CS2263
Laboratory #2 Report

Student Name: __________
Student Number: __________
Course Number: CS2263

May 21, 2025

Exercise 1

1.1

Modify the `p1.c` program from the textbook, page 27, by including the printing of the memory addresses of the variables `a`, `b`, and `c` in each function `main`, `g1`, and `g2`. (Note that `g1` and `g2` also use `a`, `b`, and `c`). For example:

```
printf("a's address is %p\n", &a);
```

Submit:

- The modified source code
- The screenshot of the output from your program
- Are the values of the variables printed from your program the same as obtained by your colleagues? Why?
- Are the addresses printed from your program the same as obtained by your colleagues? Why?
- Are the addresses printed for the variables in the function `g1` bigger or smaller than the addresses printed from the function `g2`? Why?

Exercise 2

2.1

Run the modified `p1.c` program in the debugger. Set up the breakpoints as described in Section 2.7, page 28, in the textbook.

```
(gdb) b g1
(gdb) b g2
```

Submit:

- The debugger screenshot showing the backtrace after reaching the breakpoint `g2`. How many frames are shown in the trace?
- The debugger screenshot showing the backtrace after reaching the breakpoint `g1` (as described in Section 2.7). How many frames are shown?
- Use the debugger `info frame n` command to display the starting memory address of each frame (show the screenshots). Are the stack addresses of each frame related to the addresses of the variables `a`, `b`, and `c` printed by the functions `g1` and `g2`? Explain this by showing a memory map indicating the boundaries of each frame and the storage location used by each variable in your program (similar to the diagram on slide 13 in the slide set S5).

Exercise 3

3.1

Modify the program from Exercise 4, Lab 1, so that when generating the Fibonacci numbers it uses the `calcFib(n)` function which returns the nth number in the Fibonacci sequence, and then run this new version of the program (printing the first 10 values) in the debugger. Set the breakpoints to the function `calcFib()`.

Submit:

- One debugger screenshot showing the backtrace after reaching the breakpoint `calcFib`.
- The source code.

Submission Instructions

Type (collect the output, etc.) the answers to Questions 1.1, 2.1, and 3.1 into a word processor and store them in a single PDF file (include a preamble (title, section) with your name, student number, course number, etc.). Please number all the pages! Submit this **SINGLE** PDF file on D2L as the Lab 2 Report by the due date indicated in the D2L dropbox.

Reference: Program `p1.c` from the Textbook

```
#include <stdio.h>
#include <stdlib.h>

int g1(int a, int b)
{
   int c = (a + b) * b;
```

```
    printf("g1: %d %d %d \n", a, b, c);
    return c;
}

int g2(int a, int b)
{
    int c = g1(a + 3, b - 11);
    printf("g2: %d %d %d \n", a, b, c);
    return c - b;
}

int main(int argc, char **argv)
{
    int a = 5;
    int b = 17;
    int c = g2(a - 1, b * 2);
    printf("main: %d %d %d \n", a, b, c);
    return EXIT_SUCCESS;
}
```