Lab 4 – Debugging and GDB
CPSC 2311- Spring 2024

CLEMSON
School of COMPUTING

Introduction

The goal of the lab is to give you practice with debugging and GDB: The GNU Project Debugger. GDB is a powerful tool, understanding how to use it is a good skill to have. You may have learned about GDB in either CPSC 1010 or CPSC 1020 if not both.
This is an individual assignment. You are not allowed to work with anyone on this project.

Due:. Monday, February 19, 2024, 11:59 PM

Lab Instructions

You will first watch a couple videos that walks you through how to use GDB.
https://www.youtube.com/watch?v=uhIt8YqtmuQ&feature=youtu.be
https://www.youtube.com/watch?v=xQ0ONbt-qPs

Part 1: 15% of grade
I have provided you with input.txt, driver.c, functions.c, and functions.h files. **Before** examining these files compile them. You will find these files have 1 or more errors and 1 or more warnings. These are compile time errors and warnings. Warnings are as unacceptable as errors. Now clear all of the errors and warnings.

Part 2: 35% of grade
Once you have cleared the warnings and error, compile the code using the flag -g, run it with the input.txt file. You should get a segfault. This is a runtime error. I purposely did not make the error hard to find. You have probably already spotted the error causing the segfault. **Do not fix it yet.** Use GDB to find the error. Set breakpoints to places the error could possibly be. Rerun the program using GDB to pinpoint the segfault. You **must post several screenshots saved in a pdf** (see submission instructions) showing the use of GDB. After clearing the runtime errors complete part 3. Even if you figure out the error before using GDB, you **must** demonstrate the use of GDB to find the error or you will see a deduction of 35 points. Play around with GDB posting **several** screenshots of your use of GDB. Knowing how to use GDB will help you in future classes.

Part 3: 50% of grade
Now that you have cleared the compile time and runtime errors, you will add a couple functions to this code, as well as, add the necessary code to the driver to produce the appropriate output, shown toward the end of the document. The functions are described below:

int calculateVal(int mat, int size);** - This function will return a sum of all values of the matrix **except** those that are located on the left and right diagonals. As an example, consider the following 5 X 5 matrix:

	0	1	2	3	4
0	2	2	2	2	2
1	2	2	2	2	2
2	2	2	2	2	2
3	2	2	2	2	2
4	2	2	2	2	2

The values in red, highlighted in blue are the left diagonal and the values in red highlighted in yellow are the right diagonal. The value 2 in the middle, highlighted in purple, is in both categories. This function will calculate the values of all the remaining numbers in the matrix. (32 for this matrix).
The function **calculateVal** will call functions **isRightDiagonal** and **isLeftDiagonal**.

bool isRightDiagonal(int size, int row, int col); - This function returns true if a given element in the defined 2D array is part of the right diagonal.

bool isLeftDiagonal(int row, int col); - Like the function above, this function returns true if a given element in the defined 2D array is part of the left diagonal.

HINT: Use the matrix above to help you determine the algorithm for the isRightDiagonal and isLeftDiagonal funtions. I strongly suggest you list out the matrix element locations that make up both the left and right diagonals. You should see a pattern forming and can deduce the algorithm for these functions.

Example: mat[0][0]
mat[1][1]
etc.

Here is a sample output for the input.txt file given to you.
The format of your output must be like the following:

01 01 01 01 01
01 01 01 01 01
01 01 01 01 01
01 01 01 01 01
01 01 01 01 01
Total = 16

Notice the output is 2 digits. Your file must output two digits as well.
You should test your program using several test files. Do not assume the matrix element all have the same value as shown in the example above. We will test your code with different values in each element.

FORMATTING:

Below is a sample Header.
You will need to add a similar header to all of your program files:

```
/************************
 *Your name
 *CPSC 2311 your Section
 *Your email
 ************************/
```

Your program should compile with no warnings and no errors. Points will be deducted for errors and warnings.

- Your code should be well documented. (comments)
- There should be no lines of code longer than 80 characters.
- You should use proper and consistent indention.

Here are some guidelines for documenting the code in your assignment.
Your function prototype should be in functions.h. Directly before the prototype you are **required** to have a detailed description of what the overall function does. You should explain what each parameter is and what it is used for.
Below is an example of the type of documentation I require.

```
/* Parameters: img - image_t pointer array holding the image data for
 *                   each of the input files
 * Return:       output - image_t struct containing output image data
 * This function averages every pixels rbg values from each of the
 * input images and puts those averages into a single output image
 */
```

Also, if you include comments in the body of the function (and you should) my preference is they should be placed above the line of code not beside the code. While it is perfectly fine to place comments beside code, I find it clutters your code. Therefore, for all assignments in my class you will not place comments beside code but above it.

Example:

```
Good
//This is a comment
if(something)
{
    do something;
}
```

```
        Bad
if(something) //This is a comment
{
    do something;
}
```

Submission Instructions

You will save your screenshots pdf as well as all code files (driver.c, functions.c, functions.h) into a **zip file named lastname.zip** (where lastname is your last name) on canvas under the lab 4 assignment module. **Failure to follow these instructions will incur a 10-point penalty.** Submitting a makefile is unnecessary but creating one is highly recommended, and you do not need to submit input.txt.