

Il contenuto di uno scaffale di una libreria può essere rappresentato attraverso le seguenti costanti e strutture:

```
const int R = 10;
const int C = 20;
const int MAXL = 50;
```

```
struct Libro {
    char titolo[MAXL];
    char autore[MAXL];
    int pag;
    float pr;
    bool occupato;
};
```

```
struct Scaffale {
    Libro libro[R][C];
};
```

La costante **R** indica il numero di ripiani che compongono lo scaffale, mentre **C** indica il numero massimo di libri che possono essere contenuti in un ripiano. La struttura **Scaffale** si compone di un campo **libro** di tipo matrice con **R** righe e **C** colonne. Ogni elemento di **libro** identifica una posizione sullo scaffale. Se il campo **occupato** vale *true*, la posizione è occupata da un libro e in tal caso tutti gli altri campi dell'oggetto **Libro** ne contengono i dettagli. Se invece il campo **occupato** vale *false*, allora la posizione in questione è libera e gli altri campi dell'oggetto **Libro** non sono significativi. Pertanto, i campi **titolo**, **autore**, **pag**, e **pr** sono significativi solo se **occupato** vale *true* e contengono rispettivamente il titolo, l'autore, il numero delle pagine e il prezzo di un libro. Alcuni ripiani possono essere vuoti (non contengono alcun libro). Se un ripiano contiene dei libri allora questi si trovano sicuramente nelle posizioni con i più bassi numeri di colonna.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool salva(const Scaffale* ps, const char nf[], int nr)** salva nel file avente nome **nf** i dati dei libri contenuti nel ripiano **nr**. I dati relativi ai vari libri vengono salvati su righe separate, con un formato quale il seguente:
Furore, J._Steinbeck, 123, 8.5
Delitto_e_Castigo, F._Dostojevski, 560, 12.4
...
2. **int quantiLibri(const Scaffale* ps, const char nome[])** restituisce il numero di libri contenuti nello scaffale e aventi come autore **nome**.
3. **void piuPagine(const Scaffale* ps, int* pr, int* pc)** restituisce, attraverso **pr** e **pc**, il numero del ripiano e la posizione del libro che ha maggior numero di pagine tra quelli presenti nello scaffale (l'oggetto puntato da **pr** deve essere riempito con il numero del

ripiano e quello puntato da **pc** con la posizione all'interno del ripiano). Se lo scaffale è vuoto la funzione assegna agli oggetti puntati da **pr** e **pc** i valori *-1* e *-1*.

4. **int ordPrezzo(const Scaffale* ps, Libro al[], int r)** riempie l'array **al** con i dati dei libri contenuti nel ripiano **r** ordinandoli per prezzo crescente. La funzione restituisce il numero di libri inseriti in **al**.

5. **bool elimina(Scaffale* ps, char parola[], int i)** elimina dal ripiano **i**-esimo il primo libro il cui titolo contiene la parola **parola**. La funzione compatta gli altri libri contenuti sul ripiano in modo che occupino le posizioni con i più bassi numeri di colonna. La funzione restituisce il valore *true* se un libro viene eliminato, *false* altrimenti. Suggerimento: la ricerca di una stringa all'interno di un'altra stringa può essere eseguita attraverso la funzione di libreria *strstr()*, dichiarata in *string.h*:

*char * strstr(const char * pagliaio, const char * ago)*

la funzione cerca la prima occorrenza della stringa indicata dall'argomento *ago* all'interno della stringa indicata dall'argomento *pagliaio*. La funzione restituisce il puntatore all'inizio della sottostringa nel caso in cui venga trovata, oppure il valore *NULL* se non viene trovata.

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.