```c
///HEADER
///-----------------------------------------------------------------
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

///COSTANTI
#define CTAR 8
#define CMOD 16
#define CTIPO 11
#define LEN 64

///STRUTTURA RECORD
typedef struct{
    char tar[CTAR];
    char mod[CMOD];
    char tipo[CTIPO];
    float prc;
    int km;
}TAuto;

typedef struct TNodo{
    TAuto car;
    struct TNodo* next;
}TNodo;

///PROCEDURE ESERCIZIO
TNodo* imp(TNodo* first, char *nf);
void stampa(TNodo *f);
void stampaEl(TNodo *el);
TNodo* ricTar(TNodo *f, char *elTar);
bool contrTipo(char *elTipo);
void stampaTar(TNodo *f);
void espTipo(TNodo *f, char* nf);
char* myStrConcat(char *s1, char *s2);
int length(char *s);
void guadTar(TNodo* f);
float earn(TAuto el);


///INPUT + VARIE
int lgInt(int vmin, int vmax, char* mex);           //LEGGI INT
float lgFlt(float vmin, float vmax, char* mex);     //LEGGI FLOAT
char lgChar(char *mex);                             //LEGGI CHAR
void lgStr(char *s, char *mex);                     //LEGGI STRING
void menu();
void err();



///FUNZIONI ESERCIZIO
///-----------------------------------------------------------------
#include "header.h"

/*
    DESCRIZIONE FILE .csv:
    il file f.csv conterra' un elemento per riga, con i dati separati da un ';'
    ESEMPIO:
    AA000AA;Ford;Utilitaria;99.99;120;\n

    DESCRIZIONE VARIABILI:
    first, primo elemento della lista, TNodo*
    nf, nome file, char*
    f, file, FILE*
    dim, dimensione record, size_t
```

```
     str, riga letta dal file, array di LEN char
     tro, indirizzo dell'eventuale targa non univoca, TNodo*
     cor, verifica che un tipo sia corretto, bool
     car, record ausiliario in cui importare i dati , TAuto
     el, nodo della lista appena creato, TNodo*

     PSEUDOCODIFICA:
     INIZIO
         first = NULL
         apri il file nf come f (input)
         se f esiste
         allora
             leggi la prima riga dal file f in str
             mentre il file non e' finito
                 spezzo str nei suoi item
                 assegno il primo item a car.tar
                 ricerco car.tar nella lista e ritorno il suo indirizzo in tro se trovato altrimenti NULL
                 se !tro
                 allora
                     assegno il secondo e terzo item a car.mod e .tipo
                     verifico che car.tipo sia corretto e ritorno true in cor se corretto, altrimenti false
                     se cor
                     allora
                         assegno il quarto item a car.prc
                         se car.prc > 0
                         allora
                             assegno l'ultimo item a car.km
                             se car.km
                         fse
                     fse
                 fse
             fciclo
         altrimenti
             errore
         fse
     FINE
 */
TNodo* imp(TNodo* first, char *nf){
    FILE* f = fopen(nf, "r");
    if(f != NULL){
        size_t dim = sizeof(TNodo);
        char str[LEN];
        TNodo* tro;
        bool cor;
        fgets(str,LEN,f);
        while(!feof(f)){
            TAuto car;
            strcpy(car.tar, strtok(str,";"));
            tro = ricTar(first, car.tar);
            if(tro == NULL){
                strcpy(car.mod, strtok(NULL,";"));
                strcpy(car.tipo, strtok(NULL,";"));
                cor = contrTipo(car.tipo);
                if(cor){
                    car.prc = atof(strtok(NULL,";"));
                    if(car.prc > 0){
                        car.km = atoi(strtok(NULL,";"));
                        if(car.km > 0){
                            TNodo* el = (TNodo*)malloc(dim);
                            el->car = car;
                            if(first == NULL){
                                first = el;
                                el->next = NULL;
                            }
```

```
131                             else{
132                                 el->next = first;
133                                 first = el;
134                             }
135                         }
136                     }
137                 }
138             }
139             fgets(str,LEN,f);
140         }
141     }
142     else
143         err();
144     return first;
145 }
146 
147 TNodo* ricTar(TNodo *f, char *elTar){
148     TNodo* tro = NULL;
149     TNodo* p = f;
150     while(p != NULL && tro == NULL){
151         if(strcmp(elTar, p->car.tar) == 0)
152             tro = p;
153         p = p->next;
154     }
155     return tro;
156 }
157 
158 bool contrTipo(char *elTipo){
159     bool cor = true;
160     if(strcmp(elTipo, "Utilitaria") != 0 && strcmp(elTipo, "Lusso") != 0 && strcmp(
elTipo, "Comfort") != 0)
161         cor  = false;
162 
163     return cor;
164 }
165 
166 void stampa(TNodo *f){
167     if(f != NULL){
168         TNodo* el = f;
169         while(el != NULL){
170             stampaEl(el);
171             el = el->next;
172         }
173     }
174     else
175         printf("Importa prima i dati\n");
176 }
177 
178 void stampaEl(TNodo *el){
179     printf("Targa: %s | ", el->car.tar);
180     printf("Modello: %s | ", el->car.mod);
181     printf("Tipologia: %s | ", el->car.tipo);
182     printf("Prezzo al giorno: %.2f | ", el->car.prc);
183     printf("Kilometraggio: %d\n", el->car.km);
184 }
185 
186 /*
187     f, primo elemento della lista, TNodo*
188     ausTar, targa da cercare nella lista, array di CTAR char
189     tro, indirizzo dell'eventuale record da stampare
190 
191     leggi ausTar
192     ricerca ausTar nella lista e ritorna in tro il suo indirizzo o NULL
193     se ausTar != NULL
194     allora
195         stampa il record puntato da tro
```

```
    altrimenti
        scrivi "elemento non trovato"
    fse
*/
void stampaTar(TNodo *f){
    if(f != NULL){
        char ausTar[CTAR];
        lgStr(ausTar, "Inserire targa auto interessata: ");
        TNodo* tro = ricTar(f,ausTar);
        if(tro != NULL){
            stampaEl(tro);
        }
        else
            printf("Elemento non trovato\n");
    }
    else
        printf("Importa prima i dati\n");
}

/*
    f, primo nodo della lista, TNodo*
    nf, nome file, char*
    ausTipo, tipo auto da esportare, array di CTIPO char
    fout, file su cui esportare, FILE*
    cor, verifica che ausTipo sia corretto
    dim, dimensione record da esportare, size_t
    el, nodo della lista, TNodo*

    leggi ausTipo e controlla che sia =="Utilitaria" || =="Lusso" || =="Comfort"
    leggi nf e controlla che sia !=""
    apri il file nf come fe (output)
    per ogni elemento el della lista
        se el->car.tipo == ausTipo
        allora
            scrivi el nel file
        fse
    fciclo
*/
void espTipo(TNodo *f, char* nf){
    if(f != NULL){
        char ausTipo[CTIPO];
        FILE* fout;
        bool cor;
        size_t dim = sizeof(TAuto);
        TNodo* el;
        lgStr(ausTipo, "Inserire tipo interessato: ");
        cor = contrTipo(ausTipo);
        while(!cor){
            err();
            lgStr(ausTipo, "Inserire tipo interessato: ");
            cor = contrTipo(ausTipo);
        }
        fout = fopen(nf,"w");
        el = f;
        while(el != NULL){
            if(strcmp(el->car.tipo, ausTipo) == 0)
                fwrite(&(el->car),dim,1,fout);
            el = el->next;
        }
    }
    else
        printf("Importa prima i dati\n");
}

char* myStrConcat(char *s1, char *s2){
    int d1 = length(s1);
```

```
    int d2 = length(s2);
    size_t dim = d1 + d2 + 1;
    char* strCon = (char*)malloc(dim);
    int k, j;
    for(k = 0; k<d1; k++){
       *(strCon + k) = *(s1 + k);
    }
    for(j = 0; j<d2; j++){
        *(strCon + k + j) = *(s2 + j);
    }
    *(strCon + k + j) = '\0';

    return strCon;
}

int length(char *s){
    int n = 0;
    while(*(s+n) != '\0')
        n++;
    return n;
}

/*
    leggi ausTar e controlla che sia !=""
    ricerca ausTar nella lista e ritorna in tro il suo indirizzo o NULL
    se tro != NULL
    allora
        calcolo il guadagno in guad
        scrivo "Guadagno: " guad
    altrimenti
        scrivi "elemento non trovato"
    fse

*/
void guadTar(TNodo* f){
    if(f != NULL){
        char ausTar[CTAR];
        lgStr(ausTar,"Inserire targa veicolo interessato: ");
        TNodo* p = ricTar(f,ausTar);
        if(p != NULL){
            float guad = earn(p->car);
            printf("Guadagno: %.2f\n",guad);
        }
        else
            printf("Elemento non trovato\n");
    }
    else
        printf("Importa prima i dati\n");
}

/*
    Per il calcolo del guadagno usare una funzione parametrizzata che segua le
seguenti regole:
    - per il livello "utilitaria" il guadagno al km e' 0,44 euro;
    - per il livello "lusso" il guadagno al km e' 1,99 euro;
    - per il livello "confort" il guadagno al km e' 0,99 euro.
*/
float earn(TAuto el){
    float t = 0;
    if(strcmp(el.tipo,"Utilitaria") == 0)
        t = el.km * 0.44;
    else{
        if(strcmp(el.tipo,"Lusso") == 0)
            t = el.km * 1.99;
        else
            t = el.km * 0.99;
```

```
    }
    return t;
}


///MAIN
///-----------------------------------------------------------------
#include "header.h"

/*
    Francesco Cucchi 4BI

    TESTO:
    Creare un software per la gestione delle auto poste a noleggio di una azienda. Per ogni auto vengono memorizzate
    le seguenti informazioni:
    - Targa (Univoco)
    - Modello
    - tipo auto (utilitaria, lusso, confort)
    - prezzo al giorno
    - km fatti dall'acquisto.
    Realizzare un programma che permetta di:
    1. Caricare da file *.csv il parco macchine dall'azienda disponibili per il noleggio.
    2. Stampa tutte le auto contenute nella lista.
    3. Visualizzare i dati di un'auto di cui si fornisce la targa.
    4. Salvare su file binario tutti le auto di un tipo scelto in input dall'utente.
    5. Data in input una targa calcolare il guadagno in base ai km effettuati.
    Per il calcolo del guadagno usare una funzione parametrizzata che segua le seguenti regole:
    - per il livello "utilitaria" il guadagno al km e' 0,44 euro;
    - per il livello "lusso" il guadagno al km e' 1,99 euro;
    - per il livello "confort" il guadagno al km e' 0,99 euro.

    Obblighi/Limiti:
    - Ogni singolo punto del menu deve essere preceduto dallo pseudocodice con descrizione dei parametri,
    delle variabili locali e delle strutture dei file creati.
    - Il progetto dovrà essere realizzato dividendo il codice in piu' file.
    - Per gli input si dovranno utilizzare le apposite funzioni realizzate.
*/

int main()
{
    int sc;
    TNodo* first = NULL;
    char nf[CMOD];
    do{
        menu();
        sc = lgInt(0,5,"Selezionare funzione interessata: ");
        system("cls");
        switch(sc){
            case 1:{
                lgStr(nf,"Inserire nome file: ");
                first = imp(first, nf);
                break;
            }
            case 2:{
                stampa(first);
                break;
            }
            case 3:{
                stampaTar(first);
                break;
            }
```

```
            case 4:{
                lgStr(nf,"Inserire nome file: ");
                espTipo(first, nf);
                break;
            }
            case 5:{
                guadTar(first);
                break;
            }
            case 0: break;
        }
        system("pause");
        system("cls");
    }while(sc!=0);
    return 0;
}



///INPUT
///----------------------------------------------------------------------
#include "header.h"

int lgInt(int vmin, int vmax, char* mex){
    int x;
    printf("%s", mex);
    scanf("%d", &x);
    while(x<vmin || x>vmax){
        printf("%s", "Errore\n");
        printf("%s", mex);
        scanf("%d", &x);
    }
    return x;
}

float lgFlt(float vmin, float vmax, char* mex){
    float x;
    printf("%s", mex);
    scanf("%f", &x);
    while(x<vmin || x>vmax){
        printf("%s", "Errore\n");
        printf("%s", mex);
        scanf("%f", &x);
    }
    return x;
}

char lgChar(char *mex){
    char c;
    printf("%s",mex);
    fflush(stdin);
    scanf("%s",&c);
    while(c == '\0'){
        err();
        printf("%s",mex);
        fflush(stdin);
        scanf("%s",&c);
    }
    return c;
}

void lgStr(char *s, char *mex){
    printf("%s", mex);
    fflush(stdin);
    gets(s);
    while(strcmp(s, "") == 0){
```

```
455         err();
456         printf("%s", mex);
457         fflush(stdin);
458         gets(s);
459     }
460 }
461
462 void menu(){
463     printf("1) Importazione da file .csv;\n");
464     printf("2) Stampa tutte le auto;\n");
465     printf("3) Visualizza dati auto data la targa;\n");
466     printf("4) Salvare su file binario tutte le auto di un tipo deciso in input;\n"
);
467     printf("5) Data targa calcolare guadagno;\n");
468     printf("0) Termina programma;\n");
469 }
470
471 void err(){
472     printf("Errore\n");
473 }
```