Fondamenti di Informatica T-1
modulo 2

Laboratorio 06: ***strutture e***

file di testo/binari

Esercizio 1: Strutture e file

Esercizio - lettura/filtro/scrittura

- È dato un file di testo **PEOPLE.TXT** che contiene i dati di una serie di persone (non più di 20), una persona per riga
- Si vuole realizzare un programma che, una volta letti da file i dati di queste persone, ne estragga l'insieme di persone compatibili con una nuova persona data, salvando il risultato sul file binario **PARTNERS.DAT**
  - ***Due persone sono compatibili*** se sono di sesso diverso e la differenza di età, *riferita solo all'anno*, non supera i 5 anni

Esercizio 1 – Specifica del problema

- Ogni riga del file contiene, nell'ordine:
  - cognome (non più di 20 caratteri)
  - un separatore ';'
  - nome (non più di 20 caratteri)
  - un separatore ';'
  - data di nascita nel formato **gg/mm/aaaa**
  - uno e un solo spazio
  - un carattere ('M' o 'F') che indica il sesso
- Si definisca opportunamente una struttura dati **persona** di tipo struct

Esercizio 1- Specifica del problema

Realizzare le seguenti funzioni

- una funzione **lettura(…)** che, dato il nome del file (ed eventualmente altri parametri se opportuno), legga i dati delle persone dal file e li metta in un *array di* ***persona*** di nome **elenco**
  - Quanto deve essere grande l'array?
  - Si mostri a video l'array così costruito
- una funzione **compatibili(…)** che, date due persone, restituisca *vero* solo se le due persone sono compatibili
  - si mostri a video un esempio d'uso della funzione con due persone scelte all'array a vostro piacere
- **main()** che invochi **lettura** per acquisire le persone, poi chieda all'utente i dati di una nuova persona per scrivere infine su file binario l'insieme delle persone **compatibili** con la persona data

Esercizio 1 - Suggerimenti

- Analizzare i “dati”
  - Quanti “dati complessi” sono chiamati in causa?
    - Persona
    - Data (di nascita)
  - Quali campi? E di che tipo (e dimensione…)?
- Seguire un approccio a moduli
  - Sicuramente un ***modulo separato*** per la definizione delle strutture dati
  - Un modulo per le funzioni chiamate a manipolare queste strutture dati
- Lettura da file di testo
  - Individuare il tipo dei campi che vogliamo leggere
  - …e i relativi separatori

Esercizio 2

Gestione di una libreria

- Realizzare un programma di gestione per una libreria
- Il programma deve acquisire da standard input una serie di *libri* (con numero massimo fissato a priori)
  - NOTA: si supponga che l’acquisizione avvenga per codici ordinati in senso crescente
- Il programma deve successivamente presentare un menu a tre voci:
  - Stampa a video dell’insieme di libri
  - Ricerca e stampa di un libro a partire dal codice
    - Si usi a tal scopo la RICERCA BINARIA (vedi NOTA)
  - Uscita dal programma
- Estendere poi il programma supportando il salvataggio/caricamento dei libri su/da file binario

Esercizio 2 - Note

- Ogni libro è caratterizzato da
  - Un codice (intero)
  - Un titolo (stringa)
  - Un autore
    - Un autore è descritto da nome e cognome (stringhe)
  - Il numero di copie presenti in libreria (intero)
- Si suddivida l'acquisizione in sottofunzioni
  - Acquisizione del vettore di libri
    - Acquisizione del singolo libro
    - Acquisizione del singolo autore

Esercizio 2 - Estensioni

Estensioni della libreria (utile esercizio "per casa")

- Un'ulteriore voce di menu per la gestione dei libri in esaurimento
  - In questo caso all'utente viene chiesto un valore soglia
  - Il programma deve scrivere su un file di testo il codice e il numero di copie di tutti i libri il cui numero di copie è inferiore alla soglia data
- Fare in modo che tale elenco sia ***ordinato in senso crescente*** rispetto al numero di copie presenti
  - Alternativa 1: il vettore risultato viene riempito usando l'inserimento ordinato
  - Alternativa 2: riempimento del vettore e poi uso di un algoritmo di ordinamento (modificando l'operatore di confronto)

Esercizio 3: file e conteggio

Conta parole

- Si scriva un programma C che conti il numero dei ***caratteri***, delle ***parole*** e delle ***linee*** contenute in un file di testo di nome specificato e uguale a **PAROLE.TXT**