Bug Squashing

Problem Description

You're working at a company and a junior developer has come to you for help with a task. They've been asked to find people in a list of client data (names, phone numbers, email addresses, and the like). They've attempted the task and have given you the code attached (a digital copy is also provided).

Task

Fix the problems with the code and construct a list of the things you've fixed (and a brief explanation of the problem) so that the other developer can learn.

To focus on the problem at hand, you can assume that the input is correct and valid.

Relates to Objectives

1.1-4 2.1 2.2 2.5 2.6 2.8 2.9 3.2 3.4-6 4.4 4.5 4.7 4.8

(1 point, Individual)

---

main.c

```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

struct S{
    char *firstName;
    char* lastName;
    int phone;
    char* emialAddress;
};

static int i, j;
static int count;

void sfn(struct S** ss){
    for(i = 0; i < count; i++)
        for (j = 0; j < count; j++)
            if (ss[i]->firstName > ss[j]->firstName)
                ss[i] = ss[j];
```

```
                ss[j] = ss[i];
}

int ffn(struct S** ss, char* s){
    while(++i < count)
        if(ss[i]->firstName == s)
            return 1;
    return 0;
}

void sln(struct S** ss){
    for(i = 0; i < count; i++)
        for (j = 0; j < count; j++)
            if (ss[i]->lastName > ss[j]->lastName)
                ss[i] = ss[j];
                ss[j] = ss[i];
}

int fln(struct S** ss, char* s){
    while(++i < count){
        if(ss[i]->lastName == s)
            return 1;
    }
    return 0;
}

void sem(struct S** ss){
    for(i = 0; i < count; i++) {
        for (j = 0; j < count; j++) {
            if (ss[i]->emialAddress > ss[j]->emialAddress) {
                struct S *s = ss[i];
                ss[j] = ss[i];
                ss[j] = s;
            }
        }
    }
}

int fem(struct S** ss, char* s){
    while(++i < count){
        if(ss[i]->emialAddress == s)
            return 1;
    }
    return 0;
}

void sph(struct S** ss){
    for(; i < count; i++) {
        for (; j < count; j++) {
            if (ss[i]->phone > ss[j]->phone) {
                struct S *s = ss[i];
                ss[i] = ss[j];
                ss[j] = s;
```

```
            }
        }
    }
}

int fph(struct S** ss, int s){
    while(++i < count){
        if(ss[i]->phone == s)
            return 1;
    }
    return 0;
}

int main(int argc, char ** argv) {
    int i;
    int count = 0;
    char buffer[10];

    struct S** ss = (struct S**) malloc(100*sizeof(struct S**));
    struct S* s = malloc(sizeof(*s));

    FILE *f = fopen(argv[1], "r");

    for(i = 0; i < 50; i++){

        s->firstName = (char*) malloc(80 * sizeof(s->firstName[0]));
        s->lastName = (char*) malloc(80 * sizeof(s->firstName[0]));
        s->emialAddress = (char*) malloc(80 * sizeof(s->firstName[0]));

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

        ss[count] = s;
        count += 1;
    {
        int command = 10;
        while(command != 0){
                char* val = malloc(100*sizeof(val[0]));
                gets(buffer);
                command = atoi(buffer);
                gets(buffer);
                strcpy(val, buffer);
                switch(command){
                        case 1:
                        printf("looking for email %s\n", val);
                        sem(ss);
                        printf("found it? %d\n", fem(ss, val));
                        break;
                        case 2:
                        printf("looking for firstname %s\n", val);
                        sfn(ss);
                        printf("found it? %d\n", ffn(ss, val));
                        break;
```

```
                        case 3:
                        printf("looking for lasname %s\n", val);
                        sln(ss);
                        printf("found it? %d\n", fln(ss, val));
                        break;
                        case 4:
                        printf("looking for email %s\n", val);
                        sph(ss);
                        printf("found it? %d\n", fph(ss, atoi(val)));
                        default:
                        break;
                        }
                }
        }
}}
```

---

makefile

```
main: main.c
            gcc -o main main.c -lm 2> /dev/null
```

(1 point, Individual)