

**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++);
- 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).

**Gestione visite**

Un programma per la gestione di visite mediche e di un elenco di pazienti può essere realizzato a partire dalle seguenti costanti e dai seguenti tipi:

```
const int MAXS = 100;
const int MAXV = 100;
const int MAXP = 100;

enum TipoVisita {ECOGRAFIA, TAC, RISONANZA};

struct Paziente {
    char nome[MAXS];
    char cognome[MAXS];
    int identif;
};

struct ElencoPazienti {
    Paziente paz[MAXP];
    int numPaz;
};

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

struct Visita {
    Data data;
    int idPaz;
    TipoVisita tipVis;
};

struct Calendario {
    Visita vis[MAXV];
    int numVis;
};
```

La struttura `Data` serve a memorizzare giorno, mese e anno di una data. La struttura `Paziente` contiene il nome e il cognome di un paziente e un identificatore univoco assegnato al paziente (codificato come un intero). Il tipo enumerazione `TipoVisita` indica i possibili tipi di visita. La struttura `Visita` contiene la data in cui deve essere eseguita una visita, l'identificatore del paziente che deve eseguire la visita e il tipo di visita da eseguire. La struttura `ElencoPazienti` contiene i dati dei pazienti, all'interno dell'array `paz`, e il loro numero effettivo, indicato da `numPaz`. La struttura `Calendario` contiene i dati delle visite in calendario, all'interno dell'array `vis`, e il numero effettivo di visite presenti, indicato da `numVis`.

Scrivere il corpo delle seguenti funzioni C++.

1. **`bool caricaElenco(ElencoPazienti* pe, const char n[])`** che riempie l'elenco dei pazienti puntato da `pe` con i dati contenuti nel file di nome `n`. Nel file i dati dei pazienti sono memorizzati su righe separate. In ogni riga sono contenuti, nell'ordine, una prima stringa che indica il nome del paziente, una seconda stringa che indica il cognome del paziente, un intero che indica l'identificatore univoco di tale paziente. I valori sono separati da spazi. La funzione controlla che non vi siano due o più pazienti con lo stesso identificatore. La funzione restituisce *true* se l'operazione di caricamento va a buon fine, *false* se si verifica un errore in caricamento o se ci sono due o più pazienti con lo stesso identificatore.
2. **`int nuovaVisita(const ElencoPazienti* pe, Calendario* pc, Visita v)`** che inserisce nel calendario puntato da `pc` la visita `v`. La nuova visita viene inserita nel calendario supponendo che le visite presenti nel calendario siano già ordinate per data crescente e lasciando il calendario ancora ordinato. Affinché la visita `v` possa essere inserita, deve essere presente nell'elenco dei pazienti puntato da `pe` un paziente con identificatore univoco uguale a quello contenuto nella visita `v`. La funzione restituisce l'indice della posizione occupata dalla visita appena inserita nel calendario (se l'operazione va a buon fine), -1 altrimenti.
3. **`bool primoPaziente(const ElencoPazienti* pe, const Calendario* pc, TipoVisita tv, Paziente* pp)`** che cerca nel calendario la prima visita di tipo `tv` e riempie l'oggetto puntato da `pp` con i dati del paziente che la deve eseguire. La funzione restituisce *true* se nel calendario è presente almeno una visita di tipo `tv`, altrimenti restituisce *false*.
4. **`int trovaVisite(const Calendario* pc, int id, Visita vv[])`** che riempie l'array `vv` con tutte le visite contenute nel calendario che sono relative al paziente identificato da `id`. La funzione restituisce il numero di visite inserite in `vv`.
5. **`void stampaReport(const ElencoPazienti* pe, const Calendario* pc, int id)`** che stampa a video un report relativo al paziente identificato da `id`. Il report contiene: nome e cognome del paziente in questione, e tutte le informazioni relative alle sue visite presenti nel calendario. Il report deve avere un formato come il seguente:  
#####  
Mario Rossi  
-----  
12/3/2011 TAC  
18/4/2011 RIS  
1/6/2011 RIS  
3/9/2011 ECO  
#####