

Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali
Prof. Penzo
Soluzione compito del 25/06/2010

Esercizio 1

```
#include<stdio.h>
#include<string.h>
#include<math.h>

#define MAX_COPPIE 100
#define MAX_CATEGORIE 20

typedef enum{ false, true } boolean;
typedef enum{ combustione, ibrida, elettrica } tipoAuto;

typedef struct{
    char marca[20];
    char modello[20];
}coppia;

typedef struct{
    int codice;
    coppia tipologie[MAX_COPPIE];
    int numCoppie;
    float costo;
    int punti;
} categoria;

// prototipi delle funzioni
boolean contieneMarcaModello(char marca[], char modello[], coppia C[], int numC);
void aggiornaCategoria(char marca[], char modello[], int codiceCategoria, categoria Listino[], int
numCategorie, int *numCoppie);
float costoEPuntiPerAuto(char targa[], char fileAutomobili[], int *puntiLavaggio, categoria
Listino[], int numCategorie);
void nuovoLavaggio(char targa[], tipoAuto tipo, char fileAutomobili[], categoria Listino[], int
numCategorie, float *costoLavaggio, int *puntiLavaggio);

void main(){
categoria LISTINOPREZZI[MAX_CATEGORIE];
int numCategorie; // si assume contenga il numero di elementi correntemente presenti nel listino
char targa[8];
float costoLavaggio;
int puntiLavaggio;
int valoreTipoAuto;
tipoAuto tipo;

// punto a)
printf("Inserire la targa: ");
scanf("%s", targa);
costoLavaggio = costoEPuntiPerAuto(targa,"AUTOMOBILI.TXT",
&puntiLavaggio,LISTINOPREZZI, numCategorie);
```

```
printf("Il costo del lavaggio e': %f\n", costoLavaggio);
printf("Il numero di punti accumulati con il lavaggio e': %d", puntiLavaggio);
```

// punto b)

```
printf("Inserire il tipo di auto (1 per combustione, 2 per ibrida, 3 per elettrica)");
scanf("%d",&valoreTipoAuto);
switch(valoreTipoAuto){
    case(1): tipo = combustione;
            break;
    case(2): tipo = ibrida;
            break;
    case(3): tipo = elettrica;
            break;
}
nuovoLavaggio(targa,tipo,"AUTOMOBILI.TXT",LISTINOPREZZI, numCategorie,
&costoLavaggio, &puntiLavaggio);
printf("Il costo del lavaggio e': %f\n", costoLavaggio);
printf("Il numero di punti accumulati con il lavaggio e': %d", puntiLavaggio);
}
```

// punto 1)

```
boolean contieneMarcaModello(char marca[], char modello[], coppia C[], int numC){

    boolean trovato = false;
    int i=0;

    while(i<numC && !trovato){
        if(!strcmp(C[i].marca,marca) && !strcmp(C[i].modello,modello))
            trovato = true;
        i++;
    }
    return trovato;
}
```

// punto 2)

```
void aggiornaCategoria(char marca[], char modello[], int codiceCategoria, categoria Listino[], int
numCategorie, int *numCoppie){

    boolean trovata = false;
    coppia c;
    int i=0;

    *numCoppie=0;
    while(i<numCategorie && !trovata){
        if(Listino[i].codice == codiceCategoria){
            trovata = true;
            if(Listino[i].numCoppie < MAX_COPPIE){
                strcpy(c.marca, marca);
                strcpy(c.modello, modello);
                Listino[i].tipologie[Listino[i].numCoppie] = c;
            }
        }
        i++;
    }
}
```

```

        Listino[i].numCoppie++;
    }
    else
        printf("Errore: elenco pieno");
    *numCoppie = Listino[i].numCoppie;
}
i++;
}
if(!trovata)
    printf("Errore: categoria non trovata!\n");
}

```

// punto 3)

```

float costoEPuntiPerAuto(char targa[], char fileAutomobili[], int *puntiLavaggio, categoria
Listino[], int numCategorie){

```

```

    boolean trovataAuto=false, trovataCategoria=false;
    FILE *fp;
    char targaAuto[8],marca[20],modello[20];
    int anno, punti;
    float costo=0;

    fp = fopen(fileAutomobili,"rt");
    if(fp == NULL)
        printf("Errore: file non trovato!\n");
    else{
        fscanf(fp, "%s%s%s%d%d",targaAuto,marca,modello,&anno,&punti);
        while(!feof(fp) && !trovataAuto){
            if(!strcmp(targaAuto,targa))
                trovataAuto = true;
            else
                fscanf(fp, "%s%s%s%d%d",targaAuto,marca,modello,&anno,&punti);
        }
    }
    fclose(fp);
    if(trovataAuto){
        for(int i=0; i<numCategorie && !trovataCategoria;i++){
            trovataCategoria = contieneMarcaModello(marca, modello,
Listino[i].tipologie, Listino[i].numCoppie);
            if(trovataCategoria){
                costo = Listino[i].costo;
                *puntiLavaggio = Listino[i].punti;
            }
        }
        if(!trovataCategoria)
            printf("Errore: categoria non trovata!\n");
    }
    else
        printf("Errore: auto non trovata!\n");

    return costo;
}

```

```
}
```

// punto 4)

```
void nuovoLavaggio(char targa[], tipoAuto tipo, char fileAutomobili[], categoria Listino[], int  
numCategorie, float *costoLavaggio, int *puntiLavaggio){
```

```
    *costoLavaggio = costoEPuntiPerAuto(targa, fileAutomobili, puntiLavaggio, Listino,  
numCategorie);
```

```
    switch(tipo){
```

```
        case(combustione): break;
```

```
        case(ibrida):
```

```
            *costoLavaggio = (*costoLavaggio)*90/100;
```

```
            *puntiLavaggio = (int)ceil((double)(*puntiLavaggio)*90/100);
```

```
            break;
```

```
        case(elettrica):
```

```
            *costoLavaggio = (*costoLavaggio)*85/100;
```

```
            *puntiLavaggio = (int)ceil((double)(*puntiLavaggio)*85/100);
```

```
            break;
```

```
    }
```

```
}
```