

Un archivio di donatori di sangue può essere realizzato utilizzando le strutture seguenti:

```
struct donatore
{
    char nome [LUNG];
    char gruppo[3];
    int giorniDonaz;
};
struct archivio
{
    donatore don[NUM];
    int numDon;
};
```

con **NUM** il numero massimo di donatori contenuti nell'archivio e **numDon** il numero di donatori effettivamente presenti nell'archivio. Ogni donatore è rappresentato con il suo nome (al massimo di **LUNG** caratteri), il suo gruppo sanguigno ed il numero di giorni trascorsi dall'ultima donazione. Quando è creata un'istanza di tipo **archivio**, il valore di **numDon** è posto a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **void build(const char nome[], archivio* ar)** che, apre in lettura il file il cui nome è passato come primo argomento alla funzione e ripete le azioni seguenti fino a raggiungere la fine del file: a) legge due stringhe ed un numero intero che rappresentano, rispettivamente, il nome ed il gruppo sanguigno del donatore, e i giorni trascorsi dall'ultima donazione, e b) li inserisce nell'archivio.
2. **void ordina(archivio* ar)** che ordina i donatori per valori alfabeticamente crescenti del campo **gruppo** e, all'interno dello stesso gruppo, per valori decrescenti del campo **giorniDonaz**;
3. **bool donato(archivio* ar, const char gruppo [])** che aggiorna l'archivio dopo una donazione effettuata dal donatore con il valore più alto del campo **giorniDonaz** tra quelli che hanno il gruppo sanguigno uguale a quello passato come argomento. La funzione trova il donatore, azzerà il suo campo **giorniDonaz** e sposta il donatore nell'ultima posizione tra i donatori con lo stesso gruppo sanguigno mantenendo il vettore **don** compatto; la funzione restituisce **false** se non ci sono donatori appartenenti al gruppo sanguigno, altrimenti restituisce **true**; nel realizzare la funzione si consideri l'archivio già ordinato dalla funzione **ordina()**; N.B.: la funzione deve essere realizzata senza usare la funzione *ordina()*;
4. **int donPerGruppo(archivio* ar, const char gruppo[], int num)** che restituisce il numero di donatori appartenenti al gruppo passato come argomento alla funzione che hanno donato il sangue da un numero di giorni superiore o uguale al terzo argomento passato alla funzione; nel realizzare la funzione si consideri l'archivio già ordinato dalla funzione **ordina()**;
5. **bool cerca(archivio* ar, const char nome[], char gruppo[], int* numGiorni)** che cerca il donatore il cui nome è passato come secondo argomento alla funzione e restituisce il suo gruppo sanguigno ed il numero di giorni trascorsi dall'ultima donazione utilizzando gli ultimi due argomenti della funzione, rispettivamente. La funzione restituisce **true** se viene trovato il donatore; altrimenti restituisce **false**.

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.