

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

Prima di cominciare: si scarichi dal sito <http://esamix.labx> il file **StartKit1A.zip** contenente i file necessari (*solution* di VS2005 e progetto compresi).

Avvertenze per la consegna: apporre all'inizio di ogni file sorgente un commento contenente i propri dati (**cognome, nome, numero di matricola**) e il **numero** della prova d'esame. Al termine, **consegnare tutti i file sorgenti** e i file contenuti nello StartKit. Chi non rispetta le specifiche sarà opportunamente penalizzato.

Nota 1: NON SARANNO CORRETTI gli elaborati che presenteranno un numero "non affrontabile" di errori di compilazione.

Nota 2: il main non è opzionale; i test richiesti vanno implementati.

Consiglio: per verificare l'assenza di *warning*, effettuare di tanto in tanto "Rebuild All".

Una biblioteca desidera supportare la prenotazione informatizzata dei libri in suo possesso. A tal scopo la biblioteca mantiene l'elenco dei libri che possiede e la lista dei libri prestati o riconsegnati in due file distinti, rispettivamente `libri_posseduti.txt` e `libri_prestati.txt`.

Ciascuna riga del primo file contiene: codice ISBN (esattamente 13 caratteri), il carattere '*', il cognome dell'autore (al più 20 caratteri, eventualmente contenente anche spazi), il carattere '*', il nome dell'autore (al più 20 caratteri, eventualmente contenente anche spazi), il carattere '*', ed infine il numero di copie possedute di tale libro.

Ciascuna riga del secondo file contiene: codice ISBN (esattamente 13 caratteri), uno spazio, e +1 o -1 (un intero) a seconda che tale libro sia stato riconsegnato o prestato. Nel primo file ciascun codice ISBN compare una ed una sola volta, nel secondo file 0 o più volte (vedi file di esempio nello StartKit).

Esercizio 1 – Struttura dati `Libro` e funzioni di lettura (`moduli element.h` e `libro.h/libro.c`)

Si definisca un'opportuna struttura dati `Libro` che contenga codice ISBN, nome e cognome di un autore, copie possedute e copie disponibili di un libro.

Si realizzi la funzione `int calcolaCopiePrestate(char* fileName, char* isbn)`; che, dato un codice isbn ed un nome di file contenenti i libri prestati, restituisca il numero di copie attualmente prestate del libro con codice isbn dato.

Esercizio 2 – Copie in possesso e copie disponibili (`modulo libro.h/libro.c`)

Si realizzi una funzione `list inventario(char* filePosseduti, char* filePrestati)`; che, dati i nomi di due file contenenti i libri posseduti ed i libri prestati, crei la lista di libri in possesso alla biblioteca, specificando opportunamente sia il numero di copie in possesso che il numero di copie disponibili di ciascun libro nella biblioteca. A tal fine si utilizzi la funzione di cui all'esercizio 1.

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

Esercizio 3 – Comparazione fra due libri

Si realizzi una funzione `compare(...)`; che, date due strutture dati di tipo `Libro L1` ed `L2`, effettui il confronto fra queste in base al numero di copie attualmente prestate (differenza tra numero di copie possedute e numero di copie disponibili) e restituisca -1 se `L1` è attualmente prestato meno di `L2`, +1 in caso contrario.

Se due libri sono stati letti lo stesso numero di volte, bisogna controllare l'ordine lessicografico rispetto (rispettivamente) al cognome e al nome dell'autore, restituendo -1 se l'autore di `L1` precede lessicograficamente l'autore di `L2`, +1 altrimenti. Se anche nome e cognome sono identici, allora restituire 0.

Esercizio 4 – Lista ordinata di libri più prestati

Si realizzi una funzione `piuLetti(...)`; che, data una lista di libri, crei una nuova lista di libri avente in testa i libri più prestati ed in coda i libri meno prestati. Si consiglia l'utilizzo della funzione `insord(...)`; vista a lezione per le liste modificandola opportunamente affinché usi la funzione `compare(...)`; realizzata precedentemente. Si suggerisce di prestare molta attenzione poichè tale funzione fornisce il criterio opposto di ordinamento, e quindi sarà necessario modificare opportunamente la funzione `insord(...)`.

La lista deve essere creata e gestita tramite le primitive definite per l'ADT lista.

Esercizio 5 – Selezione di libri per autore

Si realizzi una funzione `select(...)`; che, data una lista di libri e due stringhe ben formate `inizio_cognome` ed `inizio_nome`, restituisca un array contenente i libri della lista data il cui autore abbia cognome che inizia per `inizio_cognome` ed il nome per `inizio_nome`. La funzione deve restituire opportunamente un riferimento all'array creato e la relativa dimensione. A tal scopo si realizzi una funzione `iniziaPer(...)`; che date due stringhe (la prima più corta o della stessa lunghezza della seconda) restituisca 1 se i primi caratteri della seconda stringa sono esattamente i caratteri della prima, 0 altrimenti.

Il vettore restituito deve essere il più piccolo possibile.

La lista deve essere gestita tramite la notazione a puntatori e non tramite le primitive definite per l'ADT lista.

Dove specificato realizzare le funzioni nei moduli indicati; laddove non specificato si lascia libertà di scelta al candidato.

Ogni funzione realizzata deve essere opportunamente testata invocandola correttamente dalla funzione `main()`. Al fine di verificare la correttezza delle funzioni realizzate, si stampi a video l'elenco dei libri ordinati in base alla funzione di cui all'esercizio 4, il cui autore ha cognome con primi caratteri "Mel" e nome con primi caratteri "Pa".

Bonus: si abbia cura di deallocare la memoria allocata dinamicamente dal programma realizzato.

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

"element.h":

```
#ifndef _ELEMENT_H
#define _ELEMENT_H

typedef struct
{
    char isbn[14];
    char cognome[21];
    char nome[21];
    int posseduti;
    int disponibili;
} Libro;

typedef Libro element;

#endif /* _ELEMENT_H */
```

"element.h":

```
#include <stdio.h>

int readField(char buffer[], char sep, FILE *f);
```

"readField.c"

```
#include "readField.h"

int readField(char buffer[], char sep, FILE *f){
    int i = 0;
    char ch = fgetc(f);
    while (ch != sep && ch != 10 && ch != EOF){
        buffer[i] = ch;
        i++;
        ch = fgetc(f);
    }
    buffer[i] = '\0';
    return i;
}
```

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

"list.h"

```
#ifndef LIST_H
#define LIST_H

#include "element.h"

typedef struct list_element {
    element value;
    struct list_element *next;
} item;
typedef item* list;
typedef int boolean;

/* PRIMITIVE */
list emptylist(void);
boolean empty(list);
list cons(element, list);
element head(list);
list tail(list);

#endif
```

"list.c":

```
#include <stdio.h>
#include <stdlib.h>
#include "list.h"

/* OPERAZIONI PRIMITIVE */
list emptylist(void) { /* costruttore lista vuota */
    return NULL;
}

boolean empty(list l) { /* verifica se lista vuota */
    return (l==NULL);
}

list cons(element e, list l) {
    list t; /* costruttore che aggiunge in testa alla lista */
    t=(list)malloc(sizeof(item));
    t->value=e;
    t->next=l;
    return(t);
}

element head(list l) { /* selettore testa lista */
    if (empty(l)) exit(-2);
    else return (l->value);
}

list tail(list l) { /* selettore coda lista */
    if (empty(l)) exit(-1);
    else return (l->next);
}
```

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

"libri.h":

```
#ifndef _LIBRI_H
#define _LIBRI_H

#include <stdio.h>
#include "element.h"
#include "list.h"

int calcolaCopiePrestate(char* fileName, char* isbn);
list inventario(char* filePosseduti, char* filePrestati);
int compare(Libro L1, Libro L2);
list insord(element el, list l);
list piuLetti(list l);
int iniziaPer(char* str1, char* str2);
Libro* select(char* inizio_cognome, char* inizio_nome, list l, int* dim);

#endif _LIBRI_H
```

"libri.c":

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "libri.h"
#include "readField.h"

int calcolaCopiePrestate(char* fileName, char* isbn) {
    FILE* fp;
    int ok=1, prestato, copiePrestate=0;
    char temp_isbn[14];

    fp = fopen(fileName, "rt");
    if(fp==NULL) exit(-1);

    do{
        ok = ok && ( fscanf(fp, "%s %d", temp_isbn, &prestato) == 2 );
        if( ok && (strcmp(temp_isbn, isbn))==0 ){
            copiePrestate += prestato;
        }
    }while(ok);

    fclose(fp);
    return copiePrestate;
}

list inventario(char* filePosseduti, char* filePrestati){
    FILE* fp;
    list res;
    Libro book;
    int ok;

    fp = fopen(filePosseduti, "rt");
    if(fp==NULL) exit(-1);

    res = emptylist();

    do{
```

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

```
    ok = readField(book.isbn, '*', fp);
    ok = ok && readField(book.cognome, '*', fp);
    ok = ok && readField(book.nome, '*', fp);
    ok = ok && (fscanf(fp, "%d", &(book.posseduti))==1 );
    book.disponibili = book.posseduti - calcolaCopiePrestate(filePrestati,
book.isbn);
    if(ok){
        res=cons (book, res);
        fgetc(fp);
    }
}while(ok);

fclose(fp);
return res;
}

int compare(Libro L1, Libro L2){
    if( L1.posseduti-L1.disponibili > L2.posseduti-L2.disponibili){
        return +1;
    }
    else if( L1.posseduti-L1.disponibili < L2.posseduti-L2.disponibili){
        return -1;
    }
    else if(strcmp(L1.cognome, L2.cognome)>0){
        return +1;
    }
    else if(strcmp(L1.cognome, L2.cognome)<0){
        return -1;
    }
    else if(strcmp(L1.nome, L2.nome)>0){
        return +1;
    }
    else if(strcmp(L1.nome, L2.nome)<0){
        return -1;
    }
    else{
        return 0;
    }
}

list insord(element el, list l){
    if (empty(l)){
        return cons(el, l);
    }
    else if ( compare( el, head(l)) >= 0 ){
        return cons(el, l);
    }
    else{
        return cons(head(l), insord(el, tail(l)));
    }
}

list piuLetti(list l){
    list res;
    Libro book;

    res = emptylist();
    while(!empty(l)){
        book = head(l);
        res = insord(book, res);
    }
}
```

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

```
    l = tail(l);
}
return res;
}

int iniziaPer(char* str1, char* str2){
    int res = 1;
    while(res==1 && *str1 != '\0'){
        if(*str1!=*str2){
            res=0;
        }
        str1++;
        str2++;
    }
    return res;
}

Libro* select(char* inizio_cognome, char* inizio_nome, list l, int* dim){
    Libro book;
    Libro* res;
    list temp_list;

    *dim=0;
    temp_list=l;
    while(temp_list!=NULL){
        book=temp_list->value;
        if( iniziaPer(inizio_cognome,book.cognome) &&
            iniziaPer(inizio_nome,book.nome) ){
            (*dim)++;
        }
        temp_list=temp_list->next;
    }

    res=(Libro*)malloc(sizeof(Libro)*(*dim));

    temp_list=l;
    *dim=0;
    while(temp_list!=NULL){
        book=temp_list->value;
        if( iniziaPer(inizio_cognome,book.cognome) &&
            iniziaPer(inizio_nome,book.nome) ){
            res[*dim]=book;
            (*dim)++;
        }
        temp_list=temp_list->next;
    }

    return res;
}
```

Fondamenti di Informatica T-1, 2009/2010 – Modulo 2

Prova d'Esame 1A di Martedì 22 Dicembre 2009 – tempo a disposizione 2h30'

"main.c":

```
#include <stdio.h>
#include <stdlib.h>

#include "element.h"
#include "libri.h"
#include "list.h"

int main(void)
{
    list lista_libri, lista_temp, lista_piuLetti;
    Libro book;
    Libro* arrayLibro;
    int dimArray, i, copiePrestate;

    printf("Esercizio 1\n");
    copiePrestate = calcolaCopiePrestate("libri_prestati.txt",
                                        "1231231231231");

    printf("copiePrestate %d\n", copiePrestate);
    printf("\n");

    printf("Esercizio 2\n");
    lista_libri = inventario("libri_posseduti.txt", "libri_prestati.txt");
    lista_temp = lista_libri;
    while(!empty(lista_temp)){
        book = head(lista_temp);
        printf("%s %s %s %d %d\n", book.isbn, book.cognome, book.nome,
                                                    book.posseduti, book.disponibili);
        lista_temp = tail(lista_temp);
    }
    printf("\n");

    printf("Esercizio 3 e 4\n");
    lista_piuLetti = piuLetti(lista_libri);
    lista_temp = lista_piuLetti;
    while(!empty(lista_temp)){
        book = head(lista_temp);
        printf("%s %s %s %d %d\n", book.isbn, book.cognome, book.nome,
                                                    book.posseduti, book.disponibili);
        lista_temp = tail(lista_temp);
    }
    printf("\n");

    printf("Esercizio 5\n");
    arrayLibro = select("Mel", "Pa", lista_piuLetti, &dimArray);
    for(i=0; i<dimArray; i++){
        printf("%s %s %s %d %d\n", arrayLibro[i].isbn, arrayLibro[i].cognome,
                                                    arrayLibro[i].nome, arrayLibro[i].posseduti,
                                                    arrayLibro[i].disponibili);
    }
    printf("\n");

    return (0);
}
```