IC221 Lab: Memory Leaks Worksheet Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sping AY2022, 100 points total

**Task 1 (50 points)**

(5) Compile and execute memleak.c. Verify the output and review the program.

(10) Run valgrind on the memleak program. How many bytes does it say have been “definitely” lost?

|  |
| --- |
| 60 bytes in 2 blocks |

(5) On what line(s) of code does valgrind indicate a memory leak has occurred? \_\_19,34,50\_\_\_

(10) Identify and describe at least one memory leak in memleak.c.

|  |
| --- |
| Pointers are allocated space on the heap, but they are never freed.  19: new\_a = calloc(size\*2, sizeof(int)); |

(10) Fix the memory leak you identified and verify your fix with valgrind.

(10) Describe how you fixed the memory leak:

|  |
| --- |
| There were a lot of ways to do it, but I took a lazy route. I just freed the array passed into doubleup() after it was used because in the code it is getting rewritten. Then I made sure to free the array at the end of the program. |

**Task 2 (50 points)**

(5) \_\_\_\_\_ Compile and execute the memviolation.c program.

(10) Describe the output and exeuction of the program. Does it seem to be consistent?

|  |
| --- |
| “Hello World!”. It seems to be consistent |

(10) Run the program under valgrind. Identify the line of code that is causing the memory violation and its input:

|  |
| --- |
| Invalid read of size 1  Line 11: char \* str = malloc(sizeof(hello));  char hello[] = “Hello World!”; |

(15) Describe the programming bug:

|  |
| --- |
| Str should be 1 byte bigger to account for the null byte (and assigned the nullbyte at the end) |

(10) \_\_\_\_\_ Fix the memory violation and verify your fix with valgrind.

**Submission**

- Fixed memleak.c

- Fixed memviolation.c

- This completed worksheet