

## Istruzioni

- Tempo disponibile: 90 minuti.
- Non è permesso l'uso di dispositivi elettronici (a parte il PC della propria postazione).
- Il programma sarà valutato per
  - Identificazione delle strutture dati e degli algoritmi appropriati alle specifiche
  - Corretta implementazione di strutture dati e algoritmi
  - Utilizzo efficiente delle risorse
  - Stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- I programmi non compilabili saranno valutati 0 punti.
- Fare l'upload di tutti i file che compongono il programma.
- Visual Studio Code è installato su tutte le postazioni. Si può comunque usare l'editor che si preferisce fra quelli installati.

## Esercizio - Parte 1 (max 8 punti)

In un file binario sono registrati gli accessi a un parcheggio a pagamento nell'ultimo mese. In particolare, ogni accesso al parcheggio è rappresentato da un record formato da

- targa del veicolo (stringa di 8 `char`, compreso il terminatore)
- durata della sosta in ore (`float`).

Ad esempio, il file binario allegato `accessi.dat` contiene i dati nella tabella 1.

Il prezzo di ogni accesso è determinato indipendentemente dalla durata, ma dipende dal numero di totale accessi durante il mese. In particolare

- se il veicolo ha effettuato fino a 5 accessi, ogni accesso costa 2.00€;
- se il veicolo ha effettuato da 6 a 10 accessi, ogni accesso costa 1.90€;
- se il veicolo ha effettuato più di 10 accessi, ogni accesso costa 1.80€;

Scrivere un programma in linguaggio C, da compilare in un eseguibile di nome `fatture` che

- riceva come argomento della linea di comando il nome di un file del formato sopra indicato;

Targa	Durata in ore
DB478AR	0.168078
BR922WF	0.314526
AB127AR	3.400156
BR922WF	2.993127
BW753BM	1.935185
DB478AR	2.684846
AB127AR	2.370261
BR922WF	0.233553
DB478AR	3.268239
DB478AR	2.551238
FO062YH	2.611017
CD920CD	1.303766
AB610CD	0.664312
BR922WF	3.092256
BR922WF	1.223146
CD920CD	3.137466
FO062YH	2.434678
BR922WF	2.101377
BR922WF	3.019500
CD920CD	3.241497
FO062YH	2.870208
DB478AR	3.197305
CD920CD	0.833805
FO062YH	3.228436
CD920CD	3.095611
AB127AR	1.568642
DX612BL	1.055602
BR922WF	1.145357
DX612BL	0.868763
DB478AR	1.039198

Tabella 1: Contenuto del file allegato [accessi.dat](#)

- stampi a video una riga per ogni veicolo distinto, contenente la targa e l'importo dovuto.

Ad esempio, se `accessi.dat` è il file allegato, l'invocazione  
`./fatture accessi.dat`  
deve produrre un output simile al seguente:

```
DB478AR 11.40
BR922WF 15.20
AB127AR 6.00
BW753BM 2.00
FO062YH 8.00
CD920CD 10.00
AB610CD 2.00
DX612BL 4.00
```

Per tenere traccia del numero di accessi di ogni veicolo, è obbligatorio utilizzare una lista collegata. Si suggerisce di memorizzare, in ogni elemento, la targa del veicolo e un contatore degli accessi effettuati, da incrementare ad ogni accesso del veicolo risultante dal file binario.

## Esercizio - Parte 2 (max 3 punti)

Estendere il programma sviluppato nella parte 1 in modo che crei anche un file di testo di nome `ultimi3.txt` contenente le tre targhe distinte che compaiono per ultime nel file binario, una per riga e in ordine inverso rispetto al file.

## Ulteriori specifiche

- La lista collegata deve essere implementata come tipo di dato astratto (in modo cioè che il programma principale acceda alla lista solo attraverso le funzioni definite nell'interfaccia della lista).
- Verificare la correttezza della linea di comando e la corretta apertura dei file; in caso di errore, stampare un messaggio e terminare l'esecuzione.
- Il programma deve essere costituito dai seguenti file:
  - `main.c` contenente (tra eventuali altre) la funzione `main`;
  - `listaVeicoli.c` con la definizione delle funzioni su liste (ed eventuali altre);

- `listaVeicoli.h` con le definizioni dei tipi di dato e le dichiarazioni delle funzioni definite in `listaVeicoli.c` e utilizzate in `main.c`;
- `Makefile` che permetta di costruire l'eseguibile con un singolo comando `make`.