*You may use basic library functions like strlen, strcmp, etc for all questions. If you don't remember the exact call just make sure it is clear what each parameter is and what you expect the function to be doing. For example, if you are looking for a substring, but can't remember the function strstr, you can just write substring, as long as it is clear what you expect it to be doing.*

1. (5 pts) Consider the following program:

```
#include <stdio.h>

int  func3(int * three) {
        three = 33333;
        return (int) three;
}

int func2(int * two) {
        (*two) = 22222;
        return func3(two);
}

int func1(int * one) {
        (*one) = 11111;
        return func2(one);
}

int main() {
        int zero = 0;
        int rval;
        rval = func1(&zero);
        printf("rval: %d", rval);
        printf("zero: %d", zero);
        return 0;
}
```

What will this program print? Explain specifically and in your own words why it will print what it does.

2. Consider a program that is built using our conventions of:
   - separating declaration and definition into their own .h and .c files,
   - using a makefile, and
   - putting the “driver” program that uses the functions and contains the main() function in a separate file.

Suppose the files are named as follows:

makefile
functions.h
functions.c
main.c

(a) [3 pts] Indicate on the associated blank line the file within which you would expect to find the following complete single lines of text. (In other words, in which file or files listed above would you expect to find the following lines?)

_____ gcc -c main.c –o main.o

_____ void * init(){

_____ #ifndef FUNCTIONS_H

_____ FirstStruct.o: FirstStruct.h FirstStruct.c

_____ int main(){

(b) [2 pts] Assuming the code contains no errors and compiles successfully, name three additional files that you would expect to appear after running make, and indicate what kind of file each one is.

3. (10 points) Write a function that :
    a. takes a C string as a parameter,
    b. allocates space for a new string of the same size on the heap,
    c. reverses the string,
    d. and returns a pointer to the new reversed string.

    Also, you may only iterate using pointer arithmetic **(no array syntax)**.

    **You will not get any points if you use array syntax, [ ], even if your code is correct**

4. (6 points) For the following questions, please state any assumptions you make.

**a) What is the problem with following code?**

```
int main(){
	int *p = malloc(4);
	p = NULL;
	free(p);
}
```

**b) Given a 32 bit integers, what will likely happen when you run the following code?**

```
int main(){
	int *p = malloc(4);
	p[6] = 2;
	printf("%d", p[6]);
}
```

**c) How many bytes will the above code lose on exit on a 64 bit system.**

```
char * s;
int x;
int main(){
	int * ptr = malloc(sizeof(int));
	ptr = malloc(sizeof(char));
	free(ptr);
	return 0;
}
```

5. (3 points) Convert the following signed 8 bit binary to decimal

   number: 0011 1100 _____

   number: 0000 1110 _____

   number: 1111 1111 _____

6. (3 points) Convert the following decimal to 8 bit signed binary

   number: -1 _____

   number: 32 _____

   number: 15 _____

8. Write a preprocessing macro to swap 3 integers [a->b->c->a] and a main() (driver) to show how you would use your swap macro.

9. A file (name.txt) containing a list of your classmates reads as below (ALL names follow immediately with a 'new line' character).

    Renu
    Lekhana
    Krishna
    Ashish

Write code to read in the file, then rewrite it with the names in reverse order. IN other words, the file should be as following when your program is complete.

    Ashish
    Krishna
    Lekhana
    Renu

State any assumptions you make about the file or the program.