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

#define NELEM 22555
#define LINESIZE 1024

struct listing {
        int id, host_id, minimum_nights, number_of_reviews, calculated_host_listings_co
unt,availability_365;
        char *host_name, *neighbourhood_group, *neighbourhood, *room_type;
        float latitude, longitude, price;
};

void displayStruct(struct listing item, FILE* fp) {
        fprintf(fp, "%d,   ", item.id);
        fprintf(fp, "%d, ", item.host_id);
        fprintf(fp, "%s, ", item.host_name);
        fprintf(fp, "%s, ", item.neighbourhood_group);
        fprintf(fp, "%s, ", item.neighbourhood);
        fprintf(fp, "%f, ", item.latitude);
        fprintf(fp, "%f, ", item.longitude);
        fprintf(fp, "%s, ", item.room_type);
        fprintf(fp, "%f, ", item.price);
        fprintf(fp, "%d, ", item.minimum_nights);
        fprintf(fp, "%d, ", item.number_of_reviews);
        fprintf(fp, "%d, ", item.calculated_host_listings_count);
        fprintf(fp, "%d\n", item.availability_365);
}

struct listing getfields(char* line){
    struct listing item;

    item.id = atoi(strtok(line, ","));
    item.host_id = atoi(strtok(NULL, ","));
    item.host_name = strdup(strtok(NULL, ","));
    item.neighbourhood_group = strdup(strtok(NULL, ","));
    item.neighbourhood = strdup(strtok(NULL, ","));
    item.latitude = atof(strtok(NULL, ","));
    item.longitude = atof(strtok(NULL, ","));
    item.room_type = strdup(strtok(NULL, ","));
    item.price = atof(strtok(NULL, ","));
    item.minimum_nights = atoi(strtok(NULL, ","));
    item.number_of_reviews = atoi(strtok(NULL, ","));
    item.calculated_host_listings_count = atoi(strtok(NULL, ","));
    item.availability_365 = atoi(strtok(NULL, ","));

    return item;
}

static int comp_price(const void* lone, const void* ltwo) {
    //https://stackoverflow.com/a/5865594/9295513
    struct listing *m1 = (struct listing*) lone;
    struct listing *m2 = (struct listing*) ltwo;
    return (m1->price > m2->price);
}

static int comp_host_name(const void* lone, const void* ltwo) {
    struct listing *m1 = (struct listing*) lone;
    struct listing *m2 = (struct listing*) ltwo;
    char* hone = m1->host_name;
    char* htwo = m2->host_name;
    int honegttwo, i; //host_name one greater than host_name two
```

```c
    char from_one = hone[i];
    char from_two = htwo[i];
    int len = strlen(hone) > strlen(htwo) ? strlen(hone) : strlen(htwo);
    while (i < len && from_two != '\0' && from_one != '\0') {
        from_one = hone[i];
        from_two = htwo[i];
        honegttwo = from_one > from_two;
        if (!honegttwo) break;
        i++;
    }
    return honegttwo;
}


int main() {
        //http://www.cplusplus.com/reference/cstdio/fopen/
        int answer;
    printf("Do you want to sort on\n1: price\nor\n2: host name\n");
        scanf("%d", &answer);
        if (answer != 1 && answer != 2) {
            printf("You must say 1 or 2\n");
            exit(-1);
        }
        struct listing listings[NELEM];
        char line[LINESIZE];
        FILE* input;
    input = fopen("listings.csv", "r");
        int i = 0;
        while (fgets(line, LINESIZE, input) != NULL){
                listings[i++] = getfields(line);
        }
        fclose(input);
        switch(answer) {
            case 1:
                qsort(listings, NELEM, sizeof(struct listing*), comp_price); break;
            case 2:
                qsort(listings, NELEM, sizeof(struct listing*), comp_host_name); break;
        }
    FILE* output;
    output = fopen("listings_sorted.csv", "w");
        int j;
    /*
    for (j = 0; j < i; j++) {
        displayStruct(listings[j], output);
    }
    */
    fclose(output);
    return 0;
}
```