**Nama:** Clive Clay Irawan

**NIM:** 2702373412

**Kelas:** LA01

**Subject:** AOL Algorithm and Programming

**Code Documentation**

**Study Case 1:**

#include <stdio.h>

#include <string.h>

/\*AOL AlgoProg - Case Study 1

Name: Clive Clay Irawan

NIM: 2702373412

Class: LA01

Goal: Make a program that takes a string then reverse it, followed

by inverse capitalization

\*/

void reverseString(char\* text,int length) {

for(int i = 0; i<length/2;i++){

char temp = text[i];

text[i] = text[length-i-1];

text[length - i - 1] = temp;

}

}

int main (){

char S[105];

int length;

scanf("%[^\n]", S);getchar();

length = strlen(S);

reverseString(S,length);

for(int i = 0; i<length; i++){

if(S[i]>='A'&&S[i]<='Z'){

S[i] = S[i]+32;

}else if(S[i]>='a'&&S[i]<='z'){

S[i] = S[i]-32;

}

printf("%c",S[i]);

}

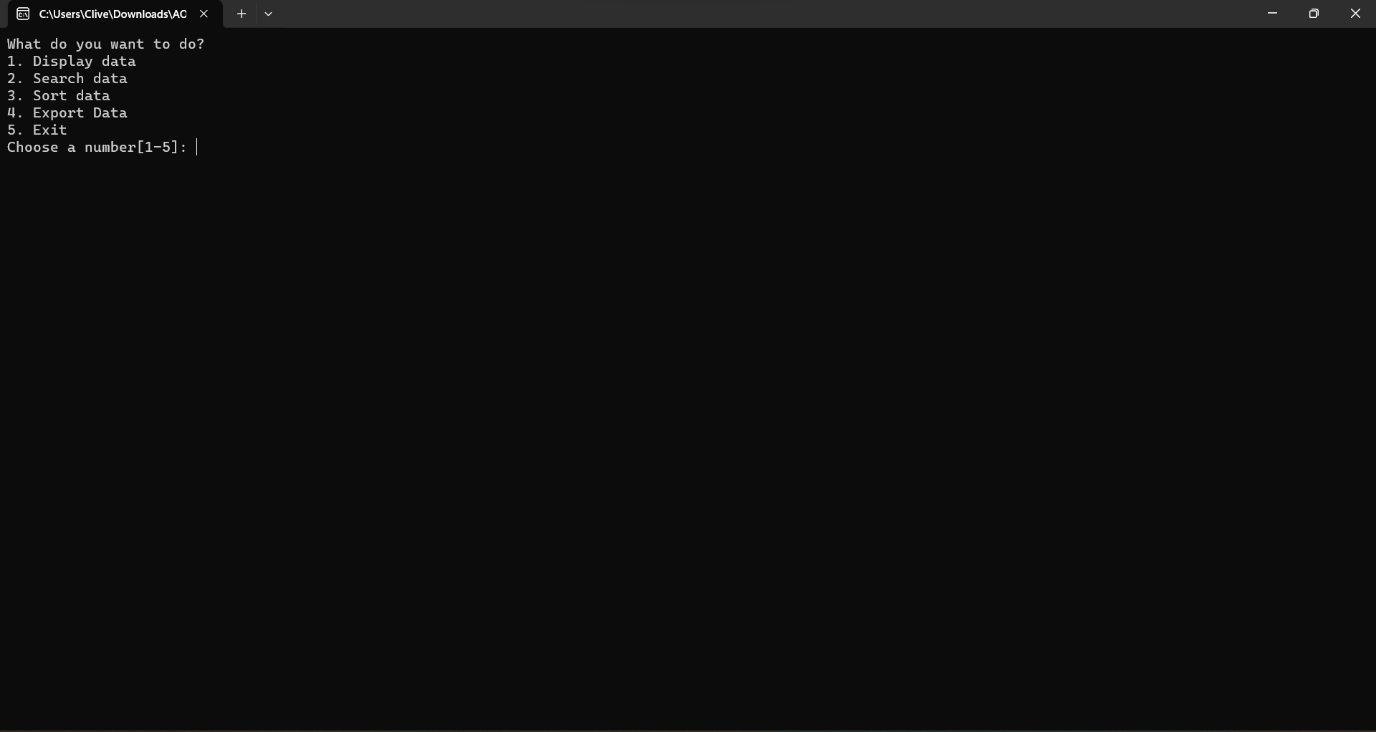
printf("\n");

return 0;

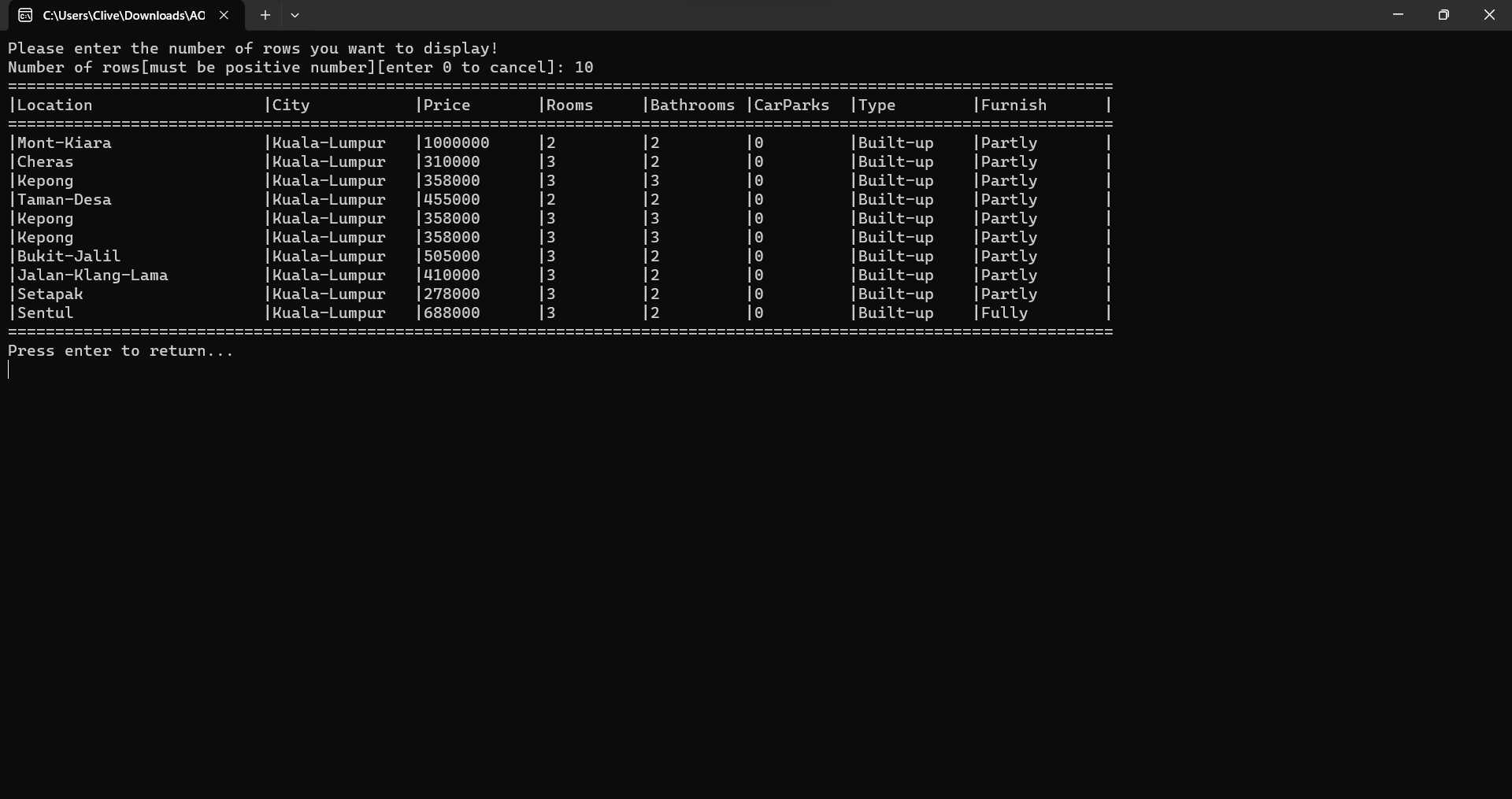
}

**Study Case 2:**

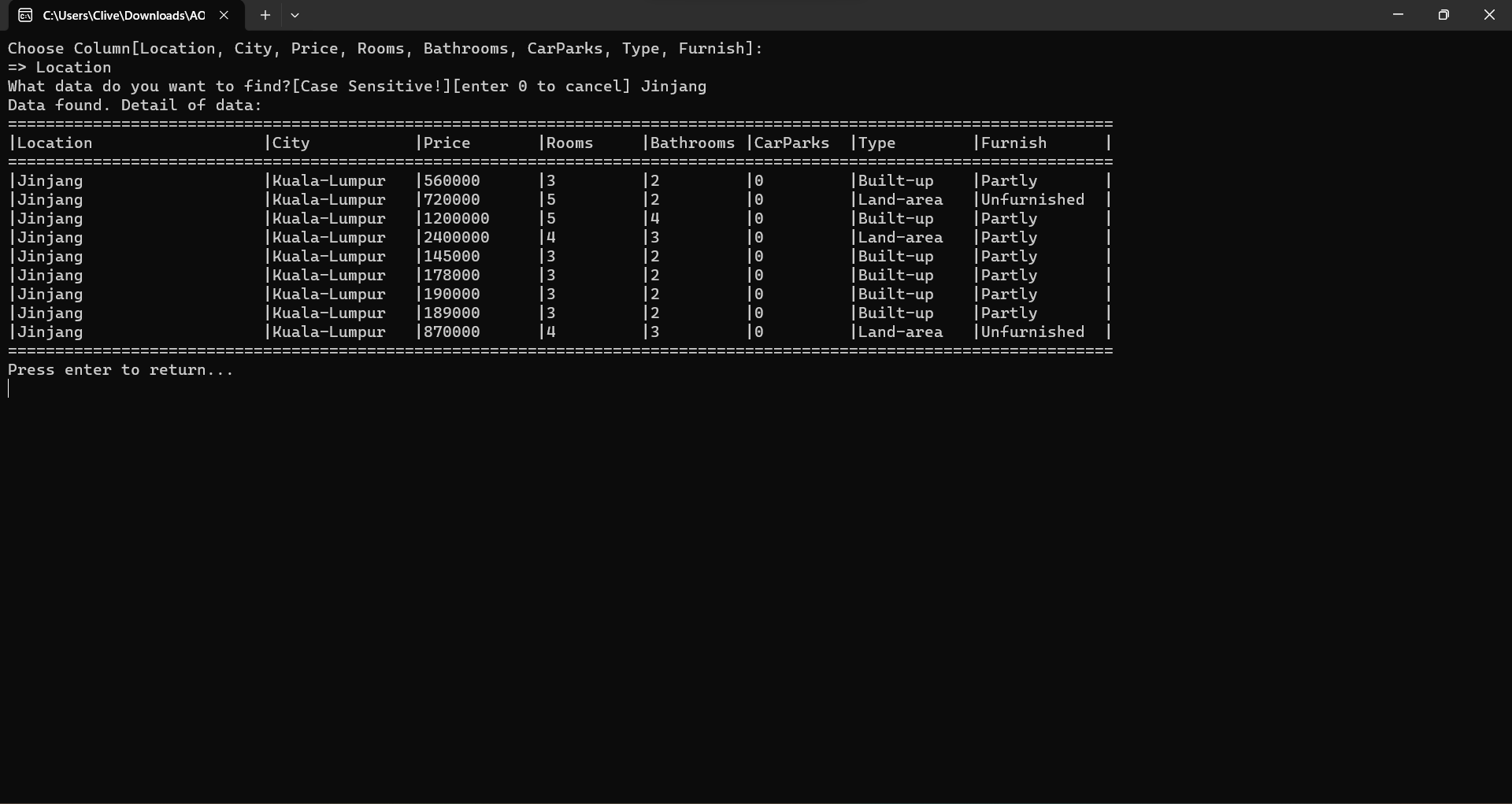
1. Main Menu:



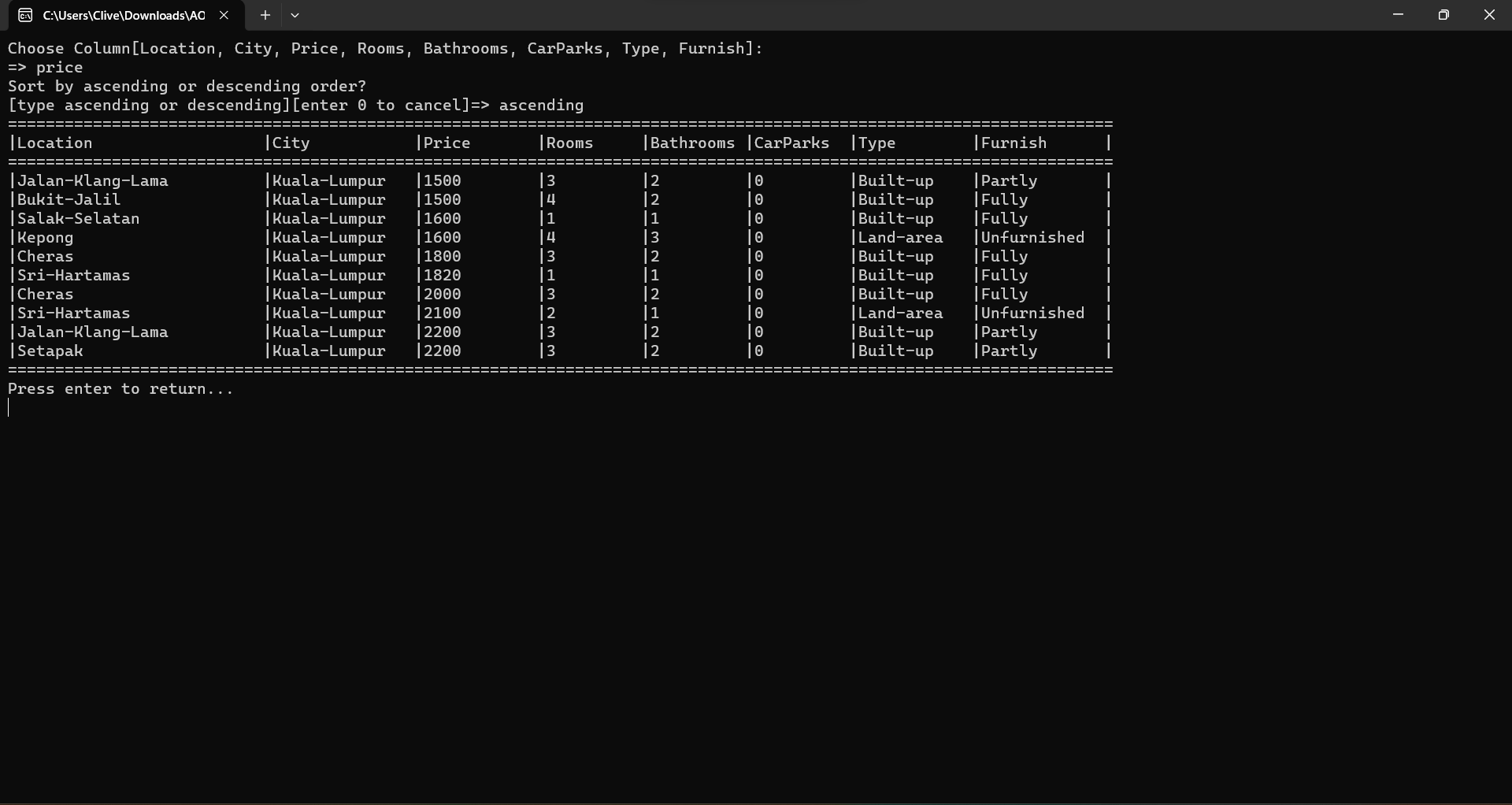
1. Menu 1



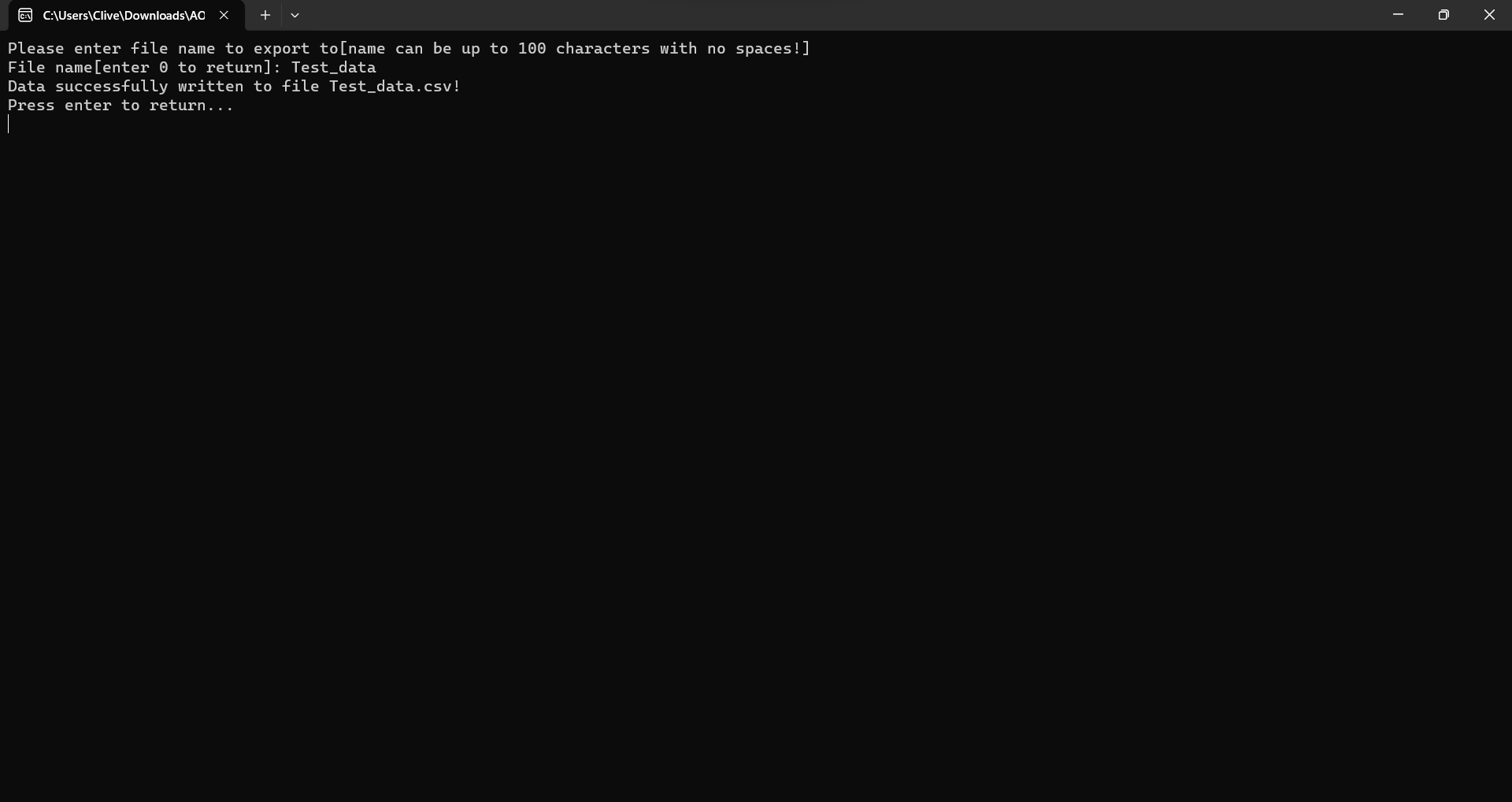
1. Menu 2



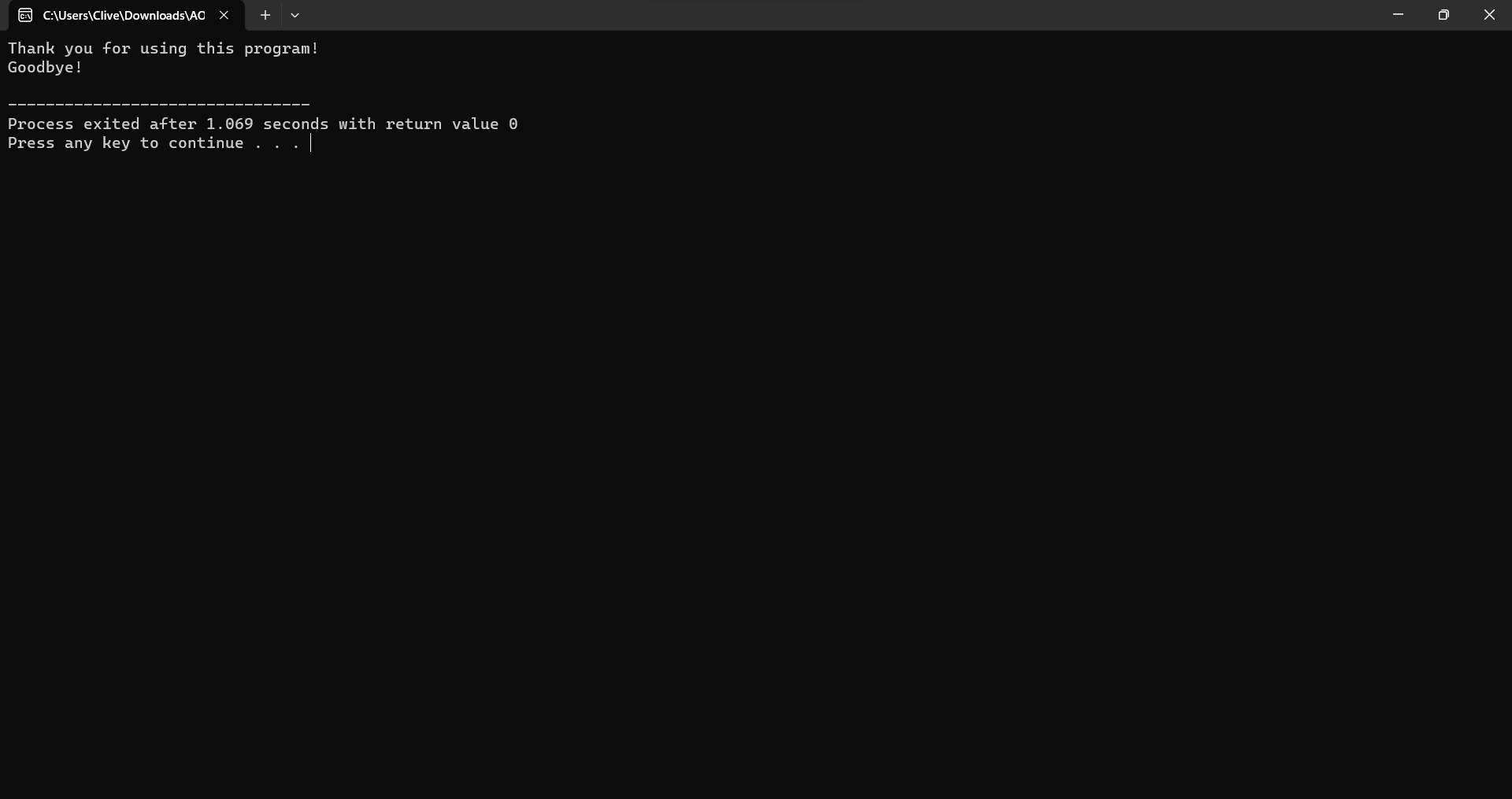
1. Menu 3



1. Menu 4



1. Menu 5



#include <stdio.h>

#include <string.h>

#include <stdlib.h>

/\*AOL AlgoProg - Case Study 2

Name: Clive Clay Irawan

NIM: 2702373412

Class: LA01

Goal: Make a program that fulfill the requirements specified in

the sheet.

\*/

struct Data{

char location[105];

char city[105];

int price;

int rooms;

int bathroom;

int carPark;

char type[25];

char furnish[25];

}sheet[3940];

char locationRow[105],cityRow[105],priceRow[105],roomsRow[105],bathroomRow[105],carparkRow[105],typeRow[105],furnishRow[105];

void swap(Data \*a, Data \*b){

//Function: bagian dari quickSort yang berguna untuk menukar antar data

Data temp = \*a;

\*a = \*b;

\*b = temp;

}

int partition(int low, int high,char columnChoice[],char sortChoice[]){

//Function: bagian dari quickSort yang berguna untuk membandingkan

int pivotIdx = high;

int i = (low-1);

for(int j=low;j<high;j++){

if(strcmp(columnChoice,priceRow)==0||strcmp(columnChoice,"price")==0){

if(strcmp(sortChoice,"descending")==0){

if(sheet[j].price>sheet[pivotIdx].price){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"ascending")==0){

if(sheet[j].price<sheet[pivotIdx].price){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,locationRow)==0||strcmp(columnChoice,"location")==0){

if(strcmp(sortChoice,"ascending")==0){

if(strcmp(sheet[j].location,sheet[pivotIdx].location)<=0){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"descending")==0){

if(strcmp(sheet[j].location,sheet[pivotIdx].location)>0){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,cityRow)==0||strcmp(columnChoice,"city")==0){

if(strcmp(sortChoice,"ascending")==0){

if(strcmp(sheet[j].city,sheet[pivotIdx].city)<=0){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"descending")==0){

if(strcmp(sheet[j].city,sheet[pivotIdx].city)>0){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,roomsRow)==0||strcmp(columnChoice,"rooms")==0){

if(strcmp(sortChoice,"descending")==0){

if(sheet[j].rooms>sheet[pivotIdx].rooms){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"ascending")==0){

if(sheet[j].rooms<sheet[pivotIdx].rooms){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,bathroomRow)==0||strcmp(columnChoice,"bathrooms")==0){

if(strcmp(sortChoice,"descending")==0){

if(sheet[j].bathroom>sheet[pivotIdx].bathroom){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"ascending")==0){

if(sheet[j].bathroom<sheet[pivotIdx].bathroom){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,carparkRow)==0||strcmp(columnChoice,"carparks")==0){

if(strcmp(sortChoice,"descending")==0){

if(sheet[j].carPark>sheet[pivotIdx].carPark){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"ascending")==0){

if(sheet[j].carPark<sheet[pivotIdx].carPark){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,typeRow)==0||strcmp(columnChoice,"type")==0){

if(strcmp(sortChoice,"ascending")==0){

if(strcmp(sheet[j].type,sheet[pivotIdx].type)<=0){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"descending")==0){

if(strcmp(sheet[j].type,sheet[pivotIdx].type)>0){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}else if(strcmp(columnChoice,furnishRow)==0||strcmp(columnChoice,"furnish")==0){

if(strcmp(sortChoice,"ascending")==0){

if(strcmp(sheet[j].furnish,sheet[pivotIdx].furnish)<=0){

i++;

swap(&sheet[i],&sheet[j]);

}

}else if(strcmp(sortChoice,"descending")==0){

if(strcmp(sheet[j].furnish,sheet[pivotIdx].furnish)>0){

i++;

swap(&sheet[i],&sheet[j]);

}

}

}

}

i++;

swap(&sheet[i],&sheet[pivotIdx]);

return i;

}

void quickSort(int low, int high,char columnChoice[],char sortChoice[]){

//Function: digunakan untuk mengsortir list

if(low>=high){

return;

}

int pi = partition(low,high,columnChoice,sortChoice);

quickSort(low,pi-1,columnChoice,sortChoice);

quickSort(pi+1,high,columnChoice,sortChoice);

}

void linearSearch(int size, char columnChoice[], char findData[],int flag){

//Function: bagian dari function 2 yang digunakan untuk membandingkan data yang dicari dengan file csv

for(int i = 0;i<size;i++){

if(strcmp(columnChoice,locationRow)==0||strcmp(columnChoice,"location")==0){

if(strcmp(sheet[i].location,findData)==0){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,cityRow)==0||strcmp(columnChoice,"city")==0){

if(strcmp(sheet[i].city,findData)==0){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,priceRow)==0||strcmp(columnChoice,"price")==0){

if(sheet[i].price==atoi(findData)){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,roomsRow)==0||strcmp(columnChoice,"rooms")==0){

if(sheet[i].rooms==atoi(findData)){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,bathroomRow)==0||strcmp(columnChoice,"bathrooms")==0){

if(sheet[i].bathroom==atoi(findData)){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,carparkRow)==0||strcmp(columnChoice,"carparks")==0){

if(sheet[i].carPark==atoi(findData)){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,typeRow)==0||strcmp(columnChoice,"type")==0){

if(strcmp(sheet[i].type,findData)==0){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}else if(strcmp(columnChoice,furnishRow)==0||strcmp(columnChoice,"furnish")==0){

if(strcmp(sheet[i].furnish,findData)==0){

flag++;

if(flag==1){

printf("Data found. Detail of data:\n");

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

}

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

}

}

if(flag==0){

printf("Data not found!\n");

return;

}

printf("=====================================================================================================================\n");

}

void readFile(int rows){

//Function: To read certain amount of data based on rows from the csv file

FILE \*fp;

fp = fopen("file.csv","r");

if (fp == NULL) {

printf("File not found!");

return;

}

fscanf(fp,"%[^,],%[^,],%[^,],%[^,],%[^,],%[^,],%[^,],%[^\n]\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

for(int i = 0; i<rows;i++){

fscanf(fp, "%[^,],%[^,],%d,%d,%d,%d,%[^,],%[^\n]\n",&sheet[i].location, &sheet[i].city,&sheet[i].price,&sheet[i].rooms,&sheet[i].bathroom,&sheet[i].carPark,&sheet[i].type,&sheet[i].furnish);

}

fclose(fp);

}

void readFileAll() {

//Function: To read every data from the csv file

FILE \*fp;

fp = fopen("file.csv", "r");

if (fp == NULL) {

printf("File not found!");

return;

}

fscanf(fp,"%[^,],%[^,],%[^,],%[^,],%[^,],%[^,],%[^,],%[^\n]\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

int i = 0;

while (fscanf(fp, "%[^,],%[^,],%d,%d,%d,%d,%[^,],%[^\n]\n",

&sheet[i].location, &sheet[i].city, &sheet[i].price,

&sheet[i].rooms, &sheet[i].bathroom, &sheet[i].carPark,

&sheet[i].type, &sheet[i].furnish) == 8) {

i++;

}

fclose(fp);

}

void displayData(int rows){

//Function: To display data from the file based on the amount of rows that the user request

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

for(int i = 0; i<rows;i++){

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,

sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

printf("=====================================================================================================================\n");

printf("Press enter to return...\n");getchar();

}

void searchData(char columnChoice[],char findData[]){

//Function: To search data based on the selected column and the data that the user requested

int size = 3940;

readFileAll();

int flag = 0;

linearSearch(size-2,columnChoice,findData,flag);

printf("Press enter to return...\n");getchar();

}

void sortData(char columnChoice[],char sortChoice[]){

//Function: To sort data based on the selected column and the chosen order

readFileAll();

int size = 3940;

quickSort(0,size-2,columnChoice,sortChoice);

printf("=====================================================================================================================\n");

printf("|%-26s|%-15s|%-12s|%-10s|%-10s|%-10s|%-12s|%-13s|\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

printf("=====================================================================================================================\n");

for(int i = 0; i<10;i++){

printf("|%-26s|%-15s|%-12d|%-10d|%-10d|%-10d|%-12s|%-13s|\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,

sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

printf("=====================================================================================================================\n");

printf("Press enter to return...\n");getchar();

}

void exportData(const char \*fileName){

//Function: To export data to a new csv file

readFileAll();

int size = 3940;

FILE \*csvFile;

char \*csvFileName;

csvFileName = (char \*)malloc(strlen(fileName));

strcpy(csvFileName, fileName);

strcat(csvFileName, ".csv");

csvFile = fopen(csvFileName,"w");

fprintf(csvFile,"%-26s, %-15s, %-10s, %-6s, %-10s, %-10s, %-10s, %-10s\n",locationRow,cityRow,priceRow,roomsRow,bathroomRow,carparkRow,typeRow,furnishRow);

for(int i = 0; i<size-1;i++){

fprintf(csvFile,"%-26s, %-15s, %-10d, %-6d, %-10d, %-10d, %-10s, %-10s\n",sheet[i].location,sheet[i].city,sheet[i].price,sheet[i].rooms,

sheet[i].bathroom,sheet[i].carPark,sheet[i].type,sheet[i].furnish);

}

fclose(csvFile);

free(csvFileName);

printf("Data successfully written to file %s.csv!\n",fileName);

printf("Press enter to return...\n");getchar();

}

int main (){

int choice;

do{

printf("What do you want to do?\n");

printf("1. Display data\n");

printf("2. Search data\n");

printf("3. Sort data\n");

printf("4. Export Data\n");

printf("5. Exit\n");

printf("Choose a number[1-5]: ");

scanf("%d", &choice);getchar();

system("cls");

switch(choice){

case 1:

int rows;

printf("Please enter the number of rows you want to display!\n");

printf("Number of rows[must be positive number][enter 0 to cancel]: ");

scanf("%d",&rows);getchar();

if(rows==0){

system("cls");

break;

}

readFile(rows);

displayData(rows);

system("cls");

break;

case 2:

char columnChoice[105],findData[105];

printf("Choose Column[Location, City, Price, Rooms, Bathrooms, CarParks, Type, Furnish]:\n");

printf("=> ");

scanf("%s",columnChoice);

printf("What data do you want to find?[Case Sensitive!][enter 0 to cancel] ");

scanf("%s",findData);getchar();

if(strcmp(findData,"0")==0){

system("cls");

break;

}

searchData(columnChoice,findData);

system("cls");

break;

case 3:

char columnChoice1[105],sortChoice[15];

printf("Choose Column[Location, City, Price, Rooms, Bathrooms, CarParks, Type, Furnish]:\n");

printf("=> ");

scanf("%s",columnChoice1);

printf("Sort by ascending or descending order?\n");

printf("[type ascending or descending][enter 0 to cancel]=> ");

scanf("%s",sortChoice);getchar();

if(strcmp(sortChoice,"0")==0){

system("cls");

break;

}

sortData(columnChoice1,sortChoice);

system("cls");

break;

case 4:

char fileName[105];

printf("Please enter file name to export to[name can be up to 100 characters with no spaces!]\n");

printf("File name[enter 0 to return]: ");

scanf("%s",fileName);getchar();

if(strcmp(fileName,"0")==0){

system("cls");

break;

}

exportData(fileName);

system("cls");

break;

case 5:

printf("Thank you for using this program!\n");

printf("Goodbye!\n");

return 0;

}

}while(choice>=1 && choice<=5||choice<1||choice>5);

return 0;

}