

Laboratorio 7 — Archivio Dinamico di Articoli con File di Testo e Binari

Assignment — inventory.c

Università di Catania — Dipartimento di Matematica e Informatica

Corso di Laurea in Informatica

Programmazione 1 e Laboratorio (Canale O-Z)

Descrizione del Programma

Si scriva un programma in linguaggio C, chiamato `inventory.c`, che gestisca un piccolo inventario di articoli. Il programma deve essere in grado di importare gli articoli da un file di testo strutturato, memorizzarli in un array dinamico di strutture, salvarli in modo compatto in un file binario, ricaricarli dal file binario per visualizzare l'elenco o aggiornare le quantità e esportarli nuovamente in un file di testo.

Il programma deve essere invocato da riga di comando nelle seguenti modalità:

```
./inventory import input.txt articoli.bin
./inventory list articoli.bin
./inventory update articoli.bin
./inventory export articoli.bin output.txt
```

È richiesto l'uso della seguente struttura dati:

```
typedef struct {
    char    codice[16];
    char    descrizione[64];
    double  prezzo;
    int     quantita;
} Articolo;
```

Su disco, il file binario `articoli.bin` deve contenere un intero `int n_articoli` seguito da un array di `Articolo` consecutivi.

Requisiti di Implementazione

Il programma deve essere strutturato nei seguenti punti:

PUNTO A: Verificare che la combinazione di argomenti sia una delle seguenti:

```
./inventory import input.txt articoli.bin, ./inventory list
```

```
articoli.bin, ./inventory update articoli.bin, ./inventory  
export articoli.bin output.txt. In caso contrario, stampare:
```

```
Errore: uso corretto ./inventory [import|list|update|export] ...
```

e terminare con exit status 1.

PUNTO B: In modalità `import`, aprire il file di testo di input, leggere una riga per volta con `fgets` e usare `strtok` con separatore `','` per ottenere i quattro campi `codice;descrizione;prezzo;quantita`. I campi devono essere validati come segue: codice non vuoto e lungo al massimo 15 caratteri, descrizione non vuota e lunga al massimo 63 caratteri, prezzo reale non negativo, quantità intera maggiore o uguale a 0. Se una riga è malformata o contiene valori non validi, stampare:

```
Errore: riga articolo non valida.
```

e terminare con exit status 1. In caso di successo, scrivere il numero di articoli e l'array binario nel file `articoli.bin` e stampare:

```
Import completato con successo.
```

PUNTO C: In modalità `list`, aprire il file binario, leggere `n_articoli` e l'intero array e stampare l'intestazione:

```
Elenco articoli:
```

Ogni articolo va stampato in una riga nel formato:

```
COD001 - Quaderno a righe - Prezzo: 1.50 - Quantita: 120
```

Se il file non può essere aperto, stampare:

```
Errore: impossibile aprire il file binario.
```

Se il file è corrotto, stampare:

```
Errore: file binario corrotto.
```

Se non ci sono articoli, stampare:

```
Elenco articoli:
```

```
Nessun articolo presente.
```

PUNTO D: In modalità `update`, aprire il binario, caricare l'array in memoria, chiedere all'utente il codice dell'articolo e la nuova quantità, cercare linearmente il codice e aggiornare il campo `quantita`. Se l'articolo non viene trovato, stampare:

`Errore: articolo non trovato.`

e terminare con exit status 1. Dopo l'aggiornamento, riscrivere l'intero file binario e stampare:

`Quantita aggiornata con successo.`

PUNTO E: In modalità `export`, caricare gli articoli dal binario e scrivere su `output.txt` una riga per ciascun articolo nel formato `codice;descrizione`. Se il file di output non può essere aperto, stampare:

`Errore: impossibile aprire il file di output.`

In caso di successo, stampare:

`Export completato con successo.`

Il programma deve deallocare sempre la memoria dinamica prima di terminare.

Output di Controllo (Esempi di Esecuzione)

Caso 1 — Import valido e lista degli articoli

Contenuto di `input.txt`:

```
COD001;Quaderno a righe;1.50;120
COD002;Penna blu;0.80;300
COD003;Libro C Programming;35.90;15
```

Comando di import:

```
./inventory import input.txt articoli.bin
```

Il programma deve stampare a schermo:

```
Import completato con successo.
```

(Exit code atteso: 0)

Comando di list:

```
./inventory list articoli.bin
```

Il programma deve stampare a schermo:

Elenco articoli:

COD001 - Quaderno a righe - Prezzo: 1.50 - Quantita: 120

COD002 - Penna blu - Prezzo: 0.80 - Quantita: 300

COD003 - Libro C Programming - Prezzo: 35.90 - Quantita: 15

(Exit code atteso: 0)

Caso 2 — Aggiornamento della quantità

Comando:

```
./inventory update articoli.bin
```

Input da tastiera:

Inserisci il codice dell'articolo da aggiornare:

COD002

Inserisci la nuova quantita:

250

Il programma deve stampare a schermo:

Quantita aggiornata con successo.

(Exit code atteso: 0)

Caso 3 — Esportazione su file di testo

Comando:

```
./inventory export articoli.bin output.txt
```

Il programma deve stampare a schermo:

Export completato con successo.

Il file `output.txt` deve contenere righe nel formato compatibile con `import`. *(Exit code atteso: 0)*