

Un pronto soccorso possiede **NUM** ambulatori. Gli ambulatori possono essere aperti o chiusi. Un ambulatorio aperto può essere libero o occupato. Quando un paziente arriva al pronto soccorso viene identificato con un colore a seconda della gravità del problema: rosso, giallo o bianco (in ordine di gravità decrescente). Se uno degli ambulatori è libero il paziente accede direttamente, altrimenti viene inserito nella coda corrispondente al suo colore. Il pronto soccorso può essere rappresentato con le strutture seguenti:

```
enum {ROSSO, GIALLO, BIANCO} colore;

struct ambulatorio {
    bool aperto;           //true -> aperto,
                          //false -> chiuso
    bool occupato;       //true -> occupato,
                          //false -> libero
};

struct paziente {
    char nome[LUN];
    char cognome[LUN];
    int anni;
}

struct coda {
    int front, back;
    paziente queue[DIM];
}

struct prontosoccorso {
    ambulatorio amb[NUM];
    coda c[3];
}
```

La struttura **coda** rappresenta una coda di pazienti (realizzata mediante un array di dimensione **DIM** e quindi in grado di gestire **DIM-1** pazienti). Di ogni paziente vengono memorizzati l'età, il nome e il cognome (al massimo di **LUN** caratteri). Quando è creata un'istanza di tipo **prontosoccorso**, tutti gli ambulatori sono aperti e liberi. Scrivere il corpo delle seguenti funzioni C++:

1. **bool arrivo(prontosoccorso* p, const char nome[], const char cognome[], int anni, colore col)** che implementa l'arrivo del paziente i cui dati sono passati come argomenti alla funzione; il quinto argomento indica la gravità del paziente; se uno degli ambulatori aperti è libero, il paziente viene inviato direttamente all'ambulatorio (che diventerà quindi occupato), altrimenti il paziente viene inserito nella coda corrispondente al suo colore; se la coda è piena, la funzione restituisce **false**; altrimenti restituisce **true**;
2. **void conta(const prontosoccorso* p, int numeropaz[])** che restituisce attraverso il vettore **numeropaz[]** passato come argomento alla funzione il numero di pazienti presenti nelle tre code; in particolare, gli elementi del vettore ad indice 0, 1 e 2 conterranno, rispettivamente, i numeri di pazienti con codice rosso, giallo e bianco;
3. **bool chiudi(prontosoccorso* b, int numamb, int numero)** che chiude l'ambulatorio indicato da **numamb** se sono verificate entrambe le condizioni seguenti: i) la coda rossa è vuota, ii) la somma dei pazienti nelle altre code è minore di **numero**. La funzione restituisce **true** se l'ambulatorio viene chiuso (o era già chiuso), **false** altrimenti;
4. **void stampa(const prontosoccorso* p, colore col)** che stampa nome, cognome ed età dei pazienti presenti nella coda di colore **col**, ordinati alfabeticamente per cognome;

5. `void dimetti(prontosoccorso* p, int numamb)` che dimette il paziente dell'ambulatorio `numamb`; se almeno una delle code non è vuota, il paziente da più tempo nella coda a più alta priorità occupa l'ambulatorio; altrimenti, se tutte le code sono vuote, l'ambulatorio diventa libero.

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.