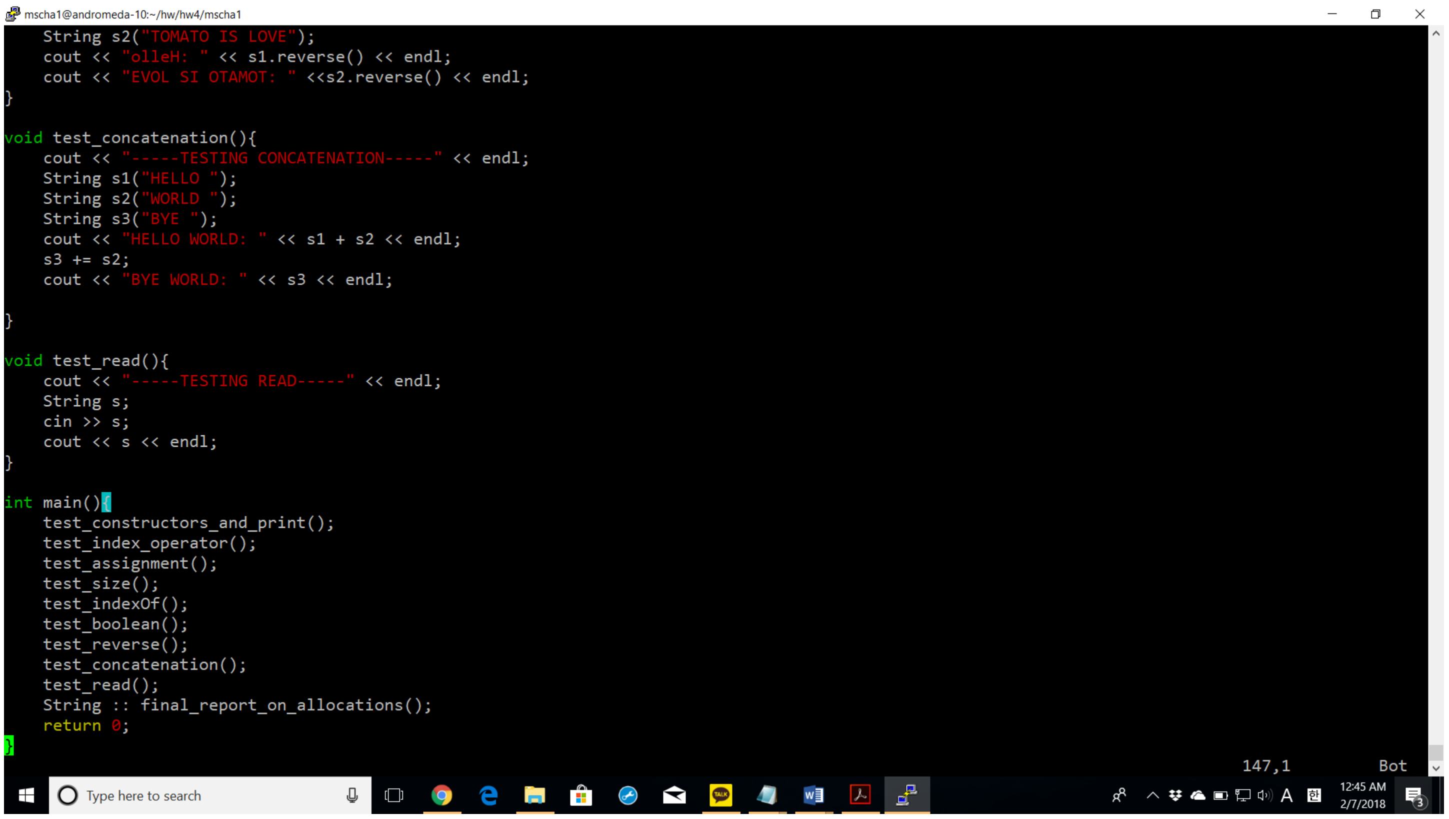
```
void test_size(){
    cout << "-----TESTING SIZE-----" << endl;
    String h("Hello World");
    cout << "11: " <<  h.size() << endl;

    String ICS("ICS 45C");
    cout << "7: " << ICS.size() << endl;
}

void test_indexOf(){
    cout << "-----TESTING INDEXOF----" << endl;
    String h("Hello World");
    cout << "2: " << h.indexOf('l') << endl;
    cout << "-1: " << h.indexOf('z') << endl;

    String haystack("The quick brown fox ran up the lazy log");
    String needle("ran");
    String quack("quack");
    cout << "20: " << haystack.indexOf(needle) << endl;
    cout << "-1: " << haystack.indexOf(quack) << endl;
}

void test_boolean(){
    cout << "-----TESTING BOOLEAN-----" << endl;
    String h1("Hello World");
    String h2("Hello World");
    String h3("Bye World");
    String a1("b");
    String a2("a");

    cout << "EQUALITY:" << endl;
    if ((h1 == h2) == 1)
        cout << "CORRECT" << endl;
    if ((h1 == h3) == 0)
        cout << "CORRECT" << endl;

                                                                                                            73,0-1        33%
```



```
    cout << "INEQUALITY:" << endl;
    if ((h1 != h2) == 0)
        cout << "CORRECT" << endl;
    if ((h1 != h3) == 1)
        cout << "CORRECT" << endl;

    cout << "GREATER THAN:" << endl;
    if ((a1 > a2) == 1)
        cout << "CORRECT" << endl;
    if ((a2 > a1) == 0)
        cout << "CORRECT" << endl;

    cout << "LESS THAN:" << endl;
    if ((a2 < a1) == 1)
        cout << "CORRECT" << endl;
    if ((a1 < a2) == 0)
        cout << "CORRECT" << endl;

    cout << "GREATER THAN OR EQUAL:" << endl;
    if ((h1 >= h2) == 1)
        cout << "CORRECT" << endl;
    if ((a1 >= a2) == 1)
        cout << "CORRECT " << endl;
    if ((a2 >= a1) == 0)
        cout << "CORRECT" << endl;

    cout << "LESS THAN OR EQUAL:" << endl;
    if ((h1 <= h2) == 1)
        cout << "CORRECT" << endl;
    if ((a2 <= a1) == 1)
        cout << "CORRECT" << endl;
    if ((a1 <= a2) == 0)
        cout << "CORRECT" << endl;
}

void test_reverse(){
                                                                    109,1         65%
```



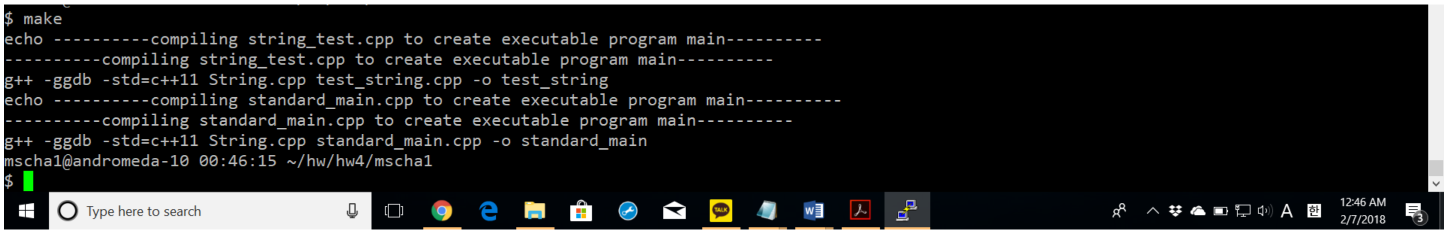
```
void test_reverse(){
    cout << "-----TESTING REVERSE----" << endl;
    String s1("Hello");
    String s2("TOMATO IS LOVE");
    cout << "olleH: " << s1.reverse() << endl;
    cout << "EVOL SI OTAMOT: " <<s2.reverse() << endl;
}

void test_concatenation(){
    cout << "-----TESTING CONCATENATION-----" << endl;
    String s1("HELLO ");
    String s2("WORLD ");
    String s3("BYE ");
    cout << "HELLO WORLD: " << s1 + s2 << endl;
    s3 += s2;
    cout << "BYE WORLD: " << s3 << endl;

}

void test_read(){
    cout << "-----TESTING READ-----" << endl;
    String s;
    cin >> s;
    cout << s << endl;
}

int main(){
    test_constructors_and_print();
    test_index_operator();
    test_assignment();
    test_size();
    test_indexOf();
    test_boolean();
    test_reverse();
    test_concatenation();
    test_read();
                                                                    144,1         97%
```



mscha1@andromeda-10:~/hw/hw4/mscha1

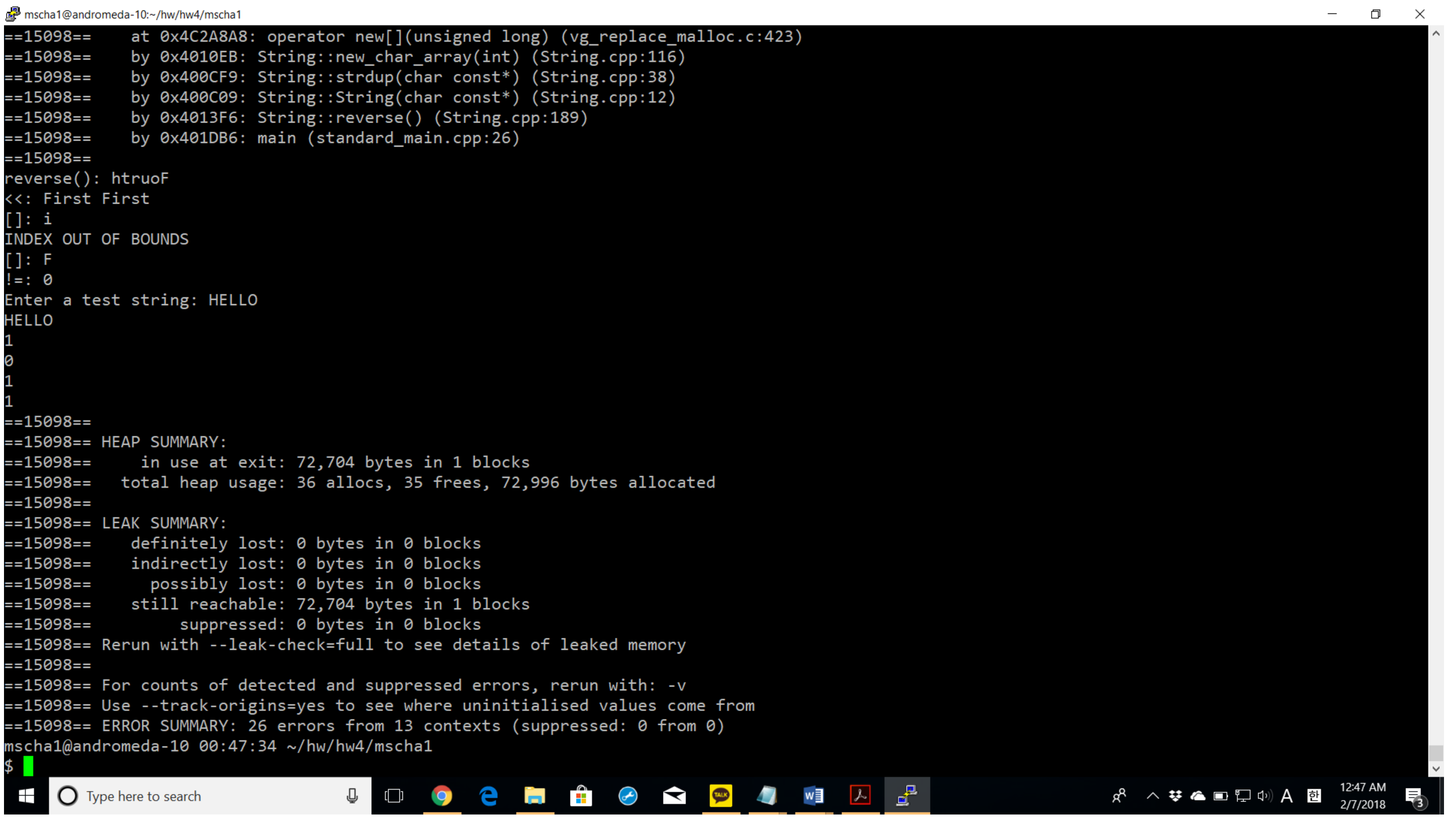
```
    String s2("TOMATO IS LOVE");
    cout << "olleH: " << s1.reverse() << endl;
    cout << "EVOL SI OTAMOT: " <<s2.reverse() << endl;
}

void test_concatenation(){
    cout << "-----TESTING CONCATENATION-----" << endl;
    String s1("HELLO ");
    String s2("WORLD ");
    String s3("BYE ");
    cout << "HELLO WORLD: " << s1 + s2 << endl;
    s3 += s2;
    cout << "BYE WORLD: " << s3 << endl;

}

void test_read(){
    cout << "-----TESTING READ-----" << endl;
    String s;
    cin >> s;
    cout << s << endl;
}

int main(){
    test_constructors_and_print();
    test_index_operator();
    test_assignment();
    test_size();
    test_indexOf();
    test_boolean();
    test_reverse();
    test_concatenation();
    test_read();
    String :: final_report_on_allocations();
    return 0;
}
                                                                        147,1         Bot
```

Screenshots of successful compiling with make command



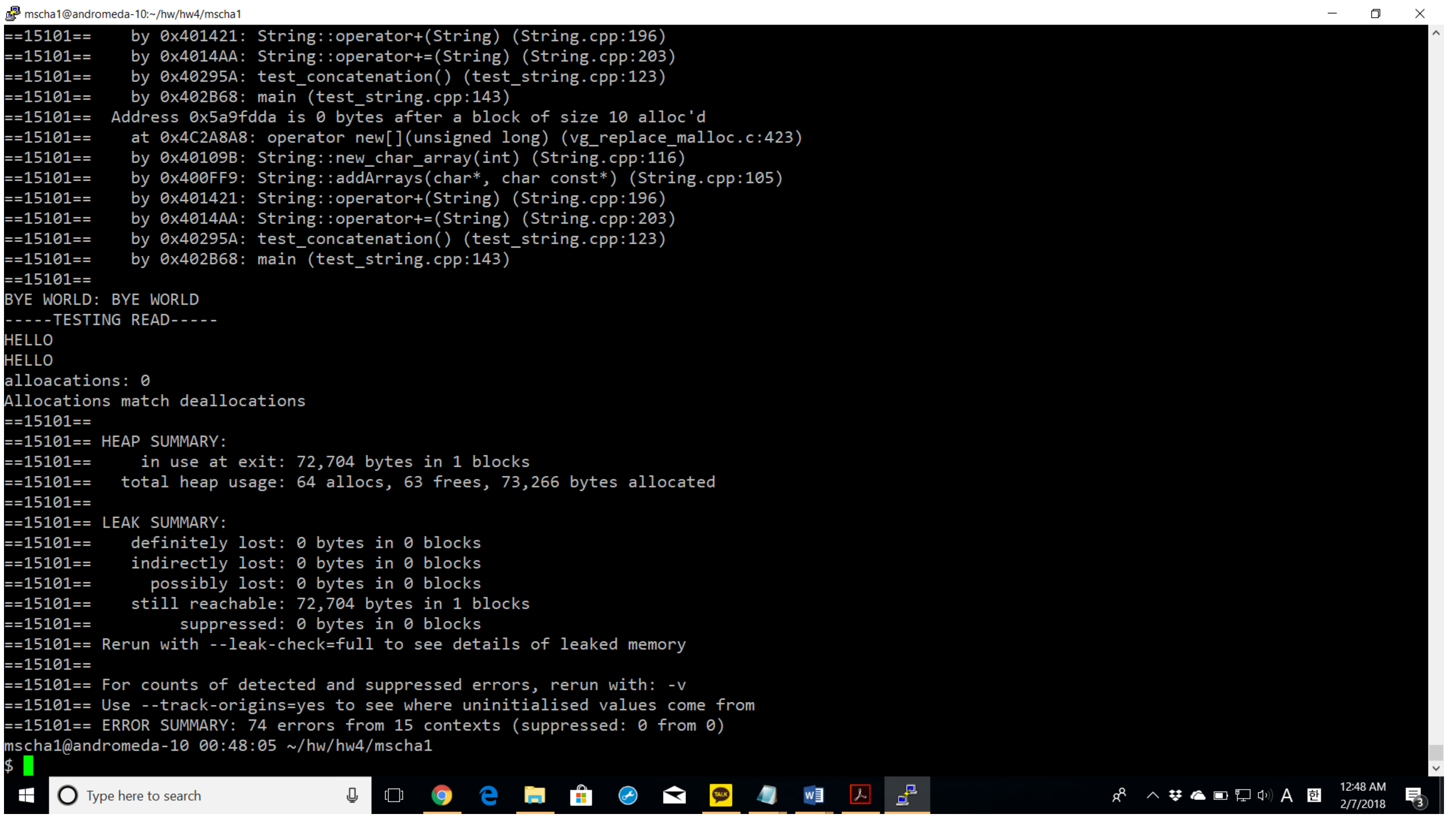
```
$ make
echo ----------compiling string_test.cpp to create executable program main----------
----------compiling string_test.cpp to create executable program main----------
g++ -ggdb -std=c++11 String.cpp test_string.cpp -o test_string
echo ----------compiling standard_main.cpp to create executable program main----------
----------compiling standard_main.cpp to create executable program main----------
g++ -ggdb -std=c++11 String.cpp standard_main.cpp -o standard_main
mscha1@andromeda-10 00:46:15 ~/hw/hw4/mscha1
$ 
```

Screenshots of successful execution with valgrind command


mscha1@andromeda-10:~/hw/hw4/mscha1

```
==15098==    at 0x4C2A8A8: operator new[](unsigned long) (vg_replace_malloc.c:423)
==15098==    by 0x4010EB: String::new_char_array(int) (String.cpp:116)
==15098==    by 0x400CF9: String::strdup(char const*) (String.cpp:38)
==15098==    by 0x400C09: String::String(char const*) (String.cpp:12)
==15098==    by 0x4013F6: String::reverse() (String.cpp:189)
==15098==    by 0x401DB6: main (standard_main.cpp:26)
==15098==
reverse(): htruoF
<<: First First
[]: i
INDEX OUT OF BOUNDS
[]: F
!=: 0
Enter a test string: HELLO
HELLO
1
0
1
1
==15098==
==15098== HEAP SUMMARY:
==15098==     in use at exit: 72,704 bytes in 1 blocks
==15098==   total heap usage: 36 allocs, 35 frees, 72,996 bytes allocated
==15098==
==15098== LEAK SUMMARY:
==15098==    definitely lost: 0 bytes in 0 blocks
==15098==    indirectly lost: 0 bytes in 0 blocks
==15098==      possibly lost: 0 bytes in 0 blocks
==15098==    still reachable: 72,704 bytes in 1 blocks
==15098==         suppressed: 0 bytes in 0 blocks
==15098== Rerun with --leak-check=full to see details of leaked memory
==15098==
==15098== For counts of detected and suppressed errors, rerun with: -v
==15098== Use --track-origins=yes to see where uninitialised values come from
==15098== ERROR SUMMARY: 26 errors from 13 contexts (suppressed: 0 from 0)
mscha1@andromeda-10 00:47:34 ~/hw/hw4/mscha1
$
```

Type here to search 12:47 AM 2/7/2018


mscha1@andromeda-10:~/hw/hw4/mscha1

```
==15101==    by 0x401421: String::operator+(String) (String.cpp:196)
==15101==    by 0x4014AA: String::operator+=(String) (String.cpp:203)
==15101==    by 0x40295A: test_concatenation() (test_string.cpp:123)
==15101==    by 0x402B68: main (test_string.cpp:143)
==15101==  Address 0x5a9fdda is 0 bytes after a block of size 10 alloc'd
==15101==    at 0x4C2A8A8: operator new[](unsigned long) (vg_replace_malloc.c:423)
==15101==    by 0x40109B: String::new_char_array(int) (String.cpp:116)
==15101==    by 0x400FF9: String::addArrays(char*, char const*) (String.cpp:105)
==15101==    by 0x401421: String::operator+(String) (String.cpp:196)
==15101==    by 0x4014AA: String::operator+=(String) (String.cpp:203)
==15101==    by 0x40295A: test_concatenation() (test_string.cpp:123)
==15101==    by 0x402B68: main (test_string.cpp:143)
==15101==
BYE WORLD: BYE WORLD
-----TESTING READ-----
HELLO
HELLO
alloacations: 0
Allocations match deallocations
==15101==
==15101== HEAP SUMMARY:
==15101==     in use at exit: 72,704 bytes in 1 blocks
==15101==   total heap usage: 64 allocs, 63 frees, 73,266 bytes allocated
==15101==
==15101== LEAK SUMMARY:
==15101==    definitely lost: 0 bytes in 0 blocks
==15101==    indirectly lost: 0 bytes in 0 blocks
==15101==      possibly lost: 0 bytes in 0 blocks
==15101==    still reachable: 72,704 bytes in 1 blocks
==15101==         suppressed: 0 bytes in 0 blocks
==15101== Rerun with --leak-check=full to see details of leaked memory
==15101==
==15101== For counts of detected and suppressed errors, rerun with: -v
==15101== Use --track-origins=yes to see where uninitialised values come from
==15101== ERROR SUMMARY: 74 errors from 15 contexts (suppressed: 0 from 0)
mscha1@andromeda-10 00:48:05 ~/hw/hw4/mscha1
$
```

Type here to search 12:48 AM 2/7/2018