

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

Prima di cominciare: si scarichi dal sito <http://esamix.labx> il file **StartKit5A.zip** contenente i file necessari (*progetto Visual Studio* ed eventuali altri file di esempio).

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 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*, eseguire di tanto in tanto "Rebuild All".

Una società informatica è stata incaricata dalla Borsa Italiana di sviluppare un software per la rilevazione di manovre speculative nella compravendita di titoli azionari. A tal scopo, viene fornito un file di testo di nome "log.txt", contenente le operazioni di vendita dei titoli azionari. Su ogni riga del file è memorizzata una singola operazione di vendita, e non è noto a priori quante operazioni siano contenute nel file. Ogni operazione di vendita è caratterizzata da un **identificatore** del titolo azionario (una stringa di al più 23 caratteri alfabetici, senza spazi), il **prezzo** di vendita per azione (un intero, rappresentante il valore in centesimi di euro), il **numero** di azioni vendute (un intero), e la **data** in cui è avvenuta l'operazione (tre interi separati da spazi, rappresentanti rispettivamente il giorno, il mese e l'anno). Si veda, a titolo di esempio, il file "log.txt" fornito nello StartKit.

Esercizio 1 - Strutture dati Data, Vendita e funzioni di lettura/scrittura (moduli element.h e azioni.h/azioni.c)

Si definisca un'opportuna struttura dati **Data**, al fine di rappresentare un istante temporale preciso, inteso come tripla giorno/mese/anno. Si definisca poi una struttura dati **Vendita**, al fine di rappresentare una vendita di azioni, come descritto sopra. Per rappresentare l'istante temporale in cui è avvenuta la vendita, si usi una variabile di tipo **Data**, come appena definita.

Si definisca poi la funzione:

```
Vendita * leggiVendite(char* fileName, int *dim);
```

che, ricevuto in ingresso il nome di un file di testo contenente le vendite di azioni, restituisca un array di strutture dati di tipo **Vendita** allocato dinamicamente (della dimensione minima necessaria), contenente tutte le vendite presenti nel file indicato come parametro. Tramite il parametro **dim** la funzione deve restituire la dimensione del vettore. Si presti attenzione al fatto che può accadere (a causa di un bug software) che l'ultima riga del file contenga solo parte delle informazioni relative ad una vendita: in tal caso, tale vendita deve essere scartata e non deve comparire nel vettore restituito come risultato.

Il candidato definisca poi una procedura:

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

che riceva in ingresso un vettore di strutture dati di tipo **Vendita** e la dimensione di tale vettore, e ordini il vettore secondo un ordine lessicografico sul nome del titolo azionario. A parità di nome, le vendite devono essere ordinate in base alla data in cui sono state eseguite, in ordine crescente. Si usi a tal scopo un algoritmo di ordinamento a scelta tra quelli visti a lezione.

Si definisca infine una procedura:

```
void stampaVendite(Vendita * v, int dim);
```

che, ricevuta in ingresso un vettore di strutture dati di tipo **Vendita**, e la sua dimensione, stampi a video il contenuto di tale vettore.

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

Esercizio 2 – Ricerca delle speculazioni (modulo azioni.h/azioni.c)

Si definisca poi una procedura:

```
void speculazioni(Vendita * v, int dim);
```

che, ricevuta come parametro un vettore di strutture dati di tipo **Vendita** ordinato come specificato nell'esercizio 1, e la sua dimensione **dim**, stampi a video il nome dei titoli che sono stati oggetto di speculazione. Un titolo è stato oggetto di speculazione se, nello stesso mese, è stato oggetto di almeno 3 vendite. Ad esempio, considerando il file fornito nello StartKit, risulta che il titolo di nome "Unicredit" è stato oggetto di speculazione, poiché nel mese di luglio 2011 è stato venduto 5 volte.

Esercizio 3 – Vendite con importi maggiori (modulo azioni.h/azioni.c)

Si realizzi una funzione:

```
list selezionaVenditeMaggiori(Vendita * v, int dim);
```

che, ricevuto in ingresso un vettore **v** di strutture dati di tipo **Vendita**, ordinate secondo quanto richiesto nell'esercizio 1, e la dimensione **dim** di tale vettore, restituisca una lista di strutture dati di tipo **Vendita**. In particolare, la lista dovrà contenere, per ogni titolo azionario presente nel vettore **v**, la vendita di importo risultante maggiore. L'importo della vendita è definito come il prodotto del prezzo di ogni singola azione per il numero di azioni vendute.

Si realizzi una procedura:

```
void stampaListaVendite(list l);
```

che, ricevuta in ingresso una lista di strutture dati di tipo **Vendita**, stampi a video il contenuto di tale lista.

Esercizio 4 – Stampa delle speculazioni, delle vendite di importo maggiore, e de-allocazione memoria (main.c)

Il candidato realizzi nella funzione **main(...)** un programma che, usando le informazioni fornite tramite il file di esempio "log.txt", stampi a video l'elenco delle vendite di importi maggiori, e i nomi dei titoli che sono stati soggetti a speculazione, secondo quanto definito negli esercizi precedenti.

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, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

```
"element.h":
#ifndef _ELEMENT_H
#define _ELEMENT_H

#define DIM_NOME 24

typedef struct {
    int giorno;
    int mese;
    int anno;
} Data;

typedef struct {
    char nome[DIM_NOME];
    int prezzo;
    int numero;
    Data data;
} Vendita;

typedef Vendita element;
int compare(Vendita v1, Vendita v2);

#endif /* _ELEMENT_H */

"element.c":
#include "element.h"

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

int compare(Vendita v1, Vendita v2) {
    int temp;

    temp = strcmp(v1.nome, v2.nome);
    if (temp == 0)
        temp = v1.data.anno - v2.data.anno;
    if (temp == 0)
        temp = v1.data.mese - v2.data.mese;
    if (temp == 0)
        temp = v1.data.giorno - v2.data.giorno;
    return temp;
}
```

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – 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);
#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); }

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

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

"azioni.h":

```
#ifndef AZIONI
#define AZIONI

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

Vendita * leggiVendite(char* fileName, int *dim);
void ordina(Vendita v[], int n);
void stampaVendite(Vendita * v, int dim);

list selezionaVenditeMaggiori(Vendita * v, int dim);
void stampaListaVendite(list l);

void speculazioni(Vendita * v, int dim);

#endif
```

"azioni.c":

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

Vendita * leggiVendite(char* fileName, int *dim) {
    Vendita * result;
    Vendita temp;
    int i;
    FILE * fp;

    fp = fopen(fileName, "rt");
    if (fp == NULL) {
        printf("Errore aprendo il file %s\n", fileName);
        system("pause");
        exit(-1);
    }
    else {
        *dim = 0;
        while(fscanf(fp, "%s %d %d %d %d", temp.nome,
                    &temp.prezzo, &temp.numero, &temp.data.giorno,
                    &temp.data.mese, &temp.data.anno) == 6)
            (*dim)++;

        rewind(fp);
        result = (Vendita *) malloc(sizeof(Vendita) * *dim);
        i = 0;
        while(fscanf(fp, "%s %d %d %d %d", temp.nome,
                    &temp.prezzo, &temp.numero, &temp.data.giorno,
                    &temp.data.mese, &temp.data.anno) == 6) {
            result[i] = temp;
            i++;
        }
    }
}
```

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

```
        fclose(fp);
        return result;
    }
    return result; // irraggiungibile, messo solo per evitare un warning
}

void scambia(Vendita *a, Vendita *b) {
    Vendita tmp = *a;
    *a = *b;
    *b = tmp;
}

// bubble sort
void ordina(Vendita v[], int n) {
    int i, ordinato = 0;
    while (n>1 && !ordinato) {
        ordinato = 1;
        for (i=0; i<n-1; i++)
            if (compare(v[i],v[i+1])>0) {
                scambia(&v[i],&v[i+1]);
                ordinato = 0;
            }
        n--;
    }
}

void stampaVendite(Vendita * v, int dim) {
    int i;
    printf("Vendite effettuate:\n");
    for (i=0; i<dim; i++)
        printf("%s %d %d %d/%d/%d\n",
            v[i].nome, v[i].prezzo, v[i].numero, v[i].data.giorno, v[i].data.mese,
            v[i].data.anno);
    printf("\n\n");
}

list selezionaVenditeMaggiori(Vendita * v, int dim) {
    int i;
    list result;
    Vendita temp;

    result = emptylist();

    if (dim>0) {
        temp = v[0];
        if (dim == 1)
            result = cons(temp, result);
        else {
            i=1;
            while (i<dim) {
                while (i<dim && strcmp (temp.nome, v[i].nome) == 0) {
                    if (v[i].prezzo*v[i].numero > temp.prezzo*temp.numero)
                        temp = v[i];
                    i++;
                }
                result = cons(temp, result);
                temp = v[i];
                if (i == dim-1)

```

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

```
                result = cons(temp, result);
                i++;
            }
        }
    }
    return result;
}

void stampaListaVendite(list l) {
    Vendita temp;
    while (!empty(l)) {
        temp = head(l);
        printf("%s %d %d %d/%d/%d\n",
            temp.nome, temp.prezzo, temp.numero, temp.data.giorno, temp.data.mese,
temp.data.anno);
        l = tail(l);
    }
    printf("\n\n");
    return;
}

void speculazioni(Vendita * v, int dim) {
    int i;
    Vendita temp;
    int count;

    count = 0;
    for (i=0; i<dim; i++) {
        if (i==0) {
            temp = v[i];
            count++;
        }
        else {
            if (strcmp(temp.nome, v[i].nome)==0 &&
                v[i].data.anno==temp.data.anno &&
                v[i].data.mese==temp.data.mese)
                count++;
            else {
                if (count>= 3)
                    printf("Speculazione su \"%s\", nel mese %d/%d\n",
temp.nome, temp.data.mese, temp.data.anno);
                count = 0;
                temp = v[i];
            }
        }
    }
    return;
}
```

Fondamenti di Informatica T-1, 2010/2011 – Modulo 2

Prova d'Esame 5A di Mercoledì 13 Luglio 2011 – tempo a disposizione 2h

"main.c":

```
#include "element.h"
#include "azioni.h"

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

int main() {

    Vendita * elenco;
    int dim;
    list l;

    elenco = leggiVendite("log.txt", &dim);
    stampaVendite(elenco, dim);
    ordina(elenco, dim);
    stampaVendite(elenco, dim);
    system("pause");

    speculazioni(elenco, dim);
    system("pause");

    l = selezionaVenditeMaggiori(elenco, dim);
    stampaListaVendite(l);

    freelist(l);
    free(elenco);
    system("pause");

    return 0;
}
```

"log.txt":

```
Unicredit 320 15000 3 7 2011
Telecom 910 1000 6 6 2010
Seat 356 35000 4 5 2006
Telecom 810 980 7 6 2010
Intesa 467 23000 3 7 2011
Unicredit 315 11500 4 7 2011
Unicredit 303 23000 5 7 2011
Unicredit 298 56000 5 7 2011
Unicredit 287 43000 6 7 2011
Vodafone 354 550000 8 7 2011
Intesa 456 24600
```