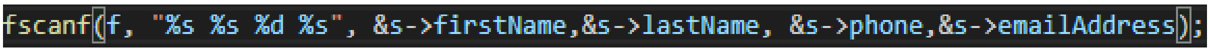
Issues found:

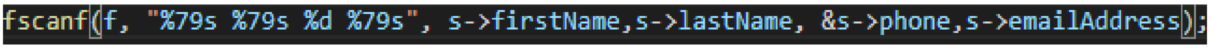
- Makefile is including maths library, but the library is not used in the program. This is bad as it means the resulting binary will be much bigger than it needs to be, as it includes the maths library. I removed the compiler flags and the include statement in main.c
- Makefile is also redirecting stderr to /dev/null, which means that no error messages will be displayed. To change this, I removed the output redirection in the make file
- Many spelling mistakes, “emialAddress” -> emailAddress
- Poorly named methods such as ffn, which is short for find first name. I have changed method names to be more readable
- Inconsistent * placement, not really a bug, just makes reading more frustrating
- In FindPhoneNumber, the if statement used to read: if(ss[i]->phone == ss), this is wrong as it is comparing an integer with a pointer, I changed it to be comparing the phone number of the struct and the integer passed in.
- 

  ```
  fscanf(f, "%s %s %d %s", &s->firstName,&s->lastName, &s->phone,&s->emailAddress);
  ```

  For this line, I removed the & for firstName, lastName and emailAddress. This is because they were being evaluated as arrays of char arrays, which was throwing an error.

  I also limited the number of characters we will be reading in to one less than the size of each char array (one byte left for the null terminating byte), I did this to avoid any overflows, where fscanf reads more data than there is space assigned to store it.

  This is the line now:

  ```
  fscanf(f, "%79s %79s %d %79s", s->firstName,s->lastName, &s->phone,s->emailAddress);
  ```

- There was no error handling when opening the file, it would just segfault… I added a null check before reading the file to make sure it existed.
- No error checking on number of arguments, this is bad because it will try access an out of bounds part of memory if there are no arguments passed to it.
- There was an issue in sortEmail, where the swap wasn’t being performed correctly and all three variables would just end up the same. To fix this, I tweaked the swap, so it was performing it’s job proper. There was a similar issue in sortLastName, and sortFirstName
- String comparison was done incorrectly, mainly in the find* methods, instead of ss[i]->emailAddress == s, it should be strcmp(ss[i]->emailAddress,s)
- There is no freeing going on…
- If the phone number is larger than 2147483647, the integer will overflow. To fix this, I changed the datatype of phone numbers to long.
- In the find* functions, if uses the condition while(++i < count), this means that we will never get to the first index, because it is adding one to I and then evaluating the expression. I fixed this by moving the i++ expression to the bottom of the while loop, so it runs the loop first and then increments i.
- In the find* functions again, we keep getting 0 for count, I solved this by passing in the count variable to the function when calling it.
- I changed the gets calls to fgets, this is because gets is very vulnerable to a buffer overflow.
- There were no checks to ensure fscanf was actually reading anything, I added a check that if it didn’t read four items, then it quits saying there was nothing more to read.

- The line ss[count] = s wasn't assigning correctly, I implemented a deep copy method to properly assign a new copy of the struct, this avoids all indices of ss pointing to the same location.
- There was no break in the 4th switch case, this means it will fall through to the default case. I added a break for this case, and it fixed the problem.
- Main wasn't returning an integer, this means we aren't specifying the exit condition of the main function, fixed to now return 0 if we make it to the end.
- I also spent a considerable amount of time in Valgrind + GDB to get rid of any errors there.
- I added a help message that shows you how to run the program and the valid options for when searching through the data.