

CORSO DI FONDAMENTI DI INFORMATICA

Prof. Maristella Matera - A.A. 2021 / 2022

Laboratorio di Programmazione in C – Laboratorio 5

Esercizio 1 – Ricorsione, liste e librerie

Si scriva un main per la gestione di una lista di interi non necessariamente ordinata. Il main deve inizializzare **una lista vuota** e, attraverso l'uso di funzioni, deve eseguire i seguenti passi (nella sequenza in cui vengono qui proposti):

- **printLista()**: stampa gli elementi della lista;
- **inserisciElemento()**: inserisce nella lista un nuovo elemento in una posizione a scelta. Nel caso di inserimenti multipli, il main richiama ripetutamente questa funzione fino a quando l'utente non decide di fermarsi inserendo il valore sentinella -1. **Dopo l'inserimento, il main stampa la lista modificata.**
- **lunghezzaLista()**: stampa la lunghezza della lista;
- **cancellaRipetizioni()**: cancella eventuali ripetizioni dalla lista. **Dopo le cancellazioni, il main stampa la lista modificata. Questa funzione può essere ricorsiva.**
- **sommaLista()**: trasforma la lista in modo che ogni elemento sia sostituito con la somma di sé stesso e tutti gli elementi che lo seguono. **Dopo la modifica, il main stampa la nuova lista. Questa funzione può essere ricorsiva.**

N.B.: Le funzioni richiamate dal main per la gestione della lista devono essere implementate all'interno di una o più librerie (ad esempio, separando le funzioni ricorsive dalle altre) che il programma dovrà includere.

Esercizio 2 – Cruciverba (tratto dal TdE del 2 Febbraio 2016)

La soluzione di un cruciverba può essere memorizzata in un file organizzato nel seguente modo:

- La prima riga contiene due numeri interi: il numero di righe e il numero di colonne del cruciverba;
- Le righe successive corrispondono alle righe del cruciverba (le caselle nere sono codificate come asterischi.)

Di seguito si riporta un esempio di schema di cruciverba memorizzato nel file:

```
5 6
SUGLI*
ODIATI
TIRI*N
TRI*CD
OE*POI
```

Definire un programma in C in grado di leggere dal file lo schema del cruciverba e scrivere in un secondo file, chiamato **parole.txt**, l'elenco delle parole, sia orizzontali, sia verticali, che hanno bisogno di una definizione (cioè le sole parole di lunghezza ≥ 2).

Il file **parole.txt** dovrà memorizzare prima tutte le parole orizzontali (una per riga), quindi una riga vuota, e infine tutte le parole verticali (una per riga).

Per esempio, dato lo schema del cruciverba riportato sopra, il file **parole.txt** dovrà memorizzare:

SUGLI
ODIATI
TIRI
TRI
CD
OE
POI

SOTTO
UDIRE
GIRI
LAI
IT
CO
INDI

Si scriva un programma in C che riceve come argomento sulla linea di comando il nome del file che memorizza lo schema del cruciverba; quindi scrive nel file **parole.txt** l'elenco delle parole di lunghezza ≥ 2 , così come illustrato nell'esempio precedente. Si assuma che la dimensione massima di un cruciverba sia 20×20 .

Il programma deve essere realizzato in modo modulare, scomponendolo almeno nelle seguenti funzioni:

1. La funzione main con passaggio come argomento della linea di comando del nome del file con lo schema del cruciverba;
2. Una funzione che legge il cruciverba dal file trasferendone il contenuto in un'opportuna struttura dati;
3. Una o più funzioni per la ricerca delle parole nella struttura dati costruita al punto precedente e scrittura nel file **parole.txt**.

Esercizio 3 – Lista della spesa

Scrivere un programma che permetta la creazione di una lista della spesa. Ogni elemento è formato da: nome, quantità, costo (per singolo elemento).

Il programma deve consentire all'utente di:

1. **inserire** un nuovo elemento nella lista della spesa: verifica se l'elemento è già presente, in questo caso somma la nuova quantità a quella precedente (si assume come valido l'ultimo costo inserito)
2. **eliminare** l'ultimo elemento inserito (se l'ultima operazione svolta è stata un aggiornamento della quantità di un prodotto già inserito, allora eliminare l'ultimo elemento della lista)
3. **calcolare** l'importo totale della spesa
4. **stampare** la lista della spesa su un file di testo in cui ogni riga corrisponde a un elemento e l'ultima riga riporta il totale della spesa
5. **terminare** il programma.

Implementare la scelta della funzionalità da eseguire tramite un menù che viene stampato a video finché l'utente non sceglie di terminare il programma.

Per il confronto tra stringhe utilizzare la funzione **strcmp(stringa1, stringa2)** che restituisce 0 se le stringhe sono uguali (funzione disponibile in **string.h**).