

Descrizione del programma

Scrivere un programma in C che:

- **A.** Prenda un input da tastiera (argomenti della funzione main) costituito da 1) nome di un file di input che contiene dati binari (ES: **data.bin**), 2) nome di un file di testo di output (es: **output.txt**). Il file contiene una collezione di record. Ogni record contiene le seguenti informazioni

codice	(max 30 caratteri)
nome	(max 30 caratteri)
cognome	(max 30 caratteri)
Eta'	(intero positivo)

- **B.** legga tutti i record contenuti nel file binario di input e li **memorizzi in un array di record di dimensione 20 o superiore**; **NB:** i record dell'array che non conterranno dati vanno marcati con il valore ZERO nel campo eta';
- **C.** stampi tutti i record dell'array su standard output (solo i record che contengono dati), dal primo all'ultimo;
- **D.** ordini l'array di record in **ordine lessicografico crescente rispetto al cognome** (si ordini solo la porzione iniziale dell'array, ovvero quella che contiene dati) mediante l'algoritmo **insertion sort**;
- **E.** stampi tutti i record dell'array su standard output (solo i record che contengono dati), dal primo all'ultimo;
- **F.** scriva nel file di output, **in formato testo**, tutti i record dell'array che hanno un campo codice che inizia per X o per Z, **un record per riga, nel seguente modo:**

codice	nome	cognome	eta
---------------	-------------	----------------	------------

Il programma potrà' essere strutturato in un unico file sorgente, ma dovra' contenere almeno le seguenti funzioni:

- ≠ **ReadInput (4pt):** funzione che prende in input l'array di puntatori a carattere argv ed il numero di argomenti argc della funzione main, controlla che gli argomenti richiesti siano nel numero e nei limiti specificati, e restituisca tali parametri in una struct; se il controllo non va a buon fine, stampa un messaggio sullo standard error e termina il programma.
- ≠ **buildArray() (8pt):** funzione che legge il file binario di input e crea l'array di record;
- ≠ **printRecord() (4pt):** funzione che stampa tutti i record contenuti all'interno dell'array (solo i record che contengono dati);

€ **sortArray()** (7pt): funzione ordina l'array di record come specificato al pnto D.

€ **elab()** (7pt): funzione che crea il file di output in formato testo come specificato nel punto F.

VIETATO usare variabili globali.

Durata della prova: 120 minuti. NB: Inserire nome, cognome e numero di matricola all'interno del file sorgente.

OUTPUT DI CONTROLLO

./a.out data.bin output.txt

**** PrintArray() ****

X001 Mario Rossi 24
Y002 Tizio Saio 34
Z004 Osvaldo Pierantoni 44
X005 Libero Rossi 60
X006 Poncho Sanchez 50
Y012 Aurelio Izzo 23
Z008 Brad Pitt 60
Z009 Speedy Gonzales 10
X011 Henry Potter 40
Y034 Robert DeNiro 82
Z014 Al Pacino 85
P033 Joe Pesci 82

**** PrintArray() (after sorting)****

Y034 Robert DeNiro 82
Z009 Speedy Gonzales 10
Y012 Aurelio Izzo 23
Z014 Al Pacino 85
P033 Joe Pesci 82
Z004 Osvaldo Pierantoni 44
Z008 Brad Pitt 60
X011 Henry Potter 40
X001 Mario Rossi 24
X005 Libero Rossi 60
Y002 Tizio Saio 34
X006 Poncho Sanchez 50

(contenuto del file output.txt)

Z009 Speedy Gonzales 10

Z014 Al Pacino 85

Z004 Osvaldo Pierantoni 44

Z008 Brad Pitt 60

X011 Henry Potter 40

X001 Mario Rossi 24

X005 Libero Rossi 60

X006 Poncho Sanchez 50