

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

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.

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

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

Ogni anno viene organizzata una famosa rassegna di musica leggera italiana, dove cantanti famosi (e non) si esibiscono cantando un brano musicale. Nella serata finale della rassegna, i telespettatori da casa possono esprimere la loro preferenza telefonando ad un numero speciale, e indicando il titolo della loro canzone preferita. Il sistema telefonico, completamente automatizzato, memorizza su un file di testo tali informazioni, e in particolare su ogni riga viene salvata una singola preferenza comunicata via telefono. Ogni preferenza (ogni riga) contiene quindi le seguenti informazioni: l'**orario** esatto in cui è cominciata la telefonata (tre interi, formato "hh:mm:ss"); a seguire, separato da uno spazio, il numero di **telefono** del chiamante (una stringa di al più 31 caratteri numerici, con spazi e simbolo '+'); a seguire, separato dal carattere '@', il **titolo** della canzone per cui si esprime la preferenza (una stringa di al più 255 caratteri utili, contenente spazi). Ogni riga, compresa l'ultima, è terminata da un carattere 'newline'.

Non è noto a priori quante righe (cioè, quante preferenze) siano memorizzate nel file. Si noti che i telespettatori (identificati in maniera univoca dal loro numero di telefono) possono esprimere più volte la loro preferenza: sul file vengono memorizzate comunque tutte preferenze espresse. Il sistema poi registra preferenze fino allo scadere della mezzanotte, cioè fino alle 23:59:59.

Si veda, a titolo di esempio, il file di testo "preferenze.txt" fornito nello StartKit.

Esercizio 1 - Strutture dati Timestamp, Preferenza, e funzioni di lett./scritt. (mod. element.h e pref.h/c)

Si definiscano opportune strutture dati **Timestamp**, (per memorizzare ore, minuti e secondi di un istante temporale) e **Preferenza**, al fine di rappresentare una preferenza registrata dal sistema (timestamp, numero telefonico di chi ha chiamato, e titolo della canzone preferita).

Si definisca la funzione:

```
Preferenza leggiPref(FILE * fp);
```

che, ricevuto in ingresso un puntatore ad una struttura dati di tipo **FILE**, legga i dati contenuti su una sola riga e li restituisca tramite una struttura dati di tipo **Preferenza**. Qualora vi siano problemi nella lettura, la funzione deve restituire una preferenza con timestamp pari a 00:00:00.

Si definisca la funzione:

```
Preferenza * leggiTutte(char * fileName, int * dim);
```

che, ricevuto in ingresso il nome di un file, legga da tale file tutte le preferenze e le restituisca tramite un array di strutture dati di tipo **Preferenza** allocato dinamicamente (della dimensione minima necessaria). Tramite il parametro **dim** passato per riferimento, la funzione deve restituire la dimensione del vettore. Qualora la funzione incontri dei problemi nella lettura, o vi siano errori nell'apertura del file, la funzione dovrà stampare un messaggio di errore a video, restituire un puntatore a NULL, e zero come valore della dimensione.

Si definisca la procedura:

```
void stampaPref(Preferenza * v, int dim);
```

che, ricevuto in ingresso un vettore di strutture dati di tipo **Preferenza**, e la sua dimensione **dim**, stampi a video le preferenze.

Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra, avendo cura di deallocare la memoria, se necessario.

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

Esercizio 2 – Ordinamento ed estrazione titoli (moduli element.h/c e pref.h/c)

Il candidato definisca una procedura:

```
void ordina(Preferenza * v, int dim);
```

che, ricevuti in ingresso un vettore di strutture dati di tipo **Preferenza** e la dimensione di tale vettore, ordini il vettore secondo il seguente criterio: in ordine alfabetico (lessicografico) rispetto al numero di telefono della preferenza; a parità di telefono, in ordine crescente in base al timestamp (le telefonate effettuate *prima* vengono *prima* nel vettore). A tal scopo, il candidato utilizzi l'algoritmo di ordinamento "naive sort" visto a lezione.

Si definisca una struttura dati di nome **Canzone**, che contenga due campi: il titolo di una canzone, ed un intero che rappresenterà il numero totale di preferenze espresse per quella canzone.

Si definisca poi una funzione:

```
list filtra(Preferenza * v, int dim);
```

che, ricevuto in ingresso un vettore di strutture dati di tipo **Preferenza**, e la sua dimensione **dim**, restituisca in uscita una lista di strutture dati di tipo **Canzone**, contenente l'elenco delle canzoni, senza ripetizioni. Il campo intero rappresentante il numero totale di preferenze dovrà essere posto a zero.

Esercizio 3 – Conteggio dei voti (modulo pref.h/pref.c)

Lo stesso telespettatore, identificato univocamente dal numero di telefono, può votare più volte. Il regolamento del concorso canoro impone però che venga considerato come valido solo l'ultimo voto espresso in termini temporali. Quindi un telespettatore che vota più volte esprime una e una sola preferenza valida, l'ultima. Le preferenze precedenti non devono essere conteggiate.

Si definisca una funzione:

```
list totali(Preferenza * v, int dim, list titoli);
```

che ricevuti come parametri un vettore di strutture dati di tipo **Preferenza**, la sua dimensione, e una lista di strutture dati di tipo **Canzone**, restituisca una nuova lista di strutture dati **Canzone**, dove sono presenti i titoli passati in ingresso, ma i totali sono stati conteggiati in base alle preferenze espresse nel vettore **v**. Ovviamente, nella conta delle preferenze, bisogna ignorare le preferenze ripetute (fa fede quindi l'ultima preferenza espressa temporalmente dal telespettatore).

Esercizio 4 Stampa dei risultati, e de-allocazione memoria (main.c)

Il candidato realizzi nella funzione **main(...)** un programma che legga dal file delle preferenze tutte le preferenze, e stampi a video la classifica delle canzoni, con il relativo punteggio.

Al termine del programma, il candidato abbia cura di de-allocare tutta la memoria allocata dinamicamente, ivi compresa la memoria allocata per le liste.

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

"element.h":

```
#include <string.h>

#ifndef _ELEMENT_H
#define _ELEMENT_H

#define DIM_TELEFONO 32
#define DIM_TITOLO 256

typedef struct {
    int ore;
    int minuti;
    int sec;
} Timestamp;

typedef struct {
    Timestamp timestamp;
    char telefono[DIM_TELEFONO];
    char titolo[DIM_TITOLO];
} Preferenza;

typedef struct {
    char titolo[DIM_TITOLO];
    int tot;
} Canzone;
typedef Canzone element;

int compare(Preferenza p1, Preferenza p2);
int equals(Canzone c1, Canzone c2);

#endif
```

"element.c":

```
#include "element.h"

int compare(Preferenza p1, Preferenza p2) {
    int result;
    result = strcmp(p1.telefono, p2.telefono);
    if (!result)
        result = p1.timestamp.ore - p2.timestamp.ore;
    if (!result)
        result = p1.timestamp.minuti - p2.timestamp.minuti;
    if (!result)
        result = p1.timestamp.sec - p2.timestamp.sec;
    return result;
}

int equals(Canzone c1, Canzone c2) {
    return strcmp(c1.titolo, c2.titolo)==0;
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

```
"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);

void freelist(list l);
int member(element el, list l);

#endif

"list.c":
#include <stdio.h>
#include <stdlib.h>
#include <string.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 */
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

```
{
  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);
}

void showlist(list l) {
  element temp;
  if (!empty(l)) {
    temp = head(l);
    printf("Titolo: %s, Voti: %d\n",
           temp.titolo, temp.tot);
    showlist(tail(l));
    return;
  }
  else {
    printf("\n\n");
    return;
  }
}

int member(element el, list l) {
  int result = 0;
  while (!empty(l) && !result) {
    result = equals(el, head(l));
    if (!result)
      l = tail(l);
  }
  return result;
}

void freelist(list l) {
  if (empty(l))
    return;
  else {
    freelist(tail(l));
    free(l);
  }
  return;
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

"pref.h":

```
#ifndef _PREF_H
#define _PREF_H

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

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

// Es. 1
Preferenza leggiPref(FILE * fp);
Preferenza * leggiTutte(char * fileName, int * dim);
void stampaPref(Preferenza * v, int dim);

// Es. 2
void ordina(Preferenza * v, int dim);
list filtra(Preferenza * v, int dim);

// Es. 3
list totali(Preferenza * v, int dim, list titoli);
#endif
```

"pref.c":

```
#include "pref.h"

Preferenza leggiPref(FILE * fp) {
    Preferenza result;
    int readOk = 1;
    char ch;
    int i;
    if (readOk)
        readOk = (fscanf(fp, "%d:%d:%d", &(result.timestamp.ore),
&(result.timestamp.minuti), &(result.timestamp.sec) ) == 3);
    if (readOk) {
        ch = fgetc(fp); // skip the blank separator
        i=0;
        ch = fgetc(fp);
        while (i<DIM_TELEFONO-1 && ch!='@') {
            result.telefono[i] = ch;
            i++;
            ch = fgetc(fp);
        }
        result.telefono[i] = '\0';
        if (i==DIM_TELEFONO-1 && ch != '@')
            readOk = 0;
    }
    if (readOk) {
        i=0;
        ch = fgetc(fp);
        while (i<DIM_TITOLO-1 && ch!='\n') {
            result.titolo[i] = ch;
            i++;
            ch = fgetc(fp);
        }
    }
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

```
        result.titolo[i] = '\0';
        if (i==DIM_TITOLO-1 && ch != '\n')
            readOk = 0;
    }
    if (!readOk) {
        result.timestamp.ore = 0;
        result.timestamp.minuti = 0;
        result.timestamp.sec = 0;
    }
    return result;
}
Preferenza * leggiTutte(char * fileName, int * dim) {
    FILE * fp;
    Preferenza * result;
    Preferenza temp;
    int count;

    *dim = 0;
    fp = fopen(fileName, "rt");
    if (fp != NULL) {
        count = 0;
        temp = leggiPref(fp);
        while (temp.timestamp.ore!=00 || temp.timestamp.minuti!=00 ||
temp.timestamp.sec!=00) {
            count++;
            temp = leggiPref(fp);
        }
        rewind(fp);
        result = (Preferenza*) malloc(sizeof(Preferenza) * count);
        count = 0;
        temp = leggiPref(fp);
        while (temp.timestamp.ore!=00 || temp.timestamp.minuti!=00 ||
temp.timestamp.sec!=00) {
            result[count] = temp;
            count++;
            temp = leggiPref(fp);
        }
        *dim = count;
        fclose(fp);
    }
    else {
        result = NULL;
        *dim=0;
    }
    return result;
}
void stampaPref(Preferenza * v, int dim) {
    int i;
    for (i=0; i<dim; i++) {
        printf("%d:%d:%d %s@s\n",
                v[i].timestamp.ore, v[i].timestamp.minuti, v[i].timestamp.sec,
                v[i].telefono, v[i].titolo);
    }
}
void scambia(Preferenza *a, Preferenza *b) {
    Preferenza tmp = *a;
    *a = *b;
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

```
*b = tmp;
}
int trovaPosMax(Preferenza v[], int n){
    int i, posMax=0;

    for (i=1; i<n; i++)
        if (compare(v[posMax],v[i])<0)
            posMax=i;
    return posMax;
}
void naiveSort(Preferenza v[], int n){
    int p;
    while (n>1) {
        p = trovaPosMax(v,n);
        if (p<n-1) scambia(&v[p],&v[n-1]);
        n--;
    }
}
void ordina(Preferenza * v, int dim) {
    naiveSort(v, dim);
}
list filtra(Preferenza * v, int dim) {
    Canzone temp;
    int i;
    list result;

    temp.tot = 0;
    result = emptylist();
    for (i=0; i<dim; i++) {
        strcpy(temp.titolo, v[i].titolo);
        if (!member(temp, result))
            result = cons(temp, result);
    }
    return result;
}
list totali(Preferenza * v, int dim, list titoli) {
    list result;
    Canzone temp;
    int i;
    result = emptylist();
    ordina(v, dim);
    while (!empty(titoli)) {
        temp = head(titoli);
        temp.tot = 0;
        for (i=0; i <dim; i++) {
            if (
                (strcmp(temp.titolo, v[i].titolo)==0 && i==dim-1) ||
                (strcmp(temp.titolo, v[i].titolo)==0
                 && i<dim-1 && strcmp(v[i].telefono, v[i+1].telefono)!=0)
            )
                (temp.tot)++;
        }
        result = cons(temp, result);
        titoli = tail(titoli);
    }
    return result;
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

"main.c":

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

#include "element.h"
#include "pref.h"

int main() {
    { // Es. 1
        Preferenza * v;
        int dim;

        v = leggiTutte("preferenze.txt", &dim);
        stampaPref(v, dim);
        printf("\n\n");
        free(v);
    }

    { // Es. 2
        Preferenza * v;
        int dim;
        list l;

        v = leggiTutte("preferenze.txt", &dim);
        ordina(v, dim);
        stampaPref(v, dim);
        printf("\n\n");
        l = filtra(v, dim);
        showlist(l);
        free(v);
        freelist(l);
    }

    { // Es. 3 && 4
        Preferenza * v;
        int dim;
        list l;
        list tot;

        v = leggiTutte("preferenze.txt", &dim);
        printf("\n\n");
        l = filtra(v, dim);
        tot = totali(v, dim, l);
        showlist(tot);
        free(v);
        freelist(l);
        freelist(tot);
    }
    return 0;
}
```

Fondamenti di Informatica T-1, 2016/2017 – Modulo 2

Prova d'Esame 1A di Giovedì 12 Gennaio 2017 – tempo a disposizione 2h

“preferenze.txt”:

```
21:56:34 +39 051 1112223333@Vincere l'odio
21:56:37 +39 051 2223334444@Vincere l'odio
21:59:03 +39 051 3334445555@Cieli immensi
22:01:00 +39 051 1112223333@Cieli immensi
22:01:13 +39 051 4445556666@Cieli immensi
22:04:45 +39 051 1112223333@Guardando il cielo
22:04:49 +39 051 1112223333@Vincere l'odio
22:29:07 +39 051 5556667777@Vincere l'odio
22:29:45 +39 051 6667778888@Vincere l'odio
```