

Un sistema di gestione di scorte farmaceutiche può essere realizzato mediante le strutture seguenti:

```
struct Farmaco
{
    char cat;
    char nomeComm[MAXL];
    char principio[MAXL];
    float costo;
    int numConf;
};

struct ScorteFarm
{
    Farmaco sco[MAXNUM];
    int numFar;
};
```

con **MAXNUM** il numero massimo di farmaci che possono essere gestiti e **numFar** il numero di farmaci effettivamente presenti. Ogni farmaco è individuato da un carattere **cat** che identifica la categoria del farmaco, da due stringhe **nomeComm** e **principio** (al massimo di **MAXL** caratteri) indicanti rispettivamente il nome commerciale e il principio attivo, dal float **costo** che memorizza il costo del farmaco e da un numero intero **numConf** che rappresenta il numero di confezioni ancora disponibili. Quando è creata un'istanza della struttura **ScorteFarm**, il valore di **numFar** è inizializzato a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **void build(ScorteFarm* s, const char nome[])** che apre in lettura il file il cui nome è passato come secondo argomento alla funzione e ripete **NUM** volte le seguenti azioni (con **NUM** costante predefinita): a) legge in sequenza un carattere, due stringhe, un float, ed un numero intero che rappresentano rispettivamente la categoria, il nome commerciale, il principio attivo, il costo e il numero di confezioni di un farmaco; b) inserisce il farmaco nel sistema di gestione delle scorte.
2. **void ordina(ScorteFarm* s)** che ordina i farmaci per valori alfabeticamente crescenti del campo **cat** e, per farmaci con lo stesso valore del campo **cat**, per valori alfabeticamente crescenti del campo **nomeComm**.
3. **int cateCosto(const ScorteFarm* s, Farmaco fc[], char c, float cost)** che inserisce nell'array **fc[]**, di dimensione massima **MAXNUM**, tutti i farmaci che appartengono alla categoria **c** e hanno un costo minore o uguale a **cost**. La funzione restituisce il numero di tali farmaci.
4. **float acquisto(ScorteFarm* s, const char nomeC[])** nel caso in cui il farmaco di nome commerciale **nomeC** sia presente nel sistema di gestione delle scorte, decrementa il numero **numConf** di confezioni del farmaco e, se non ci sono più confezioni disponibili, elimina il farmaco dal sistema di gestione mantenendo l'array **sco** compatto e ordinato; la funzione restituisce il costo del farmaco se presente nel sistema di gestione, altrimenti restituisce zero.
5. **float costoMedio(const ScorteFarm* a, char p)** che restituisce il costo medio dei farmaci il cui principio attivo comincia con il carattere specificato da **p**, oppure di tutti i farmaci presenti nel sistema di gestione delle scorte se **p** è uguale al carattere *****. Se nessun farmaco ha il principio attivo che inizia con il carattere specificato da **p**, la funzione restituisce **-1**;

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.