

Un sistema per l'iscrizione alla prova orale di un esame può essere rappresentato come una coda mediante le seguenti costanti e strutture:

```
const int MAXL = 100;
const int MAXN = 50;

struct Iscrizione {
    char nome[MAXL];
    int matricola;
    int votoScritto;
};

struct CodaIscr {
    int front, back;
    Iscrizione iscr[MAXN];
};
```

Per ogni iscrizione vengono memorizzati il nome dello studente (una stringa di al più **MAXL** caratteri), il suo numero di matricola e il voto ottenuto nella prova scritta. Una coda di iscrizioni è in grado di gestire i dati di al più **MAXN-1** iscrizioni (memorizzandoli all'interno del campo **iscr**). I campi **front** e **back** indicano rispettivamente le posizioni da usare per le operazioni di estrazione e inserimento. Quando viene creata una istanza di **CodaIscr** i suoi campi **front** e **back** sono impostati a 0.

Scrivere il corpo delle seguenti funzioni C++.

1. **int inserisci(CodaIscr\* pc, char n[], int m, int v)** inserisce una nuova iscrizione nella coda; i campi **n**, **m** e **v** indicano rispettivamente il nome, il numero di matricola e il voto dell'iscrizione da inserire. La funzione restituisce un intero che indica la posizione nella coda dell'iscrizione appena inserita (1 per la prima, 2 per la seconda, e così via). Nel caso in cui non sia possibile inserire una nuova iscrizione (coda piena), la funzione restituisce il valore -1.
2. **bool scambia(CodaIscr\* pc, int a, int b)** scambia le posizioni delle iscrizioni relative agli studenti con numeri di matricola **a** e **b**. Se lo scambio ha successo la funzione restituisce il valore *true*, altrimenti restituisce il valore *false* (per esempio se non esiste un'iscrizione relativa ad uno dei due o ad entrambi i numeri di matricola).
3. **bool salva(const CodaIscr \*pc, const char nf[])** salva il contenuto della coda nel file di nome **nf**. I dati delle iscrizioni vengono salvati nel seguente formato:  
*Giuseppe Greco, 432567, 23*  
*Sabrina Monti, 459975, 28*  
...  
La funzione restituisce il valore *false* nel caso in cui l'operazione non vada a buon fine, *true* altrimenti.
4. **bool controlla(const CodaIscr\* pc)** controlla se nella coda sono contenute due iscrizioni aventi lo stesso numero di matricola ed in tal caso restituisce il valore *false*. Se invece non ci sono due iscrizioni con lo stesso numero di matricola restituisce *true*.
5. **bool votoPiuAlto(const CodaIscr\* pc, Iscrizione\* pi)** trova l'iscrizione contenuta nella coda che ha il voto più alto. Tale iscrizione viene restituita al chiamante attraverso

l'oggetto puntato da **pi**. La funzione restituisce *true* se c'è almeno una iscrizione in coda, *false* se la coda è vuota.

### 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.