

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 *APRIQUESTO.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).

Cantina

Un programma per la gestione di una cantina può essere realizzato usando le seguenti costanti e i seguenti tipi:

```
const int CODL = 8;
const int MAXS = 100;
const int MAXO = 50;
const int MAXV = 80;
```

```
struct Vino {
    char nome[MAXS];
    int anno;
    float quantita;
    float prezzo;
    char codice[CODL];
};
```

```
struct Magazzino {
    Vino vi[MAXV];
    int numVi;
};
```

```
struct Ordine {
    char codice[CODL];
    float quantita;
};
```

```
struct CodaOrd {
    Ordine ord[MAXO];
    int front;
    int back;
};
```

```
struct Cantina {
    Magazzino mag;
    CodaOrd coda;
};
```

Una cantina si compone di un magazzino, in cui sono memorizzati i dati dei vini disponibili, e di una coda, in cui sono memorizzati i dati degli ordini di vendita. La struttura `Vino` memorizza: il nome del vino (una stringa), l'anno di produzione (un intero), la quantità di tale vino presente in magazzino in litri (un float), il prezzo al litro (un float), il codice univoco di tale tipo di vino (una stringa). La struttura `Magazzino` memorizza i dati dei vini presenti in magazzino nell'array `vi` (al più `MAXV` elementi), mentre il campo `numVi` indica il numero di diversi tipi di vino effettivamente presenti. La struttura `Ordine` memorizza il codice del vino coinvolto nell'ordine di vendita e la quantità che deve essere venduta. Gli ordini sono memorizzati in una coda definita dal tipo `CodaOrd` (in grado di contenere al più `MAXO-1` ordini).

Scrivere il corpo delle seguenti funzioni C++.

1. **`void calcolaCodice(const char n[], int a, char cod[])`** che riempie l'array `cod` con il codice del vino avente nome `n` e anno `a`. Il codice di un vino è una stringa di 7 caratteri (più il carattere nullo) così composta:
 - i primi 5 caratteri sono le consonanti del nome; se nel nome ci sono meno di 5 consonanti, i caratteri mancanti vengono posti uguali a 'x'.
 - gli ultimi due caratteri corrispondono alle due cifre meno significative dell'anno di produzione.Per esempio, il codice di un brunello del 1994 sarà *brnll94*, il codice di un chianti del 2006 sarà *chntx06*.
2. **`int caricaVino(Cantina* pc, const char nf[])`** che riempie il magazzino della cantina puntata da `pc` con i dati dei vini contenuti nel file di nome `nf`. Nel file, i dati dei vini sono memorizzati su righe separate. Ogni riga contiene nell'ordine: il nome del vino (come una singola parola), l'anno di produzione, la quantità di tale vino presente in magazzino e il suo prezzo al litro. Il codice di ogni vino che viene inserito nel magazzino deve essere calcolato usando la funzione *calcolaCodice()* indicata al punto 1. Al chiamante viene restituito un intero che indica il numero di vini letti dal file e inseriti nel magazzino.
3. **`bool nuovoOrdine(Cantina* pc, Ordine o)`** che, se possibile, inserisce un nuovo ordine nella coda degli ordini della cantina puntata da `pc`. Affinché sia possibile inserire l'ordine `o` devono valere entrambe le seguenti condizioni: *i)* nel magazzino è presente un vino avente lo stesso codice specificato nell'ordine `o`; *ii)* la coda non è piena. Se l'ordine viene inserito in coda la funzione restituisce *true*, altrimenti restituisce *false*.
4. **`float evadi(Cantina* pc)`** che evade il maggior numero possibile di ordini presenti in coda, partendo dalla testa e fermandosi quando viene incontrato un ordine che non può essere evaso. Affinché un ordine possa essere evaso, in magazzino deve essere presente del vino avente lo stesso codice dell'ordine e la quantità di tale vino presente in magazzino deve essere maggiore o uguale alla quantità specificata nell'ordine. Quando un ordine viene evaso viene rimosso dalla testa della coda e il vino presente in magazzino viene ridotto della quantità specificata nell'ordine. L'evasione degli ordini si ferma non appena si incontra un ordine che non può essere evaso. La funzione restituisce il valore cumulativo di tutti gli ordini evasi, dove il valore del singolo ordine è pari alla quantità di vino specificata nell'ordine moltiplicata per il prezzo al litro del vino in questione.
5. **`void stampa(const Cantina* pc)`** che stampa a video i dati dei vini presenti nel magazzino e i dati degli ordini in coda. I vini sono stampati in ordine di anno crescente. Produrre a video un'uscita formattata come il seguente esempio:

```
##### Magazzino #####
Nome: brunello
Anno: 1994
Quantita: 123.5
```

Prezzo: 66
Codice: brnll94

Nome: amarone
Anno: 1999
Quantita: 45.7
Prezzo: 85.6
Codice: mrnxx99

Nome: chianti
Anno: 2006
Quantita: 260
Prezzo: 5.2
Codice: chntx06

Nome: firriato
Anno: 2008
Quantita: 184
Prezzo: 5.5
Codice: frrtx08

Nome: primitivo
Anno: 2009
Quantita: 350
Prezzo: 3.5
Codice: prmtv09

Coda #####
Codice: chntx06
Quantita: 25

Codice: mrnxx99
Quantita: 12