INFORMATICA E CALCOLATORI (ELETTRONICA) **Scritto 11 Febbraio 2004**

INFORMATICA (BIOMEDICA)

Un sistema per la gestione delle prenotazioni di un ristorante può essere realizzato utilizzando le strutture seguenti:

```
struct cliente
{    char cognome[LUNG]; int numPosti;
};
struct coda
{    int front; int back; cliente queue[DIM];
};
struct gestione
{    int numDisp; coda cc;
};
```

numDisp numero di posti a sedere disponibili nel ristorante (il numero massimo è **POSTI**). Ogni cliente è caratterizzato dal suo cognome e dal numero di posti che richiede. Se il numero di posti richiesti da un cliente è disponibile, allora **numDisp** viene decrementato del numero di posti richiesti dal cliente; altrimenti, il cliente viene inserito nella coda **cc**, che può contenere al massimo **DIM** clienti. Quando è creata un'istanza della struttura **gestione**, il valore di **numDisp** è posto uguale a **POSTI** e la coda è vuota.

Scrivere il corpo delle seguenti funzioni C++.

1. **void build(const char nome[], gestione* g)** che, apre in lettura il file il cui nome è passato come primo argomento alla funzione, e fino a raggiungere la fine del file oppure fino a riempire la coda: a) legge una stringa ed un numero intero che rappresentano, rispettivamente, il cognome del cliente ed il numero di posti richiesti dal cliente, b) se il numero di posti richiesti è minore o uguale a **numDisp**, sottrae il numero di posti a **numDisp;** altrimenti, inserisce il cliente in coda;
2. **bool nuovoSer(gestione* g)** che controlla se può essere servito un cliente presente in coda: un cliente può essere servito se il numero di posti richiesti dal cliente è minore o uguale a **numDisp**; il cliente servito è il primo nella coda che soddisfa la condizione; il cliente servito viene eliminato dalla coda che viene ricompattata, mantenendo l'ordine con cui sono state effettuate le prenotazioni; **numDisp** viene decrementato del numero di posti richiesti dal cliente servito; la funzione restituisce **false** se la coda è vuota oppure se nessun cliente può essere servito; altrimenti restituisce **true**;
3. **void stat(const gestione* g, float* media, float* varianza)** che restituisce, tramite i due ultimi argomenti, la media e la varianza dei campi **numPosti** dei cliente inseriti in coda;
4. **void visual(const gestione* g)** che stampa su uscita standard la coda nel modo seguente: di ogni cliente viene stampato il cognome ed un numero di asterischi pari al numero di posti richiesti; ogni cliente viene mostrato su una riga diversa;
5. **void salva(const char nome[],const gestione* g)** che salva sul file il cui nome è passato come primo argomento alla funzione i clienti contenuti nella coda a partire dal primo cliente inserito; per ogni cliente, viene salvato il cognome ed il numero di posti richiesti separati da uno spazio; ogni cliente occupa una riga diversa del file.

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.