

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).

Bookmark

Un sistema per la gestione dei bookmark può essere realizzato usando le seguenti costanti e i seguenti tipi:

```
const int MAXL = 100;
const int MAXN = 50;

enum Categoria {SPORT, NEWS, DIVERTIMENTO, STUDIO};

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

struct Bookmark {
    char url[MAXL];
    Categoria cat;
    int visite;
    Data data;
};

struct BMSistGest {
    Bookmark bm[MAXN];
    int numBM;
};
```

La struttura `Bookmark` memorizza: l'url del sito web (come una stringa), la sua categoria (di tipo enumerazione), il numero delle visite eseguite al sito (come un intero), e la data in cui il bookmark è stato aggiunto al sistema di gestione. La struttura `BMSistGest` è composta da un array `bm` che contiene i dati dei bookmark e da un intero `numBM` che indica il numero di bookmark effettivamente presenti.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool salva(const BMSistGest* pa, const char filename[])** Salva tutti i bookmark contenuti nel sistema di gestione indicato da *pa* nel file di nome *filename*. Nel file deve essere creata una riga per ogni bookmark e i bookmark devono essere raggruppati per categoria. Rispettare il formato indicato dal seguente esempio:

<i>www.ing.unipi.it</i>	<i>studio</i>	<i>0</i>	<i>23/1/2008</i>
<i>www.wikipedia.it</i>	<i>studio</i>	<i>2</i>	<i>16/11/2007</i>
<i>www.gazzetta.it</i>	<i>sport</i>	<i>3</i>	<i>11/7/2008</i>
<i>www.repubblica.it</i>	<i>news</i>	<i>2</i>	<i>9/4/2008</i>
<i>www.cnn.com</i>	<i>news</i>	<i>1</i>	<i>29/12/2008</i>

La funzione restituisce *true* se il salvataggio va a buon fine, *false* se si verifica un errore.

2. **bool visita(BMSistGest* pa, const char nomesito[])** Incrementa il numero delle visite relative al bookmark il cui url è indicato dal parametro *nomesito*. La funzione restituisce *true* se un bookmark con tale url è presente nel sistema di gestione, altrimenti restituisce *false*.
3. **int piuVisitati(const BMSistGest* pa, Bookmark v[])** Riempie l'array *v* con i tre bookmark più visitati tra quelli presenti nel sistema di gestione puntato da *pa*. La funzione restituisce il numero di elementi di *v* che sono stati effettivamente riempiti (potrebbero essere meno di tre nel caso in cui il sistema di gestione contenga meno di tre bookmark).
4. **int eliminaCat(BMSistGest* pa, Categoria c)** Elimina dal sistema di gestione puntato da *pa* tutti i bookmark che hanno categoria pari a *c* (la funzione provvede a compattare opportunamente l'array *bm*). La funzione restituisce il numero di bookmark eliminati.
5. **int quanti(const BMSistGest* pa, Data da, Data a)** Restituisce il numero di bookmark contenuti nel sistema di gestione puntato da *pa* che sono stati aggiunti nell'intervallo di tempo compreso tra *da* e *a*, estremi inclusi.