Resources: Getting Started with GDB

A debugger is a program that allows you to monitor your program as it runs, inspecting the flow of the program and the values of variables.

To use `gdb`, the GNU DeBugger on a C++ program, you must recompile the program, telling the compiler to leave debugger information in the program. This is accomplished by added the `-g` option to all `g++` commands. If you are using a `Makefile`, edit the file to accomplish this task. Remove any existing object files `rm *.o`. Then build again `make`.

To run the debugger on your program (I'll assume your program is called `ppm_menu` for now.), launch like this:

```
gdb ./ppm_menu
```

You will now see the `gdb)` prompt. GDB has opened your program, but is not running it yet. Let's assume you want to watch your program's flow of execution in a couple of functions named `imageMenu` and `takeAction`. You can tell the debugger you want it to pause execution anytime either of these functions are called by setting break points. Like this:

```
break imageMenu
break takeAction
```

Now you can start running the program, in the debugger by using the `run` command:

```
run
```

The program will execute until one of the break points is reached, or the program terminates. Let's say that the program enters the `imageMenu` function and so GDB pauses execution. You can now step through the statements in this function one at a time using the `next` command.

```
next
next
...
next
```

Every time you say `next`, gdb will run the next line of code. Along the way, you might want to examine the value of a variable. You can see variable values using the `print` command. For example, if you wanted to see the value of a variable named `output_image`, you could say:

```
print output_image
```

If you are done stepping through the program one line at a time and want to continue running the program at full speed, you can use the `continue` command. The program will continue until another break point is reached, or the program terminates.

If you just want to stop the debugger, use the `quit` command.

The debugger has many more features, but that should get you started.