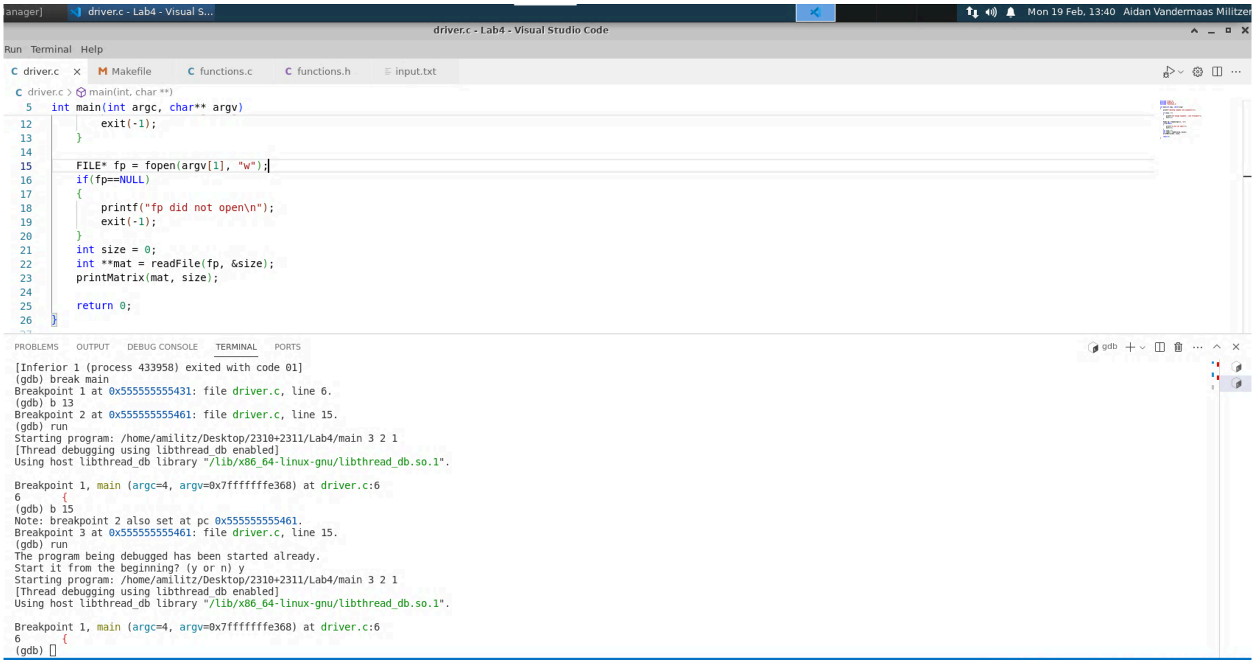
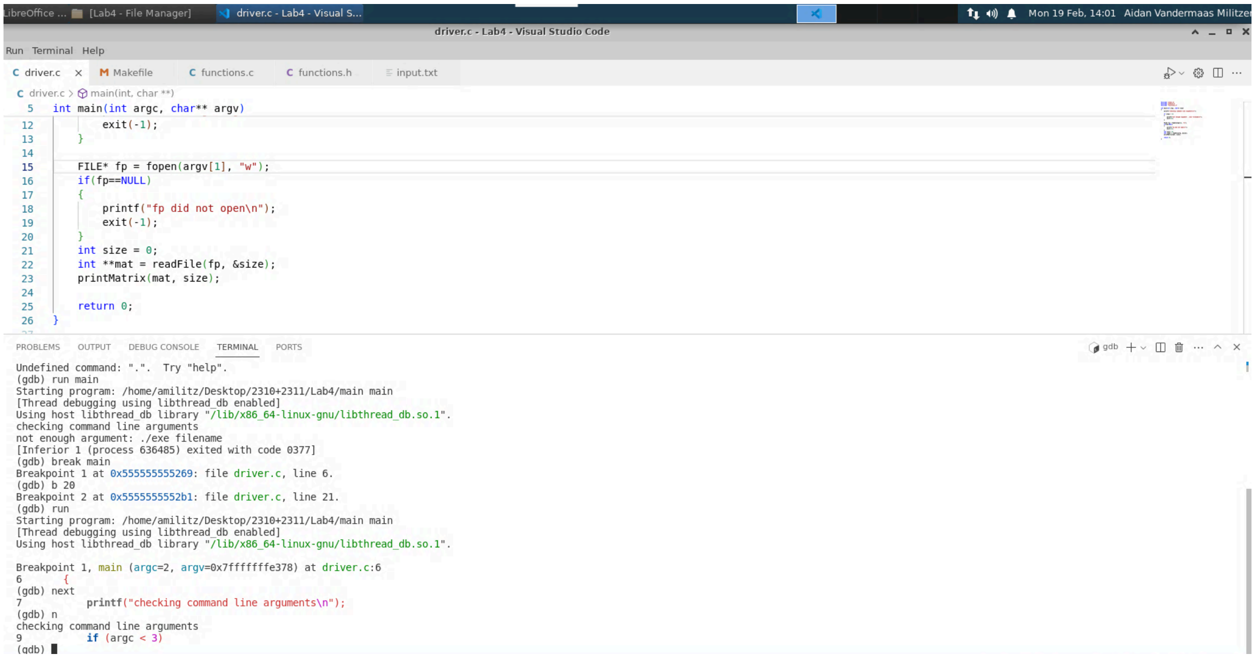


anager]
driver.c - Lab4 - Visual S...
Mon 19 Feb, 13:40 Aidan Vandermaas Militzer
driver.c - Lab4 - Visual Studio Code
Run Terminal Help
driver.c
Makefile
functions.c
functions.h
input.txt
driver.c > main(int, char **)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
gdb

```
5   int main(int argc, char** argv)
12          exit(-1);
13      }
14
15      FILE* fp = fopen(argv[1], "w");
16      if(fp==NULL)
17      {
18          printf("fp did not open\n");
19          exit(-1);
20      }
21      int size = 0;
22      int **mat = readFile(fp, &size);
23      printMatrix(mat, size);
24
25      return 0;
26  }
```

```
[Inferior 1 (process 433958) exited with code 01]
(gdb) break main
Breakpoint 1 at 0x555555555431: file driver.c, line 6.
(gdb) b 13
Breakpoint 2 at 0x555555555461: file driver.c, line 15.
(gdb) run
Starting program: /home/amilitz/Desktop/2310+2311/Lab4/main 3 2 1
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=4, argv=0x7fffffffe368) at driver.c:6
6	{
(gdb) b 15
Note: breakpoint 2 also set at pc 0x555555555461.
Breakpoint 3 at 0x555555555461: file driver.c, line 15.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/amilitz/Desktop/2310+2311/Lab4/main 3 2 1
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=4, argv=0x7fffffffe368) at driver.c:6
6	{
(gdb) 
```

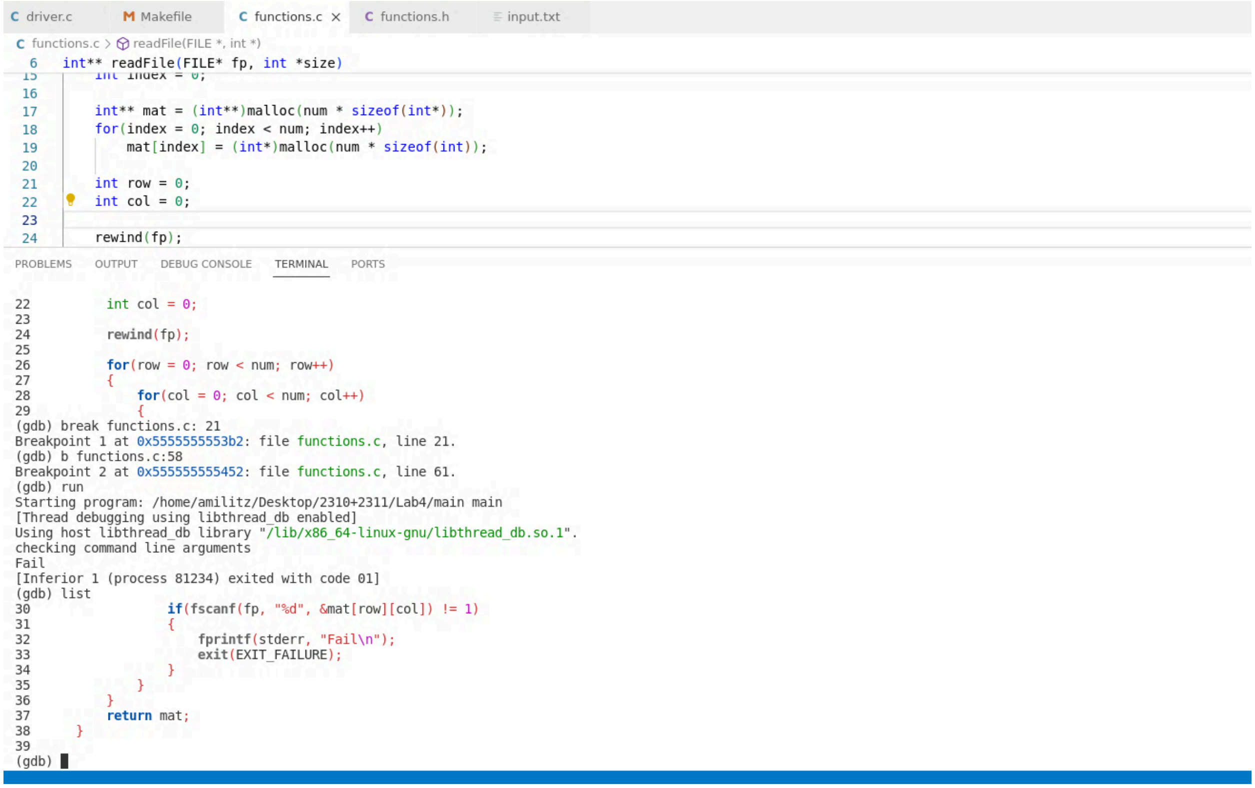
Using gdb, I was able to set breakpoints with the command break main to automatically set a breakpoint at line 6, then I wrote the command to set a second breakpoint at line 10, written as b 10. Typing the command run after setting the breakpoints will run the code between the breakpoints.



```c
int main(int argc, char** argv)
        exit(-1);
    }

    FILE* fp = fopen(argv[1], "w");
    if(fp==NULL)
    {
        printf("fp did not open\n");
        exit(-1);
    }
    int size = 0;
    int **mat = readFile(fp, &size);
    printMatrix(mat, size);

    return 0;
}
```

```
Undefined command: ".".  Try "help".
(gdb) run main
Starting program: /home/amilitz/Desktop/2310+2311/Lab4/main main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
checking command line arguments
not enough argument: ./exe filename
[Inferior 1 (process 636485) exited with code 0377]
(gdb) break main
Breakpoint 1 at 0x555555555269: file driver.c, line 6.
(gdb) b 20
Breakpoint 2 at 0x5555555552b1: file driver.c, line 21.
(gdb) run
Starting program: /home/amilitz/Desktop/2310+2311/Lab4/main main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=2, argv=0x7fffffffe378) at driver.c:6
6       {
(gdb) next
7           printf("checking command line arguments\n");
(gdb) n
checking command line arguments
9           if (argc < 3)
(gdb) 
```

Here you can see that using the next and n commands, I am able to look at each line of code that is being executed. At the bottom I can see that the line of code if(argc < 3) is being ran, however, when I set the breakpoints and ran the code the value of argc is equal to 2 which is preventing the code from running.



```
6   int** readFile(FILE* fp, int *size)
15      int index = 0;
16
17      int** mat = (int**)malloc(num * sizeof(int*));
18      for(index = 0; index < num; index++)
19          mat[index] = (int*)malloc(num * sizeof(int));
20
21      int row = 0;
22      int col = 0;
23
24      rewind(fp);
```

```
22          int col = 0;
23
24          rewind(fp);
25
26          for(row = 0; row < num; row++)
27          {
28              for(col = 0; col < num; col++)
29              {
(gdb) break functions.c: 21
Breakpoint 1 at 0x5555555553b2: file functions.c, line 21.
(gdb) b functions.c:58
Breakpoint 2 at 0x555555555452: file functions.c, line 61.
(gdb) run
Starting program: /home/amilitz/Desktop/2310+2311/Lab4/main main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
checking command line arguments
Fail
[Inferior 1 (process 81234) exited with code 01]
(gdb) list
30                  if(fscanf(fp, "%d", &mat[row][col]) != 1)
31                  {
32                      fprintf(stderr, "Fail\n");
33                      exit(EXIT_FAILURE);
34                  }
35              }
36          }
37          return mat;
38      }
39
(gdb) 
```

Here I used break functions.c: 21 to set the first point of the run and b functions.c:58 to end the run as the second breakpoint. The program runs the segment. If an error occurs, you can use list to list the code in the segments as well.