Midterm 2 Study Guide

# Chapter 5 Loops

1. Be able to convert any type of loop to any other
   1. For to while
   2. While to for
   3. Do while to while
   4. While to do while
   5. For to to do while
   6. Do while to for
2. What is the difference between a for, while, and do while loop?
   1. When should you use each
3. Know for each type of loop how the following affect their behavior
   1. Break
   2. Continue
   3. Return
4. Be able to solve problems with loops including nested loops

# Chapter 6 Arrays

1. Be able to create a static array
   1. What code would you write to declare
      1. An array of 10 integers
      2. An array of 4 doubles
      3. An array of integers initialized to 4, 5, 10
      4. An array of 12 characters
      5. A string that can hold words 5 letters long
      6. A string initialized to the word hello
2. What is the difference between an array of characters and a C string?
3. Be able to use all of the string functions covered in class as well as write the C code to implement them
   1. strlen
   2. Strcpy
   3. Strncpy
   4. Strcat
   5. Strncat
   6. Strcmp
4. Be able to create a work with multidimensional arrays
   1. Declare a 5 X 4 array of ints
   2. Declare an array of 10 strings whose maximum length is 20 characters
   3. Declare an 3 X 6 X 9 X 12 array of doubles

# Chapter 7 & 8 Pointers

1. Know where space for different parts of your programs is stored and how long it lasts
   1. Code
   2. Global and static variables
   3. Local variables
   4. Arguments
   5. Dynamically allocated space (space created by malloc)
2. Be able to work with pointers in C to solve problems
3. If you want to be able to modify a variable of type X your function must accept at least a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Be able to use call by pointer to be able to “return” multiple values from a function
   1. A function foo that “returns” both an int and a double
   2. A function that “returns” an int, int\*, and int\*\*
5. Be able to allocate space with malloc
   1. Dynamically create an array of 7 integers and assign it to an integer pointer named ar
   2. Dynamically create space for a string that will contain 9 characters and assign it to a char\* named str
6. Be able to dynamically create multidimensional arrays
   1. Dynamically create an array of doubles that is 3 X 4
   2. Dynamically create an array of doubles that is 5 X 10 X 12
7. Be able to reallocate space in an array.
   1. Resize an array of ints called ar to contain 10 elements
   2. Resize an array of chars called ar to contain 20 elements
   3. Resize a 2 X 5 array of doubles into a 2 X 8 array
8. Be able to delete an array that has been dynamically allocated.
   1. Delete a single dimensional array of ints that contains 10 elements
   2. Delete a 2D array of doubles is 3 X 5
   3. Delete an array of 17 strings
   4. Delete a 3D array of ints that is 3 X 2 X 2

# Command Line Arguments

1. Be able to write programs that accept command line arguments
2. What is the first command line argument passed to your program?
3. If I called a program as follows: ./a.out hello goodbye why sigh what would the value of argc be?

# Debugging

1. Be able to do the following in gdb
   1. Print out the value of a variable
   2. Print out the elements of an array
   3. Set a breakpoint on a given line
   4. Run a program
   5. Go to the next breakpoint in the code
   6. Go to the next line of code and if it is a function go inside of it
   7. Go to the next line of code but if it is a function go over it

# Practice Programs

1. Write a function that returns true if a string 1 begins with string2.
2. Write a void function that will append an element to the end of a list.
3. Write a function that will delete a 2D array of ints and set the pointer to NULL
4. Write a function that will sort an array of strings by their length.