

# Esercizi di programmazione\*

## 1 Lezione 16

**Esercizio 1** Si supponga di eseguire l'istruzione `n = scanf("%d%f%d",&i,&r,&j);` dove `i`, `j` e `n` sono variabili intere e `r` è una variabile reale. Supponendo di aver scritto da tastiera i seguenti ingressi, si indichi il valore delle quattro variabili e il contenuto dello *stream* di ingresso dopo la chiamata.

- 10 20 30
- 1.0 2.0 3.0
- 0.10 0.20 0.30
- a.1 .2 3

**Esercizio 2** Si scriva un programma `MAGAZZINO.C` che riceve da tastiera un articolo alla volta, nel formato

*articolo , prezzo , giorno / mese / anno*

dove *articolo* è una singola parola di lunghezza massima pari a 10, *prezzo* è un numero reale minore di 1000 e *giorno*, *mese* e *anno* indicano una data.

L'ultimo articolo viene rappresentato da una riga contenente solo la parola chiave `STOP`. Si può dare per scontato che non vi siano più di 10 articoli.

Il programma deve poi stampare a video il catalogo secondo il formato dell'esempio seguente:

Articolo	Prezzo	Data
Art. 1	12.50	21/05/2010
Art. 2	199.90	01/04/2010

**Esercizio 3** Si scriva un programma `DURATA.C` che riceve da tastiera due orari, nel formato

*ora1 : minuto1    ora1 : minuto2*

con un numero qualsiasi di spazi fra loro, e restituisce la differenza in minuti fra i due orari.

---

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

**Esercizio 4** Si scriva un programma ESTENSIONE.C che riceve da tastiera il nome di un file, ne cambia l'estensione in `.txt` e restituisce a video il risultato. Si definisca `estensione` quella parte del nome del file che sta dopo l'ultimo punto. Si tenga conto del fatto che il nome originale del file può contenere più di un punto, ma anche nessun punto.

Suggerimento: si può costruire il risultato sia con la funzione `strcat` sia con la funzione `sprintf`.

**Esercizio 5** Si scriva un programma PARAM.C che riceve da linea di comando una qualsiasi sequenza di parametri e la stampa a video, una riga alla volta, nel formato:

```
printf("Il parametro n. %d e' uguale a %s\n",p,argv[p]);
```

Si sostituisca `%d` con:

- `%2d` per avere una larghezza minima di 2 cifre: per numeri da 1 cifra, questo significa aggiungere uno spazio PRIMA del numero (con 10 o pi parametri, questo consente di avere un allineamento perfetto dell'uscita)
- `%-2d` per allineare i numeri a sinistra anziché a destra
- `%02d` per aggiungere uno 0 prima del numero anziché uno spazio

**Esercizio 6** Si scriva un programma DATA.C che riceva da linea di comando una data nel formato

GG/MM/AAAA

dove `GG` è un numero di due cifre che indica il giorno, `MM` un numero di due cifre che indica il mese e `AAAA` un numero di quattro cifre che indica l'anno, e la trasforma nella corrispondente data in formato letterale, cioè con il mese scritto a parole, verificando che i tre parametri siano effettivamente tre numeri, che siano compresi negli intervalli corretti (da 1 a 12 i mesi, da 1 a 28, 29, 30 o 31 i giorni, ecc...).