

Un sistema per la gestione delle garanzie di autoveicoli può essere realizzato utilizzando le strutture seguenti:

```
struct garanzia
{
    char cognome [LUNG];
    int numTel;
    int mese;
};
struct gestGar
{
    garanzia gar[NUM];
    int numGar;
};
```

con **NUM** il numero massimo di garanzie contenute nel sistema e **numGar** il numero di garanzie effettivamente presenti nel sistema. Ogni pratica di garanzia è rappresentata con il cognome (al massimo di **LUNG** caratteri) del proprietario dell'autoveicolo, il numero di telaio dell'autoveicolo ed il mese di scadenza della garanzia. Quando è creata un'istanza di tipo **gestGar**, il valore di **numGar** è posto a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(const char nome[], gestGar\* g)** che, apre in lettura il file il cui nome è passato come primo argomento alla funzione, legge un numero intero **num** dal file e ripete **num** volte le azioni seguenti: a) legge una stringa e due numeri interi che rappresentano, rispettivamente, il cognome del proprietario dell'autoveicolo, il numero di telaio dell'autoveicolo ed il mese di scadenza della garanzia, e b) li inserisce nel sistema di gestione. La funzione restituisce **true** se l'operazione di lettura dal file non genera errori; altrimenti restituisce **false**;
2. **void ordina(gestGar\* g)** che ordina le garanzie per valori crescenti del campo **mese** e, all'interno dello stesso mese, per valori alfabeticamente crescenti del campo **cognome**;
3. **bool fineGar(gestGar\* g, int mese)** che elimina dal sistema le garanzie che scadono nel mese indicato come secondo argomento e compatta il vettore; la funzione restituisce **false** se non ci sono garanzie che scadono nel mese passato come secondo argomento, altrimenti restituisce **true**; **N.B.:** si supponga che le garanzie siano già state ordinate dalla funzione **ordina()**;
4. **int numCog(const gestGar\* g, char c)** che restituisce il numero di garanzie in cui il cognome del proprietario dell'autoveicolo termina con il carattere passato come secondo argomento alla funzione;
5. **bool estendi(gestGar\* g, int telaio)** che estende la garanzia del veicolo il cui numero di telaio è passato come secondo argomento alla funzione; la garanzia viene estesa ponendo 12 come mese di scadenza. La funzione deve provvedere a riordinare il vettore senza utilizzare la funzione **ordina()**. La funzione restituisce **true** se viene trovata la pratica relativa al numero di telaio passato come secondo argomento; altrimenti restituisce **false**.

### **NOTE SULLO SVOLGIMENTO DELLA PROVA:**

- PER SVOLGERE L'ELABORATO, APRIRE il Dev-C++ (dal Menù **Avvio** (o **Start**) nella barra degli strumenti in fondo allo schermo, selezionare Programmi e quindi Dev-C++);
- PRIMA DI INIZIARE LO SVOLGIMENTO DELL'ELABORATO, selezionare la voce **Identifica studente** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e inserire i dati richiesti;
- Per svolgere l'elaborato, aprire il progetto *esaInf.dev* presente nel directory *c:\esame\esaInf* e scrivere le funzioni richieste nel file *compito.cpp*, già presente nel progetto;
- **per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina);**
- SE L'ELABORATO È STATO COMPILATO SENZA ERRORI, PRIMA DELLA CONSEGNA, selezionare la voce **Consegna Compito** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e premere il tasto INVIO fino a quando non sparisce la finestra che è stata attivata.