**Take-Home Quiz 8 (15 pts) – Recursion & More Pointers**

**NOTE: Please submit your hard copy solution in lab this week**

1. (7 pts) Write a function called recursively\_reverse\_string() that accepts a *pointer* to a string as a parameter, and any other parameters you see fit, recursively *reverses* the string, and returns a *pointer* to the *reversed* string. For example, the reversed string of input “string” is “gnirts”.

//pass in left = 0 and right = strlen(source) - 1

//when calling this function

char \* recursively\_reverse\_string(char source[],

int left, int right) {

//we've moved past the halfway point of the string //and any reversal operation on the back half

//would simply put the elements back in the

//original location.

if(left >= right) {

return source;

}

else {

//swap the left and right elements

char temp = source[left];

source[left] = source[right];

source[right] = temp;

//do this shit again on the next pair

//going inwards

return recursively\_reverse\_string(source,

left + 1, right - 1);

}

}

1. (8 pts) Write a function recursive\_string\_copy() which accepts a *pointer* to a source string and a *pointer* to a destination string as parameters, recursively *copies* from the source to destination (including the null character), and returns nothing. You may not use any functions from <string.h>. Hint: each recursive step requires that you pass in the address of the next character to copy from the source and the address of the next destination character location.

//when calling this function externally,

//pass in 0 for index

void recursive\_string\_copy(char sorc[], char dest[], int index) {

//copy the current character from source to dest

dest[index] = sorc[index];

//if the character just copied is not the null //character, copy the next element

if(dest[index] != '\0') {

recursive\_string\_copy(sorc, dest, index + 1);

}

}