

Fondamenti di Informatica L-A

Esercitazione 7

I File

Paolo Torroni

Università degli Studi di Bologna
Laurea in Ingegneria Elettronica e delle Telecomunicazioni

Anno Accademico 2007/2008

Nota introduttiva

- ▶ In questa esercitazione sono proposte varie casistiche di lettura/modifica/scrittura di file di testo e binari.
- ▶ Non tutti gli esercizi sono proposti per le due ore di laboratorio di oggi!
- ▶ In particolare, oggi guarderemo gli **esercizi 1 e 2**.
- ▶ Gli altri esercizi sono però caldamente **consigliati** in vista dell'esame scritto.
- ▶ Sul sito web verranno messi a disposizione a breve:
 - ▶ Il testo di questa esercitazione;
 - ▶ I file binario e di testo necessari;
 - ▶ Un file sorgente C con alcune funzioni sui file che potrebbero essere utili alla soluzione di alcuni degli esercizi proposti (e su cui si consiglia di prepararsi).

△ 1. Lettura di un file binario ▽

- ▶ Salvare nel direttorio di lavoro contenente l'eseguibile (es. C:\temp\ese8\bin) il file "ListaB.pdf" disponibile nella **Parte 2** di questa esercitazione.
 - ▶ Alla finestra di download, scegliere "salva" invece che "apri". Il file dovrebbe essere salvato sul desktop. Trascinarlo dal desktop al direttorio dell'eseguibile.
- ▶ È un **file binario**, composto di record il cui formato è:
 - ▶ un intero corto ID;
 - ▶ un intero lungo MATRICOLA;
 - ▶ una stringa NOME di 30 caratteri, compreso il terminatore;
 - ▶ una stringa COGNOME di 30 caratteri, compreso il terminatore.
- ▶ Partendo dal file binario "ListaB.pdf", scrivere un programma in C che:
 - ▶ legga da input un numero intero N
 - ▶ stampi a video NOME e COGNOME dello studente con la MATRICOLA più vicina in assoluto a N .

△ 2. Lettura di un file di testo ▽

- ▶ Salvare nel direttorio di lavoro contenente l'eseguibile (es. C:\temp\ese8\bin) il file "ListaT.pdf" disponibile nella **Parte 3**.
 - ▶ Seguire la stessa modalità indicata per il punto 1.
- ▶ È un **file di testo**, composto di righe il cui formato è:
 - ▶ un intero ID seguito dal carattere virgola (',');
 - ▶ un intero MATRICOLA seguito da virgola;
 - ▶ una stringa di testo NOME seguito da virgola;
 - ▶ una stringa di testo COGNOME seguito da virgola.
- ▶ I campi ID, MATRICOLA, NOME e COGNOME **non** contengono virgole al proprio interno.
- ▶ Partendo dal file di testo "ListaT.pdf", scrivere un programma in C che:
 - ▶ legga da input un numero intero N ;
 - ▶ stampi a video NOME e COGNOME dello studente con la MATRICOLA più vicina in assoluto a N .

△ 3. Scrittura di un file di testo in un file binario ▽

- ▶ Salvare nel direttorio dell'eseguibile il file "ListaT.pdf".
- ▶ Salvare il contenuto del file "ListaT.pdf" in un file binario "NuovaLista.bin" che contiene esattamente l'informazione di ListaT.pdf, però in formato binario.

△ 4. Scrittura di un file binario in due file di testo ▽

- ▶ Salvare nel direttorio dell'eseguibile il file "ListaB.pdf".
- ▶ Salvare il contenuto del file "ListaB.pdf" in **due** file di testo, "ListaN.txt" e "ListaC.txt" che contengano:
 - ▶ ListaN.txt → i campi ID, MATRICOLA e NOME di tutti i record contenuti in ListaB.pdf.
 - ▶ ListaC.txt → i campi ID, MATRICOLA e COGNOME di tutti i record contenuti in ListaB.pdf.

△ 5. Modifica di un file binario ▽

- ▶ Salvare nel direttorio dell'eseguibile il file "ListaB.pdf".
- ▶ Leggere una stringa S da input.
- ▶ Calcolare in N il valor medio dei campi ID dei primi 10 elementi del file ListaB.pdf.
- ▶ Aggiornare il file "ListaB.pdf", sommando N al campo ID dei soli record in cui il campo COGNOME è maggiore (in ordine lessicografico) di S .

△ 6. Modifica di un file di testo ▽

- ▶ Salvare nel direttorio dell'eseguibile il file "ListaT.pdf".
- ▶ Aggiornare il file "ListaT.pdf", per fare in modo che tutte le parole dei campi NOME e COGNOME di tutti i record abbiano il primo carattere maiuscolo e gli altri minuscoli.

△ 7. Esercizi aggiuntivi sui file ▽

Sempre a partire dai file trattati in questa esercitazione (ListaT.pdf, ListaB.pdf, ListaN.txt e ListaXC.txt), si provino a implementare algoritmi per:

1. L'**ordinamento** con merge sort di un file di testo/binario secondo il campo NOME, COGNOME, ID o MATRICOLA;
2. L'**unione** del contenuto dei file ListaN.txt e ListaC.txt in un unico file ListaNC.txt che deve avere lo stesso contenuto di ListaT.pdf
3. La **ricerca binaria** di un record su file binario ordinato. L'algoritmo per la ricerca binaria è spiegato nel prossimo lucido.

»» Ricerca binaria ««

- ▶ La funzione

```
int BinarySearch(int *V, int value, int low, int high)
```

funziona con un vettore V di interi, dove cerca un valore $value$ tra gli elementi di posizione low e $high$.

- ▶ Ad esempio, se consideriamo un vettore A di 10 interi, e cerchiamo l'elemento m , dato un intero j usiamo l'istruzione

$$j = \text{BinarySearch}(A, m, 0, 9);$$

per scoprire:

- ▶ che m è in $A[j]$, se $j > 0$;
- ▶ oppure che m non è in A , se j vale -1 .
- ▶ L'algoritmo **funziona solo se A è un vettore ordinato**.
- ▶ È definito nella pagina seguente.

»» int BinarySearch(int *V, int value, int low, int high) ««

1. se ($high < low$) /* **caso base**: non ci sono elementi tra low e high in cui cercare $m \Rightarrow$ elemento non trovato */
 - 1.1 Restituisci -1
2. altrimenti
 - 2.1 Sia mid l'indice dell'elemento medio tra low e high
 - 2.2 Verifica se m è alla posizione $A[mid]$
 - 2.3 Se così è, restituisci mid. /* **caso base**: elemento trovato */
 - 2.4 Altrimenti,
 - 2.4.1 Se ($A[mid] > m$) /* bisogna cercarlo prima di mid */
chiama ricorsivamente BinarySearch tra low e mid-1;
 - 2.4.2 Se ($A[mid] < m$) /* bisogna cercarlo dopo mid */
chiama ricorsivamente BinarySearch tra mid+1 e high.