

```
/* Sample program to read a comma separated file into a structure and
   display the array of structures */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LINESIZE 1024
struct listing {
    int id, host_id, minimum_nights, number_of_reviews, calculated_host_listings_count,
    availability_365;
    char * host_name, * neighbourhood_group, * neighbourhood, * room_type;
    float latitude, longitude, price;
};

struct listing getfields(char * line) {
    struct listing item;
    /* Note: you have to pass the string to strtok on the first
       invocation and then pass NULL on subsequent invocations */
    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;
}

void displayStruct(struct listing item) {
    printf("ID : %d\n", item.id);
    printf("Host ID : %d\n", item.host_id);
    printf("Host Name : %s\n", item.host_name);
    printf("Neighbourhood Group : %s\n", item.neighbourhood_group);
    printf("Neighbourhood : %s\n", item.neighbourhood);
    printf("Latitude : %f\n", item.latitude);
    printf("Longitude : %f\n", item.longitude);
    printf("Room Type : %s\n", item.room_type);
    printf("Price : %f\n", item.price);
    printf("Minimum Nights : %d\n", item.minimum_nights);
    printf("Number of Reviews : %d\n", item.number_of_reviews);
    printf("Calculated Host Listings Count : %d\n", item.calculated_host_listings_count);
    printf("Availability_365 : %d\n\n", item.availability_365);
}

// I followed the below guide on how to setup the comparators below.
// https://www.geeksforgeeks.org/comparator-function-of-qsort-in-c/
int comparePrice(const void* p1, const void* p2){
    int n = ((struct listing *)p1)->price;
    int m = ((struct listing *)p2)->price;
    return n-m;
}

int compareHostName(const void* p1, const void* p2){
    char *n = ((struct listing *)p1)->host_name;
    char *m = ((struct listing *)p2)->host_name;
    return strcmp(n,m);
}
```

```
}

int main(int argc, char * args[]) {
    struct listing list_items[22555];
    char line[LINESIZE];
    int i, count;
    FILE * fptr = fopen("listings.csv", "r");
    if (fptr == NULL) {
        printf("Error reading input file listings.csv\n");
        exit ( - 1);
    }
    fgets(line,LINESIZE,fptr); // absorbs the header string: its not needed

    count = 0;
    while (fgets(line, LINESIZE, fptr) != NULL) {
        list_items[count++] = getfields(line);
    }

    fclose(fptr);

    qsort(list_items,count,sizeof(list_items[0]),comparePrice);
    FILE * fptr1 = fopen("listings_price.csv", "w");
    fputs("id,host_id,host_name,neighbourhood_group,neighbourhood,latitude,longitude,room_type,price,minimum_nights,number_of_reviews,calculated_host_listings_count,availability_365\n",fptr1);
    for (i = 0; i < count; i++){
        //printf("%f\n",list_items[i].price);
        fprintf(fptr1,"%d,%d,%s,%s,%s,%f,%f,%s,%f,%d,%d,%d,%d\n",
            list_items[i].id,
            list_items[i].host_id,
            list_items[i].host_name,
            list_items[i].neighbourhood_group,
            list_items[i].neighbourhood,
            list_items[i].latitude,
            list_items[i].longitude,
            list_items[i].room_type,
            list_items[i].price,
            list_items[i].minimum_nights,
            list_items[i].number_of_reviews,
            list_items[i].calculated_host_listings_count,
            list_items[i].availability_365
        );
    }

    qsort(list_items,count,sizeof(list_items[0]),compareHostName);
    FILE * fptr2 = fopen("listings_hostname.csv", "w");
    fputs("id,host_id,host_name,neighbourhood_group,neighbourhood,latitude,longitude,room_type,price,minimum_nights,number_of_reviews,calculated_host_listings_count,availability_365\n",fptr2);
    for (i = 0; i < count; i++){
        //printf("%s\n",list_items[i].host_name);
        fprintf(fptr2,"%d,%d,%s,%s,%s,%f,%f,%s,%f,%d,%d,%d,%d\n",
            list_items[i].id,
            list_items[i].host_id,
            list_items[i].host_name,
            list_items[i].neighbourhood_group,
            list_items[i].neighbourhood,
            list_items[i].latitude,
            list_items[i].longitude,
            list_items[i].room_type,
            list_items[i].price,
            list_items[i].minimum_nights,
            list_items[i].number_of_reviews,
```

**lab6.c**

**Sun Oct 03 01:09:06 2021**

**3**

```
        list_items[i].calculated_host_listings_count,  
        list_items[i].availability_365  
    );  
}  
fclose(fptr2);  
return 0;  
}
```