

### ESERCIZIO 1 (3 PUNTI)

Realizzare una funzione che prenda in input un numero intero e restituisca il numero di cifre che lo compongono, ignorando il segno

### ESERCIZIO 2 (3 PUNTI)

Realizzare una funzione ricorsiva

```
int CountOccurrences(int A[], int size, int val);
```

che operi su array statici di interi e restituisca il **numero di elementi di A uguali a val.**

Il prototipo non può essere modificato, ma si possono usare funzioni ausiliarie.

**ESERCIZIO 3:** Consideriamo il tipo di dato “insieme di turisti”

**3.A (1 PUNTO)** Realizzare le strutture che descrivano:  
- *turista* (codice fiscale, cognome, nome, nazionalità)  
- *insieme\_di\_turisti*, realizzato tramite **liste semplici**

**3.B (1 PUNTO)** Realizzare la funzione booleana *StessoTurista*, che ritorna true se due variabili di tipo turista sono uguali

**3.C (2 PUNTI)** Produrre i prototipi (o interfacce) delle funzioni stando molto attenti ai parametri formali  
- *InserisciTurista*  
- *CancellaTurista*  
- *InsiemeVuoto* (funzione booleana)  
- *StampaInsiemeTuristi*  
- *UnioneTraInsiemi*

**3.D (3 PUNTI)** Implementare la funzione *UnioneTraInsiemi*