

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).

Coda messaggi

Una coda di messaggi di posta elettronica può essere rappresentata attraverso le seguenti costanti e i seguenti tipi:

```
const int MAXN = 20;
const int MAXS = 100;

enum Prior {high, medium, low};

struct Data {
    int giorno;
    int mese;
    int anno;
};

struct Messaggio {
    Prior pri;
    char soggetto[MAXS];
    Data invio;
    char destinatario[MAXS];
    char testo[MAXS];
};

struct CodaMes{
    Messaggio mes[MAXN];
    int front;
    int back;
};
```

La struttura *Messaggio* memorizza: il soggetto del messaggio (come una stringa), il destinatario (come una stringa), il testo (come una stringa), la data di invio (di tipo *Data*) e la priorità (di tipo *Prior*). La struttura *CodaMes* rappresenta una coda di messaggi.

Scrivere il corpo delle seguenti funzioni C++.

1. **int quanti(const CodaMes* pc, Data d, Data a)** che restituisce il numero di messaggi aventi una data di invio compresa tra d e a (successiva o uguale a d e precedente o uguale ad a).
2. **bool piuLungo(const CodaMes* pc, Messaggio* pm)** che, se possibile, riempie l'oggetto puntato da pm con il messaggio avente il testo più lungo. La funzione restituisce *true* nel caso in cui ci sia almeno un messaggio, altrimenti restituisce *false*.
3. **bool salvaOrd(const CodaMes* pc, const char filen[])** che salva nel file di nome *filen* i dati dei messaggi ordinandoli per valori alfabeticamente crescenti del campo soggetto, come illustrato dal seguente esempio:

```
-----  
Soggetto: appunti  
Dest.: giuseppe@yahoo.com  
Prior.: medium  
Data: 16/6/2011  
Testo: Giuseppe puoi portarmi gli appunti?  
-----  
Soggetto: ciao  
Dest.: mario@abcdef.com  
Prior.: medium  
Data: 21/6/2011  
Testo: Ciao Mario come va?  
-----  
Soggetto: riunione  
Dest.: alfredo@gmail.com  
Prior.: high  
Data: 18/6/2011  
Testo: domani riunione importante!
```

La funzione restituisce *true* se il salvataggio va a buon fine, *false* altrimenti.

4. **int rimuovi(CodaMes* pc, const char dest[])** che elimina dalla coda tutti i messaggi che hanno un campo destinatario uguale a *dest*. La funzione restituisce il numero di messaggi rimossi.
5. **int priorita(const CodaMes* pc, Prior p, Messaggio v[])** che riempie l'array *v* con i messaggi contenuti nella coda e aventi priorità pari a *p*. La funzione restituisce un valore che indica quanti elementi di *v* sono stati effettivamente riempiti.