**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”

char\* recursively\_reverse\_string(char\* str, int start, int end)

{

char temp;

//Base case

if (start >= end)

return;

else

{

temp = \*(str + start);

\*(str + start) = \*(str + end);

\*(str + end) = temp;

//Recursion

reverse\_string(str, ++start, --end);

}

return str;

}

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.

void recursive\_string\_copy(char\* destination\_str, char\* source\_str)

{

\*destination\_str = \*source\_str;

//Base case

if (\*source\_str == '\0')

{

return;

}

else

{

recursive\_string\_copy(++destination\_str, ++source\_str);

}

}