

# Esercizi di programmazione\*

## 1 Lezione 9

*Suggerimento: per la lettura e scrittura di stringhe, numeri interi e reali, si faccia riferimento alla libreria IOCREMA.H*

**Esercizio 1** Si scrivano programmi che riproducono le funzioni `strlen`, `strcpy`, `strncpy`, `strcmp`, `strncmp` e `strcat`.

**Esercizio 2** Si scriva un codice ANAGRAM.C che, date due parole, determina se esse sono anagrammi oppure no.

*Suggerimento:* Un vettore ausiliario di interi permette di determinare quante volte ogni lettera appare in ciascuna delle due parole.

**Esercizio 3** Si scriva un codice CESARE.C che applica a un messaggio la cifratura di Cesare (spostamento dei singoli caratteri in avanti o indietro di un numero fisso di valori, definito da una costante).

**Esercizio 4** Si scriva un codice MAI-MIN.C che rende maiuscole le lettere minuscole e minuscole le lettere maiuscole di un messaggio dato.

*Suggerimento:* Per convenzione tutte le codifiche intere dei caratteri assegnano un insieme di valori consecutivi alle lettere minuscole da 'a' a 'z' e un altro insieme di valori consecutivi alle lettere maiuscole da 'A' a 'Z'. Quindi, si può capire se una lettera è maiuscola o minuscola confrontandola con i caratteri estremi 'a', 'z', 'A' e 'Z'. E si può convertirla sfruttando i valori delle codifiche di tali caratteri.

**Esercizio 5** Quali delle seguenti istruzioni sono equivalenti?

1. `str[0] = '\0';`
2. `strcpy(str, '');`
3. `strcat(str, '');`

**Esercizio 6** Che valore ha la stringa `s` dopo l'esecuzione delle seguenti istruzioni?

```
strcpy(s1,computer);
strcpy(s2,science);
if (strcmp(s1,s2) < 0)
    strcat(s1,s2);
else
    strcat(s2,s1);
s1[strlen(s1)-6] = '\0';
```

---

\*tratti o ispirati dal testo di K.N. King

**Esercizio 7** Si scriva un codice `EXTENSION.C` che, dato il nome di un file (completo di path), individua il nome e l'estensione del file stesso, li copia in apposite stringhe e li stampa a video. Infine, il codice deve stampare a video il nome originale del file (completo di path) sostituendo l'estensione con `.txt`. Si ammetta il caso in cui il nome non contiene punti (estensione vuota).

*Suggerimento:* Si assuma che il nome completo del file contenga al massimo un punto (`'.'`) e che l'estensione sia la sottostringa che segue il punto (eventualmente vuota, se il nome non contiene punti). Si assuma una lunghezza massima per le stringhe, definita con la direttiva `#define`.

**Esercizio 8** Si scriva un codice `MAXMIN.C`, che definisca un vettore di stringhe, lo inizializzi inserendovi i nomi dei dodici mesi e stampi a video il mese lessicograficamente “minimo” e quello lessicograficamente “massimo”.

*Suggerimento:* Un vettore di stringhe è un vettore di vettori di `char`. Conviene usare l'istruzione `typedef` per sottolineare la struttura di questi dati.

**Esercizio 9** Si scriva un codice `REVERT.C` che, data una stringa contenente una frase, inverte l'ordine delle parole di tale frase (separate da uno o più spazi) e stampa a video il risultato, calcola la loro lunghezza media e la stampa a video.

**Esercizio 10** Si scriva un codice `STRING_TO_INT.C`, che converte una stringa contenente la rappresentazione decimale di un numero naturale nel corrispondente valore.

**Esercizio 11** Si scriva un codice `STRING_MATRIX.C` che, data una stringa contenente una frase, produce un vettore di stringhe, corrispondenti alle singole parole.

*Suggerimento:* Si assuma che la lunghezza massima e il numero massimo delle parole nella frase siano noti e assegnati attraverso la direttiva `#define`.

**Esercizio 12** Si scriva un codice `PALINDROMA.C` che, data una stringa contenente una frase, determina se la frase è palindroma o no (cioè uguale alla frase ottenuta invertendo esattamente l'ordine dei caratteri).

**Esercizio 13** Si scriva un codice `DATA.C` che, data una stringa contenente una data nel formato `gg/mm/aaaa`, restituisca la stessa data in formato testuale (ad esempio, `10/04/2010` diventa `10 aprile 2010`)