

Fondamenti di Informatica T-1 modulo 2

Laboratorio 07: *file e allocazione dinamica*

Esercizio 1

Gestione degli esami di uno studente

- Realizzare un programma che permetta di gestire gli esami di uno studente
- Funzionalità richieste
 - Caricamento degli esami sia da **file di testo** che da **file binario**
 - Si assuma che la prima riga (il primo campo) del file da cui leggere gli esami contenga **IL NUMERO DI ESAMI** presenti nel file
 - Stampa degli esami
 - Calcolo della media pesata sul numero di crediti
 - Salvataggio su file di testo dell'elenco degli esami la cui dicitura contiene una stringa data, unitamente alla media calcolata solo su questi esami

2

Esercizio 1 - Note

- Un esame è caratterizzato da
 - Dicitura, contenente anche degli spazi (ma, sempre, solo ed esattamente **35** caratteri)
 - Numero di crediti (intero)
 - Voto (intero)
- *Ancora non gestiamo liste...*
- Quindi modelliamo l'insieme degli esami come un vettore (di strutture opportune) la cui dimensione **NON È NOTA A PRIORI**
 - Suggestivo: utilizzare una struttura definita in termini di
 - Dimensione LOGICA (e FISICA) dell'array
 - Un array di strutture esami

3

Esercizio 1 - Suggerimenti

- Salvataggio su file dell'elenco degli esami la cui dicitura **CONTIENE** una stringa data, unitamente alla media calcolata solo su questi esami
 - È un tipico caso di filtro (sulla dicitura dell'esame)
 - Ho un vettore di esami, ottengo un nuovo vettore di esami
 - Poi lo salvo ricorrendo ad un'ALTRA funzione
- Come realizzo il filtro?
 - Riguardare **string.h**
- Come posso conoscere la dimensione del vettore filtrato?
 - Suggestivo: scandire l'elenco di partenza due volte
 - La prima per calcolare la dimensione del secondo elenco
 - La seconda per riempirlo

4

Esercizio 1 - Requisiti

Funzionalità da realizzare:

- `Boolean leggiEsamiTxt(char *nomeFile, VettoreEsami* vett);`
- `Boolean leggiEsamiBin(char *nomeFile, VettoreEsami* vett);`
- `void stampaEsami(VettoreEsami vett);`
- `float media(VettoreEsami vett);`
- `VettoreEsami filtra(VettoreEsami vett, char *pattern);`
- `Boolean salvaReport(VettoreEsami vett, char* nomeFile);`

5

Esercizio 2

Gestione articoli in vendita

Realizzare un programma che permetta di gestire gli articoli in vendita con le seguenti funzionalità:

- Caricamento del prezzo e della quantità degli articoli già venduti da due **file di testo** `listino.txt` e `venduti.txt`
 - Ciascuna riga di `listino.txt` specifica, separati tra loro da uno spazio, la tipologia di articolo in vendita (al più dieci caratteri senza spazi), la sua marca (al più 10 caratteri senza spazi) e il suo prezzo in euro (`float`)
 - Ciascuna riga di `venduti.txt` specifica, separati tra loro da uno spazio, la tipologia e la marca di ciascun articolo venduto
- Stampa dell'elenco degli articoli già venduti suddivisi per marca e tipo con prezzo unitario e quantità totale
- Salvataggio su file binario dell'elenco precedente
- Calcolo dell'incasso ottenuto suddiviso per **marca**

6

Esercizio 2 - Note

- Ovviamente possono esserci più occorrenze di uno stesso articolo in `venduti.txt`
- Non è detto che ogni articolo presente in `listino.txt` sia presente anche in `venduti.txt`
- Invece se un articolo è presente in `venduti.txt` allora è sicuramente presente anche in `listino.txt`

listino.txt

```
acqua fiuggi 7.0
acqua recoaro 6.0
pasta barilla 0.3
pasta dececco 0.5
acqua dececco 0.2
pasta fiuggi 0.1
```

venduti.txt

```
acqua recoaro
acqua recoaro
pasta barilla
pasta barilla
acqua recoaro
pasta dececco
pasta fiuggi
pasta fiuggi
acqua dececco
```

7

Esercizio 2 - Suggerimenti

- Realizzare **una struttura dati item** in cui sia possibile specificare la tipologia di un articolo, la sua marca, il prezzo in euro e la quantità di articoli già venduti
- Realizzare **una struttura dati income** in cui sia possibile specificare la marca di un articolo e l'incasso relativo a tale marca
- Si ricorda l'esistenza della funzione `void rewind(*FILE)` che riporta la testina di lettura a inizio file e della funzione `int strcmp(char* st, char* ct)` per il confronto tra stringhe

8

Esercizio 2 - Requisiti

Funzionalità da realizzare:

- `item* articoli(FILE* listino, FILE* venduti, int* len);`
- `Boolean scriviArticoli(char* nomeFileBinario, item* vett, int len);`
- `income* calcolaIncasso(item* vett, int lenVett, int* lenIncasso);`