//The program works without any errors, everything wanted has been tested and it's one to one

//I used bubble sort to sort my arrays of struct. I know that there are so many ways to implement that,

//but this seemed like the most effective way to me. If there any more effective way than this code, can you pls feedback me hocam?

//I paid attention to leave the main function blank as much as possible and do most of the operations in other functions, can you also

//give feedback for this? That's the way we had better to do right?

//Name: Mehmet Fatih Celik

//ID: 2385268

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct cars{

char carModel[10];

int year;

double price;

};

void menu(struct cars \*myCars, int \*carNum, int \*myCarNum){

int choice;

printf("BMW Cars\n\n");

do{

printf("1) Add car\n");

printf("2) Find the cheapest car\n");

printf("3) Store cars and exit\n");

printf("What would you like to do? ");

scanf("%d",&choice);

if (choice==1){

addCar(myCars, carNum, myCarNum);

}

else if (choice==2){

findCheapest(myCars, myCarNum);

}

else if (choice==3){

storeCars(myCars, myCarNum);

}

else{

printf("\nPlease enter a valid number!\n\n");

}

}while(choice!=3);

}

void storeCars(struct cars \*myCars, int \*myCarNum){

int i,j;

struct cars temp;

for(i=0;i<(\*myCarNum)-1;i++){

for(j=0;j<((\*myCarNum)-1-i);j++){

if(myCars[j].year<myCars[j+1].year){

temp = myCars[j];

myCars[j] = myCars[j+1];

myCars[j+1] = temp;

}

}

}

FILE \*fptr;

fptr=fopen("BMWcars.txt","w");

if(fptr == NULL){

printf("Error!");

exit(1);

}

fprintf(fptr, "Car Model\tCar Year\tCar Price\n------------------------------------------\n");

for(i=0;i<\*myCarNum;i++){

fprintf(fptr, "%s\t%d\t\t%.0lf\n",myCars[i].carModel,myCars[i].year,myCars[i].price);

}

printf("BMWCars.txt is successfully created!");

fclose(fptr);

}

void findCheapest(struct cars \*myCars, int \*myCarNum){

int i, year = myCars[0].year;

char car\_model[20];

strcpy(car\_model,myCars[0].carModel);

double cheapest = myCars[0].price;

for(i=1;i<\*myCarNum;i++){

if (cheapest > myCars[i].price){

cheapest = myCars[i].price;

year = myCars[i].year;

strcpy(car\_model,myCars[i].carModel);

}

}

printf("Cheapest BMW car is %s %d %.0lf$!\n\n",car\_model,year,cheapest);

}

void addCar(struct cars \*myCars, int \*carNum, int \*myCarNum){

if (\*myCarNum == \*carNum){

myCars = (struct cars \*)realloc(myCars,((\*myCarNum)+1)\*sizeof(struct cars));

if (myCars == NULL){

printf("Unsuccesfull allocation!\n");

exit(1);

}

(\*carNum)++;

}

printf("Enter the model of the car: ");

fflush(stdin);

gets(myCars[\*myCarNum].carModel);

printf("Enter the year of the car: ");

scanf(" %d",&myCars[\*myCarNum].year);

printf("Enter the price of the car: ");

scanf("%lf",&myCars[\*myCarNum].price);

printf("\n%s is added!\n\n",myCars[\*myCarNum].carModel);

(\*myCarNum)++;

}

int main(){

struct cars \*myCars;

int carNum=2, myCarNum=0;

myCars = (struct cars \*)malloc(carNum\*sizeof(struct cars));

if (myCars == NULL){

printf("Unsuccesfull allocation!\n");

exit(1);

}

menu(myCars, &carNum, &myCarNum);

free(myCars);

return 0;

}