

INFORMATICA (BIOMEDICA)

Una versione semplificata di un sistema di gestione delle analisi del sangue può essere realizzato utilizzando le strutture seguenti:

```
struct paziente
{
    char cognome[LUN];
    int annoNascita;    //su quattro cifre
    char sesso;        //'F' per femmina, 'M' per maschio
    double hgb;        //valore emoglobina
    double rbc;        //valore globuli rossi
    double wbc;        //valore globuli bianchi
};
struct gestAn
{
    paziente p[NUM];
    int numPaz;
};
```

con **NUM** il numero massimo di pazienti gestiti dal sistema e **numPaz** il numero di pazienti effettivamente presenti nel sistema. Ogni paziente è rappresentato con il suo cognome (al massimo di **LUNG** caratteri), l'anno di nascita ed il suo sesso. Quando è creata un'istanza di tipo **gestAn**, il valore di **numPaz** è posto a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(const char nome[], gestAn* g)** che, apre in lettura il file il cui nome è passato come primo argomento alla funzione, legge un numero intero **num** dal file e ripete **num** volte le azioni seguenti: a) legge una stringa, un intero, un carattere, e tre numeri double che rappresentano rispettivamente il cognome, l'anno di nascita, il sesso ed i valori di emoglobina, globuli rossi e globuli bianchi, e b) li inserisce nel sistema di gestione. La funzione restituisce **true** se l'operazione di lettura dal file non genera errori; altrimenti restituisce **false**;
2. **void ordina(gestAn* g)** che ordina i pazienti per valori crescenti del campo **annoNascita** e, all'interno dello stesso anno di nascita, per valori alfabeticamente crescenti del campo **cognome**;
3. **void media(const gestAn* g, char sex, double med[])** che restituisce in **med[0]**, **med[1]** e **med[2]** la media dei valori di, rispettivamente, emoglobina, globuli rossi e globuli bianchi dei pazienti di sesso **sex**;
4. **void check(const paziente* paz, bool ris[3])** che restituisce in **ris[0]**, **ris[1]** e **ris[2]**, il valore **false** se i valori di, rispettivamente, emoglobina, globuli rossi e globuli bianchi dell'oggetto paziente puntato da **paz** non rientrano negli intervalli fissati nella tabella riportata nella pagina seguente; altrimenti restituisce **true** (all'inizio del file *compito.cpp* i valori nella tabella sono riportati sotto forma di costanti intere);
5. **void stampa(const gestAn* g)** che stampa il cognome, il sesso, l'anno di nascita ed i valori di emoglobina, globuli rossi e globuli bianchi di ogni paziente (ogni paziente occupa una riga); se uno dei valori non rientra nell'intervallo fissato nella tabella riportata nella pagina seguente, il valore viene fatto seguire da un asterisco (per esempio, **Verdi 1970 F 11* 5 9000***);

	Femmine	Maschi
Emoglobina (g/l)	12-16	14-18
Globuli rossi (milioni/mm ³)	4-5.5	4.5-6
Globuli bianchi (unità/mm ³)	4000-7000	5000-8000

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.