## Gestione dell'inventario del magazzino

Obiettivo dell'esercizio:

Scrivere un programma in linguaggio C che permetta di gestire un magazzino e verificare se un ordine può essere

evaso sulla base delle quantità disponibili.

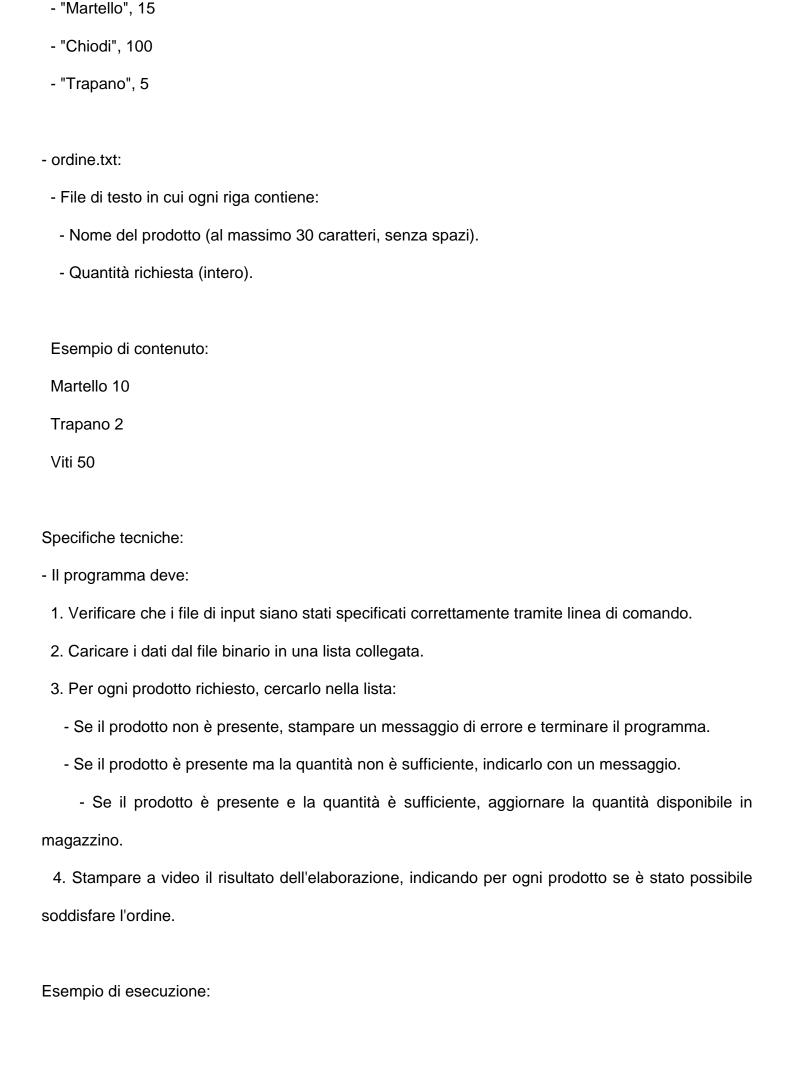
Il programma deve:

- 1. Leggere due file:
- prodotti.dat (file binario): contiene l'elenco dei prodotti disponibili in magazzino e le relative quantità.
  - ordine.txt (file di testo): contiene un elenco di prodotti richiesti e le quantità desiderate.
- 2. Caricare i dati dal file binario in una lista collegata.
- 3. Verificare per ogni prodotto richiesto se è presente in magazzino e se la quantità è sufficiente.
- 4. Stampare un messaggio per ogni prodotto richiesto indicando se l'ordine è stato soddisfatto o meno.
- 5. Terminare con un errore nel caso un prodotto richiesto non sia presente in magazzino.

Formato dei file di input:

- prodotti.dat:
  - File binario contenente record strutturati, ciascuno formato da:
    - Nome del prodotto (array di 31 caratteri, incluso il terminatore '\0').
    - Quantità disponibile (intero).

Esempio di record:



## Input: - prodotti.dat contiene: - "Martello", 15 - "Chiodi", 100 - "Trapano", 5 - ordine.txt contiene: Martello 10 Trapano 2 Viti 50 Output: Prodotto: Martello - Ordine soddisfatto (10/15 disponibili) Prodotto: Trapano - Ordine soddisfatto (2/5 disponibili) Errore: Prodotto "Viti" non trovato in magazzino. File richiesti per il progetto: 1. main.c: Contiene la funzione main e il flusso principale del programma. 2. funzioni.c: Contiene le funzioni per la gestione della lista collegata e la logica del programma. 3. funzioni.h: Contiene le dichiarazioni delle funzioni e la definizione dei tipi di dato. 4. Makefile: Permette la compilazione del programma con il comando make. File di esempio: - prodotti.dat (già fornito nel formato binario). - ordine.txt (già fornito nel formato testo). Istruzioni per la consegna:

- Consegnare tutti i file richiesti sopra, inclusi i file di input e il Makefile.
- Assicurarsi che il programma sia compilabile ed eseguibile senza errori.
- La valutazione terrà conto di:
  - Correttezza.
  - Efficienza nell'uso delle risorse.
  - Stile e chiarezza del codice.