

Un sistema di gestione dei pacchi postali può essere realizzato mediante le seguenti strutture:

```
struct pacco
{
    int codice;
    char nomeDest[MAXL];
    char indirizzo[MAXL];
    float peso;
    long dataCons;
};

struct magazzino
{
    pacco mag[MAXNUM];
    int numPacchi;
};
```

con **MAXNUM** il numero massimo di pacchi che possono essere contenuti nel magazzino e **numPacchi** il numero di pacchi effettivamente presenti. Ogni pacco è individuato da un numero intero **codice** che identifica univocamente il pacco, ed è caratterizzato da due stringhe **nomeDest** e **indirizzo** (al massimo di **MAXL** caratteri) indicanti rispettivamente il nome e l'indirizzo del destinatario del pacco, dal float **peso** che memorizza il peso del pacco e da un numero intero **long dataCons** che rappresenta la data entro la quale il pacco deve essere consegnato. La data è memorizzata scrivendo in sequenza, senza interporre spazi, l'anno su quattro cifre, il mese ed il giorno su due cifre (ad esempio, la data 01/05/2004 viene memorizzata come 20040501). Quando è creata un'istanza della struttura **magazzino**, il valore di **numPacchi** è inizializzato a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **void build(magazzino\* m, const char nome[])** che apre in lettura il file il cui nome è passato come secondo argomento alla funzione e ripete le azioni seguenti fino a raggiungere la fine del file: a) legge in sequenza un numero intero, due stringhe, un float ed un numero intero che rappresentano rispettivamente il codice del pacco, il nome e l'indirizzo del destinatario, il peso del pacco e la data di consegna, e b) li inserisce nel magazzino.
2. **void ordina(magazzino\* m)** che ordina i pacchi in magazzino per valori crescenti del campo **dataCons** e, per pacchi con la stessa data, per valori alfabeticamente crescenti del campo **nomeDest**.
3. **bool consegna(const magazzino\* m, int data, pacco spe[], int\* num)** che inserisce nell'array **spe[]**, di dimensione massima **MAXNUM**, tutti i pacchi da consegnare entro la data **data**, restituendo attraverso il parametro **num** il numero di tali pacchi. La funzione restituisce **false** se non ci sono pacchi da spedire, **true** altrimenti.
4. **bool elimina(magazzino\* m, int codice)** che elimina dal magazzino il pacco il cui codice è passato come secondo argomento alla funzione mantenendo compatto ed ordinato il vettore **mag**; se il pacco non è presente nel magazzino viene restituito **false**, altrimenti viene restituito **true**.
5. **void stampacosto(const magazzino\* m)** che stampa in ordine crescente del campo **codice** il codice del pacco ed il costo del pacco calcolato come **peso\*COSTO**, dove **COSTO** è una costante definita nel file **compito.cpp**.

### **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.