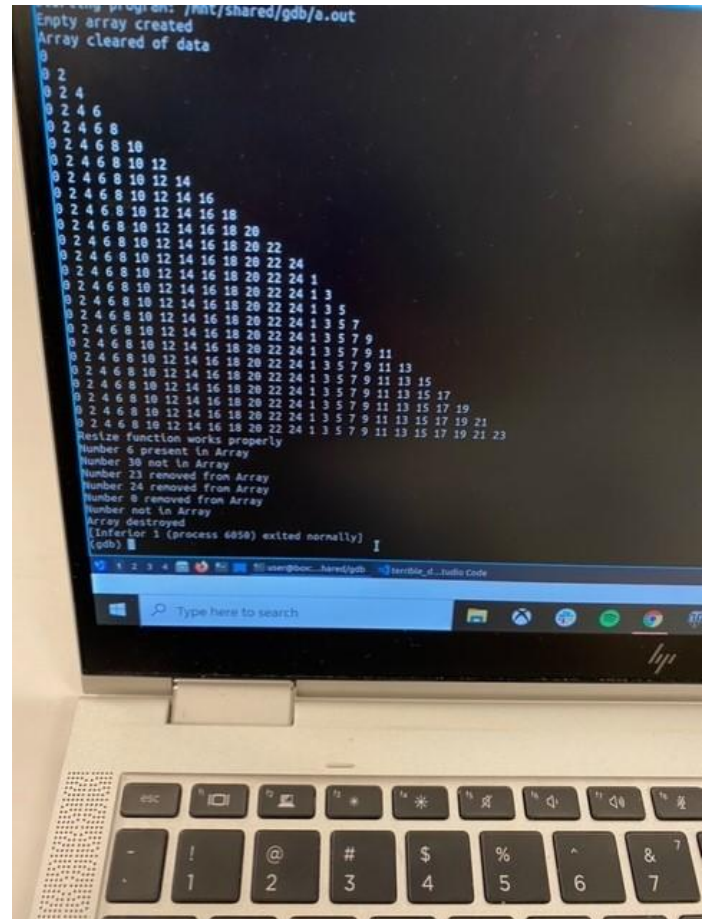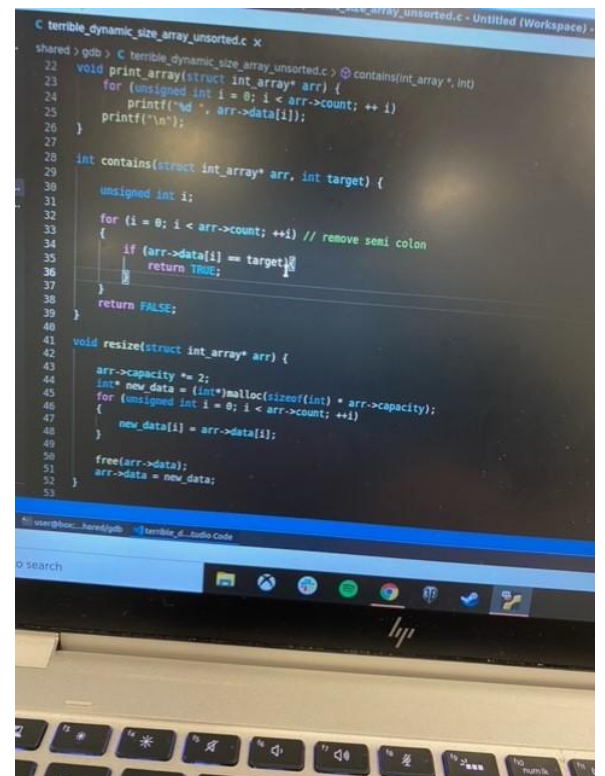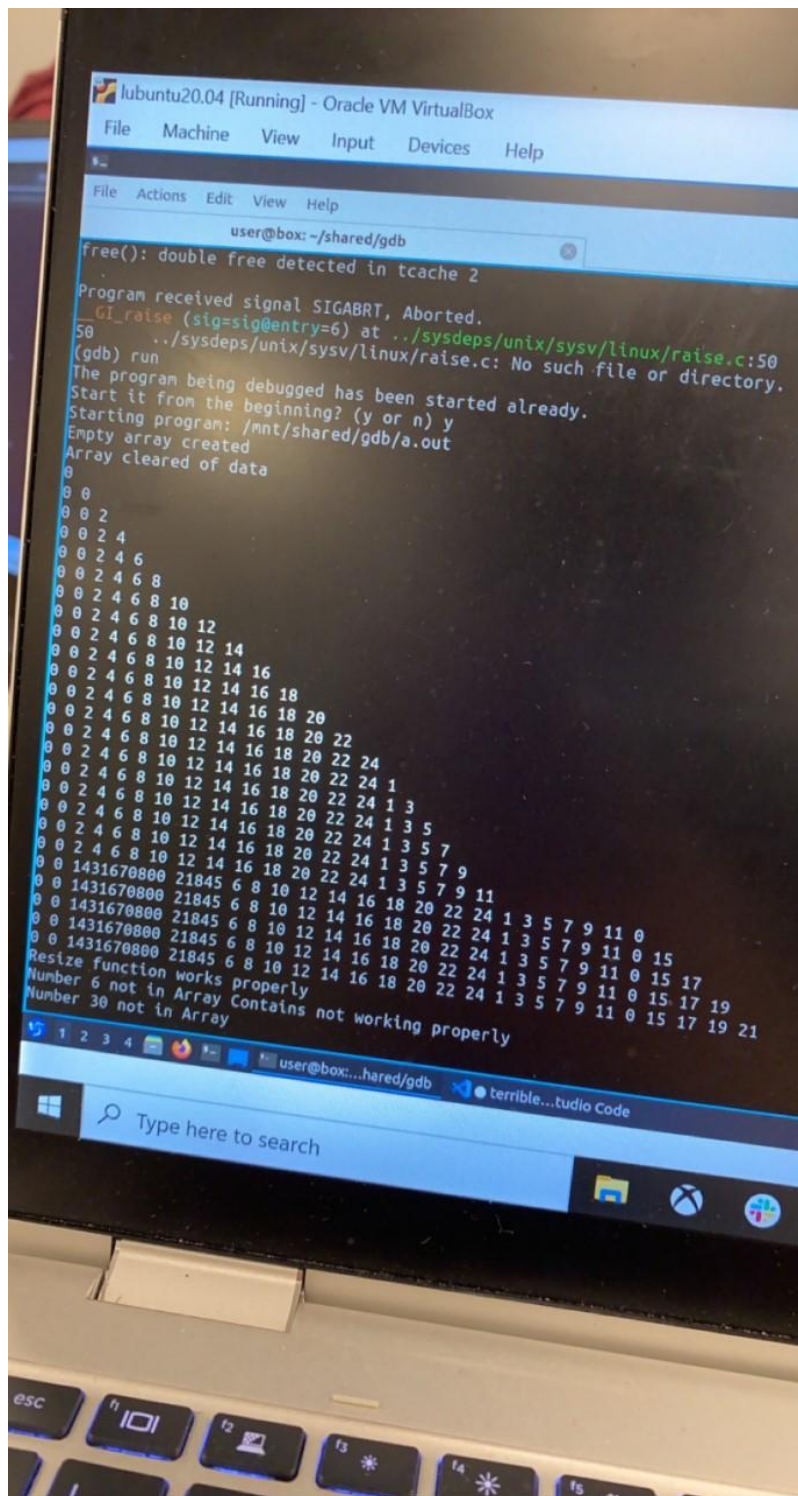Nicholas Cali

I pledge my honor that I have abided by the Stevens Honors System.

This shows that all errors were fixed using gdb. I will go into more detail later on in this document on where the errors were.

```
lubuntu20.04 [Running] - Oracle VM VirtualBox
File   Machine   View   Input   Devices   Help

File  Actions  Edit  View  Help
                user@box: ~/shared/gdb
free(): double free detected in tcache 2

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50        ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /mnt/shared/gdb/a.out
Empty array created
Array cleared of data
0 0
0 0 2
0 0 2 4
0 0 2 4 6
0 0 2 4 6 8
0 0 2 4 6 8 10
0 0 2 4 6 8 10 12
0 0 2 4 6 8 10 12 14
0 0 2 4 6 8 10 12 14 16
0 0 2 4 6 8 10 12 14 16 18
0 0 2 4 6 8 10 12 14 16 18 20
0 0 2 4 6 8 10 12 14 16 18 20 22
0 0 2 4 6 8 10 12 14 16 18 20 22 24
0 0 2 4 6 8 10 12 14 16 18 20 22 24 1
0 0 2 4 6 8 10 12 14 16 18 20 22 24 1 3
0 0 2 4 6 8 10 12 14 16 18 20 22 24 1 3 5
0 0 2 4 6 8 10 12 14 16 18 20 22 24 1 3 5 7
0 0 2 4 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11 0
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11 0 15
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11 0 15 17
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11 0 15 17 19
0 0 1431670800 21845 6 8 10 12 14 16 18 20 22 24 1 3 5 7 9 11 0 15 17 19 21
Resize function works properly
Number 6 not in Array Contains not working properly
Number 30 not in Array

Type here to search
```



```c
C terrible_dynamic_size_array_unsorted.c ×
shared > gdb > C terrible_dynamic_size_array_unsorted.c >  contains(int_array *, int)
22   void print_array(struct int_array* arr) {
23       for (unsigned int i = 0; i < arr->count; ++ i)
24           printf("%d ", arr->data[i]);
25       printf("\n");
26   }
27
28   int contains(struct int_array* arr, int target) {
29
30       unsigned int i;
31
32       for (i = 0; i < arr->count; ++i)
33       {                                      // remove semi colon
34
35           if (arr->data[i] == target)
36               return TRUE;
37
38       }
39       return FALSE;
40   }
41
42   void resize(struct int_array* arr) {
43
44       arr->capacity *= 2;
45       int* new_data = (int*)malloc(sizeof(int) * arr->capacity);
46       for (unsigned int i = 0; i < arr->count; ++i)
47       {
48           new_data[i] = arr->data[i];
49       }
50       free(arr->data);
51       arr->data = new_data;
52   }
53
```

The images above and to the right show a few of the errors. The first error we see is that there is an extra column of 0's in the triangle of values. This can be corrected in the method below.

void add(struct int_array* arr, int payload) {

    if ((arr->count == arr->capacity))

        resize(arr);

    //add - 1 to fix leading column of 0's
    arr->data[(++ arr->count)- 1] = payload;

}

The arr->data[(++arr->count) - 1] = payload line is the issue. Using breakpoints and common knowledge the previous line was arr->data[++arr->count] = payload. This was adding the extra column of 0's. The '-1' moves the shifts to the right to allow for the correct format. The other issue in the images above is in the larger numbers showing. In resize, deallocating before changing the variable will allow for the bigger number to fix itself. This was also fixed with gdb breakpoints. The other issues were the extra else statement which returned false and the semi-colon after the for loop. Both a syntax error and unnecessary code hinders everything. The final two errors which were displayed in the terminal and discovered by breakpoints were in remove_elem. One is a semicolon after an if statement and the other being the incorrect usage of '='. The correct usage should've been '==' and getting rid of the semi-colon. In total with the 6 errors fixed, the terminal then prints out the first image which is fully correct.