

Fondamenti di Informatica e Laboratorio T-AB  
Ingegneria dell'Automazione  
a.a. 2008/2009

# Lab 12

## Progetto su Piu' File e File

# Esercizio 1

---

- Sia data la struttura

```
struct time
{
    int hour, minute, second;
};
```

- Per semplicità si può definire il tipo Time

```
typedef struct time Time;
```

# Esercizio - 1

---

Si realizzi in un modulo **tempo.h/tempo.c** un insieme di funzioni per la gestione del tipo Time. In particolare:

- Si realizzi una funzione

```
Time leggiTime()
```

che legga da input ore, minuti e secondi, e restituisca una struttura di tipo Time opportunamente inizializzato coi valori letti

- Si realizzi una funzione

```
int leggiMoreTimes(Time v[], int dim)
```

La funzione deve leggere da input delle strutture Time (a tal scopo si utilizzi la funzione definita sopra) e salvarle nel vettore v, di dimensione fisica dim. La funzione deve restituire il numero di elementi letti. La lettura termina se l'utente inserisce un tempo con ora negativa.

# Esercizio - 1

---

- Si progetti una funzione in grado di calcolare la differenza fra due strutture **Time** e che restituisca il risultato in termini di una nuova struttura **Time**
- L'interfaccia della funzione è facilmente desumibile dalle specifiche:  

```
Time subtract(Time t1, Time t2);
```
- Due possibili approcci:
  1. Trasformare in secondi, eseguire la differenza, trasformare in ore, minuti, secondi
  2. Eseguire la sottrazione direttamente tenendo conto dei riporti

## Esercizio 2

---

- Realizzare un programma che, aperto un file di testo di nome “Prova.txt” in modalità “scrittura”, provveda a leggere da input delle parole separate da spazi (stringhe di al più 63 caratteri) e le scriva nel file di testo.
- Il programma termina quando l’utente inserisce la parola “fine”. Si abbia cura di chiudere il file prima di terminare definitivamente il programma.
- Si controlli il corretto funzionamento del programma accedendo direttamente al file di testo con un editor (ad es. Notepad).

## Esercizio 3

---

Sia dato il file di testo "dati.txt" contenente i dati relativi agli studenti immatricolati al primo anno della Facoltà di Ingegneria

In particolare, le informazioni sono memorizzate nel file "dati.txt" come segue: ognuna delle linee del file contiene i dati relativi ad un nuovo studente; in particolare:

- 1 Matricola: un intero che indica il numero di matricola dello studente
- 2 CdL: un intero che indica il corso di laurea (CdL) dello studente (es. 2145)

# Esercizio 3

---

Sia dato un secondo file di testo, “indirizzi.txt” che contiene, invece, l’indirizzo di ogni studente, e in particolare, in ogni linea, separati da uno spazio:

- **Matricola**: il numero di matricola dello studente (un intero)
- **Nome**: il nome dello studente, al più 20 caratteri senza spazi
- **Cognome**: il cognome dello studente, al più 30 caratteri senza spazi
- **Via**: una stringa di al più 30 caratteri senza spazi, che riporta la via di residenza dello studente
- **Città**: una stringa che riporta la città di residenza dello studente, al più 30 caratteri senza spazi
- **CAP**: un intero che rappresenta il codice di avviamento postale dello studente

## Esercizio 3

---

Si scriva un programma in linguaggio C che:

1. A partire dai file "dati.txt" e "indirizzi.txt" **costruisca una tabella T** contenente, per ogni studente, Matricola, Nome, Cognome, Via, Città, CAP e CdL

Si ricorda l'esistenza della procedura di libreria **void rewind (FILE \*f)** che riporta la testina di lettura a inizio file

## Esercizio 3

---

2. A partire dalla tabella T, e dato da input un intero C che rappresenta un CdL, **stampi la percentuale di studenti** (rispetto al numero totale delle matricole) **iscritti al corso C** [Ad esempio, se il numero totale delle matricole è 1000, e quello degli studenti iscritti a C è 200, il programma stamperà “20%”]
3. Scriva su un terzo file di testo “bologna.txt”, nome, cognome e numero di matricola di tutti gli studenti che abitano a Bologna