

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).

Dieta

La dieta settimanale di un individuo può essere rappresentata attraverso le seguenti costanti e i seguenti tipi:

```
const int MAXN = 100;
const int MAXP = 10;

enum Giorno {Lun, Mar, Mer, Gio, Ven, Sab, Dom};

enum Tipo {Colazione, Pranzo, Cena};

struct Alimento {
    char nome[MAXN];
    int calorie;
    int grammi;
};

struct Pasto {
    Alimento alim[MAXP];
    int numAlim;
};

typedef Pasto Dieta[3][7];
```

Il tipo `Giorno` rappresenta i giorni della settimana, mentre `Tipo` indica il tipo di un pasto (colazione, pranzo, o cena). La struttura `Alimento` contiene il nome dell'alimento, le calorie apportate e la quantità espressa in grammi. Un pasto è composto da un certo numero di alimenti. Gli alimenti che compongono un pasto sono memorizzati nell'array `alim`, mentre il loro numero è indicato da `numAlim`. Il tipo `Dieta` è definito come una matrice di pasti avente 3 righe e 7 colonne. Le colonne corrispondono ai giorni della settimana (indice di colonna 0 corrisponde a lunedì, 1 a martedì, e così via), mentre le righe corrispondono ai tre pasti della giornata (indice di riga 0 corrisponde a colazione, 1 a pranzo e 2 a cena).

Scrivere il corpo delle seguenti funzioni C++.

1. **bool inserisci(Dieta d, Alimento p, Tipo t, Giorno g)** che, se possibile, inserisce l'alimento `p` nella dieta `d`. In particolare, l'alimento deve essere aggiunto al pasto del giorno `g` e di tipo `t`. L'inserimento non è possibile se l'array `alim` del pasto in questione è pieno. In tal caso viene restituito il valore *false* al chiamante. Nel caso in cui, invece, l'alimento venga inserito la funzione restituisce il valore *true*.
2. **Giorno piuCalorie(const Dieta d, Tipo t)** che restituisce il giorno della settimana in cui il pasto di tipo `t` ha più calorie.
3. **void stampa(const Dieta d, Giorno g)** che stampa a video l'elenco degli alimenti che compongono i pasti relativi al giorno `g`. La stampa deve avere il formato indicato dal seguente esempio:

```
latte 120 100
biscotti 200 60
-----
pasta_al_pomodoro 250 100
insalata 100 100
frutta 200 150
-----
patate_lesse 100 300
formaggio 200 130
-----
```

4. **int quanteVolte(const Dieta d, const char n[])** che restituisce il numero di presenze, nella dieta settimanale, dell'alimento avente nome `n`.
5. **bool salva(const Dieta d, const char nomef[])** che salva, nel file di nome `nomef`, una lista contenente tutti gli alimenti presenti nella dieta settimanale. Gli alimenti devono apparire nella lista una sola volta e per ognuno di essi deve essere riportata la quantità totale presente nella dieta settimanale. La lista deve essere ordinata per quantità totale decrescente. La funzione restituisce *true* nel caso in cui l'operazione vada a buon fine, *false* altrimenti. Esempio di parte di lista:

```
900 patate_lesse
750 frutta
600 latte
600 birra
400 minestrone
400 couscous
400 polenta
360 biscotti
...
```