

## Esercizi seconda settimana (consegnati 29.09.2023)

Eventuali chiarimenti sulle tracce, possono essere esposti via e-mail al docente, mentre alle soluzioni ad alcuni esercizi viene dedicata la parte iniziale della lezione di martedì, se necessario.

(domande relative ad argomenti *fino* al cap. 3 del libro King, seguire le slides per gli argomenti effettivamente svolti)

### SEZIONE 1: Domande

Imparare a rispondere alle seguenti domande (Le risposte corrette si trovano nel libro di testo). Sono utili come verifica dello studio personale teorico.

- 1) Definire il concetto di variabile
- 2) Perché è necessario dichiarare il tipo delle variabili nei programmi (in C)?
- 3) Le variabili devono o possono essere definite prima del loro uso?
- 4) Cosa vuol dire inizializzare una variabile?
- 5) Cosa succede se uso una variabile (che ho dichiarato) senza averla inizializzata?
- 6) Qual è la differenza tra inizializzare e assegnare?
- 7) Quali sono i token di un programma in C? Definirli.
- 8) Cosa è `stdio.h`?
- 9) Cosa significa `#include<stdio.h>`? E' sempre obbligatorio inserirla in un programma C? Cosa succede se non si scrive?
- 10) Posso visualizzare a schermo meno di 6 cifre decimali per un float?
- 11) Descrivere come funziona la `printf`. Spiegare le specifiche di conversione `%m.pX`. Quale può essere un uso ad esempio della specifica `%p`, dove `p` è un intero?
- 12) Cosa sono le specifiche di conversione? Perché si chiamano anche segnaposti?
- 13) Come posso stampare il carattere `\` nella `printf`? E come posso stampare `"`? [non l'ho detto a lezione, cercate nel libro]
- 14) Cosa è la format string? Che tipi di caratteri può contenere?
- 15) Il compilatore C controlla che ci sia corrispondenza tra il numero di specifiche conversione e il numero di variabili indicate dopo la format string. Vero o falso? E la corrispondenza dei tipi viene controllata?
- 16) Descrivere brevemente come funziona la `scanf`. Cosa succede se dimentico `&` davanti alla variabile nella chiamata della `scanf`?
- 17) Quali sono i caratteri "bianchi"? Inserirli prima o dopo un segnaposto, è influente? (ci torneremo martedì)
- 18) Perché non è una buona idea scrivere `scanf("%d\n", &x)`? (c'è un esercizio nelle slides. Anche questo lo rivedremo bene martedì).
- 19) Cosa significa `"f"` in `printf` e `scanf`? [non l'ho detto a lezione, cercate nel libro]

## SEZIONE 2: Q&A

Esercizi tratti dalle Questions&Answers del libro di testo [Vuol dire che le risposte stanno in quella sezione del libro, alla fine delle spiegazione di ogni capitolo, prima degli esercizi] Sono curiosità utili per completare lo studio personale.

- 1) Cosa succede se manca "return 0;" alla fine del programma? E' "mandatory"?
- 2) Supponiamo di dover dichiarare 4 variabili intere, di cui conosco già il valore iniziale. Scrivere tali dichiarazioni.
- 3) Qual è la differenza tra queste due espressioni [leggetelo, poi ci torneremo]  
`float profit=2.3;`                      `float profit=2.3f;`
- 4) Come faccio a stampare il valore di una variabile intera?
- 5) C'è un limite al numero di variabili che posso stampare all'interno di una `printf`?
- 6) E' corretta una dichiarazione di questo tipo  
`int`  
`altezza;`
- 7) Cosa significa questa istruzione: `a/**/b` ? Quando il compilatore incontra un commento, cosa succede nel file `a.out`?
- 8) Come posso stampare il carattere `%` all'interno di una `printf`?
- 9) Cosa succede se la `scanf` si aspetta un intero e l'utente inserisce un input non numerico?

## SEZIONE 3: Esercizi

Gli esercizi proposti in questa sezione sono esercizi che vanno fatti “carta e penna” e poi successivamente controllati scrivendo il relativo programma .c. Infatti, le domande della prova scritta d’esame (quindi cartacea) saranno sullo stile di questi esercizi e non potranno essere risolti con l’ausilio del calcolatore. Abituatevi, quindi, a risolverli “a mano”. Sono tratti dal libro (sezione Exercises) oppure prodotti dal docente.

**Esercizi da fare (tratti dal libro King)** : [da aggiungere a quelli inseriti nelle slides] Indico il numero del capitolo, seguito dal numero dell'esercizio. La lettera W accanto al numero dell'esercizio, indica che c'è la soluzione sul sito del libro ([knking.com/books/c2](http://knking.com/books/c2)).

*Dovete seguire le selezioni proposte. Non potete fare ciecamente tutti gli esercizi, sia perché non abbiamo coperto tutti gli argomenti, sia perché non sono tutti necessari.*

2.1, 2.2(W), 2.4 (W), 2.5(W), 2.6, 3.1, 3.5 (W), 3.6

## SEZIONE 4: Esercizi di programmazione C

**Per gli esercizi di programmazione, vi chiedo di svolgere per ora solo quelli di volta in volta indicati, perché sono quelli che possono essere svolti usando quanto visto finora in C. Tutti**

**devono essere risolti, sempre, solo usando quanto spiegato fino a quel punto. Il resto degli esercizi li faremo più avanti.**

**Impostare prima l'analisi e del problema e la progettazione della soluzione, poi passare alla traduzione in C.**

per ora

1) PP 3.1 [programming project, esercizio di programmazione]

2) Divertitevi (...) con il segnaposto %m.pX.

Ovviamente **rifate il codice del programma in C visto a lezione (riportato nelle slides).** Ricordate che il nome del file .c deve essere composto da una sola parola.

- Prendete dimestichezza con i pochi comandi linux indicati (anche nel manualetto), provando a creare file, spostarli, cancellarli, creare directory, etc. Questo ovviamente suppone che stiate usando una finestra terminale per compilare ed eseguire i programmi, non ambienti di sviluppo.
  - Divertitevi a inserire errori nei programmi (togliere un ; oppure una { ) e a vedere quali messaggi vi comunica il compilatore.
-