Introduction

This tutorial is in conjunction with CPSC 1011 Lab 10's example1.c file, which is available as part of a .zip file on Canvas.

Tutorial Objectives

By successfully completing today's tutorial, you will be able to:

- Use the gdb utility to determine where run-time errors are occurring within a C program.
- Explain how to use the gdb commands to debug a program

Tutorial Steps

1. Open up `example1.c` in your favorite code editor.
2. Compile the program with `-g` flag.
3. Attempt to run the program the way you'd usually run a compiled program (e.g, `./a.out`)
4. You should see that the program experiences the dreaded `Segmentation fault`.
5. Now, try to run the program using the gdb command discussed in lab:
   `prompt % gdb: gdb -tui ./a.out`
6. Run the gdb command: `list main`
7. Run the gdb command: `break main`
8. Run the gdb command `run`
   This command starts running the program up to the break point at `main`)
9. Run the command `p data`
   The debugger stops at the line preceding the current line of code shown, so when you try to print data, at that point, the array called `data` has not been initialized yet; in other words, the line of code that is highlighted is the line that *will be executed next*).
10. Run the command `next`
11. Run the command `p data`
    Check that the values for `data` are listed.
12. Run the command `next`
13. Run the command `next`
14. Run the command `next`
15. Run the command `p i`
16. Run the command `next`
17. Run the command `p i`
18. Run the command `next`
19. Run the command `p i`
    (Do you see what's happening?)
20. Run the command `continue`
    The program continues until it gets to the next break, or, as in this case, the segmentation fault since no other break was set
21. You should see:
    Program received signal SIGSEGV, Segmentation fault.

0x000000000040052d in main () at example.c:11

and the line of code that is highlighted in the upper box is:

```
>|11                         sum += data[i];
```

22. Run the command `list`
    This will list 10 lines near where the seg fault occurred (which is already in the upper box)
23. Run the command `p i`
    This will show what value `i` has at this point (what is `i` on your screen and why?)
24. Run the command `quit`
    This command will quit the gdb debugger and give you back a regular prompt