

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

Prima di cominciare: si scarichi dal sito <http://esamix.labx> il file **StartKit4A.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 specializzata nel gestire scommesse sta realizzando il nuovo portale Web, tramite il quale gli utenti potranno effettuare scommesse online. Ogni scommessa è caratterizzata da un insieme di informazioni, tra cui il **codice identificativo unico del cliente** (un intero), il **codice identificativo della scommessa** (una stringa ben formata di esattamente 15 caratteri alfanumerici, senza spazi), l'**importo** (un float), e la **data** in cui è stata effettuata la scommessa (tramite tre interi rappresentanti l'anno, il mese e il giorno). Tutte le scommesse effettuate sono registrate in un file di testo di nome *"registro.txt"*, una scommessa in ogni riga, i valori separati da spazi. Non è noto a priori quante scommesse siano registrate nel file. Si veda a tal proposito il file *"registro.txt"* fornito come esempio nello StartKit.

In un secondo file di testo di nome *"anagrafe.txt"* si tiene traccia del **nome** (una stringa di al più 23 caratteri, senza spazi), del **cognome** (una stringa di al più 23 caratteri, senza spazi), e il **codice identificativo unico cliente** (un intero): ogni cliente è memorizzato su una riga del file, e non è noto a priori quante righe siano memorizzate nel file.

Esercizio 1 - Struttura dati Scommessa e funzioni di lettura/scrittura (moduli `element.h` e `scommesse.h/scommesse.c`)

Si definisca un'opportuna struttura dati **Scommessa**, al fine di rappresentare i dati relativi ad ogni scommessa, come descritto in precedenza.

Si definisca poi la funzione:

```
Scommessa * leggiScommesse(char* fileName, int *dim);
```

che, ricevuto in ingresso il nome di un file di testo contenente le scommesse, restituisca un array di strutture dati di tipo **Scommessa** allocato dinamicamente (non necessariamente della dimensione minima), contenente tutte le scommesse presenti nel file indicato come parametro. Tramite il parametro **dim**, la funzione deve restituire la dimensione del vettore. Siccome per errore ogni tanto vengono inserite due scommesse con lo stesso codice identificativo di scommessa, la funzione deve avere cura di scartare tali ripetizioni. Ai fini della lettura quindi la funzione deve effettivamente memorizzare ogni "prima" scommessa, e scartare eventuali ripetizioni. A tal scopo, il candidato definisca una funzione:

```
int giaPresente(Scommessa * v, int dim, Scommessa s);
```

che riceva in ingresso un vettore di strutture dati di tipo **Scommessa** e la dimensione di tale vettore, ed una scommessa particolare **s**. La funzione deve restituire un valore interpretabile come "vero" se la scommessa **s** è già presente nel vettore, falso altrimenti.

Si definisca infine una procedura:

```
void stampaScommesse(Scommessa * v, int dim);
```

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

Il candidato infine abbia cura di inserire nel **main(...)** alcune istruzioni per "testare" la corretta implementazione delle funzioni, usando come prova il file di esempio fornito nello StartKit, e de-allocando

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

eventuale memoria allocata dinamicamente. Tali istruzioni per il test devono rimanere, commentate, nella funzione main.

Esercizio 2 – Lettura dei clienti (modulo element.h/element.c e scommesse.h/scommesse.c)

Si definisca un'opportuna struttura dati **Cliente** al fine di rappresentare ogni utente del sistema, come descritto in precedenza. Inoltre la struttura deve essere predisposta per memorizzare anche tutte le scommesse effettuate dal singolo cliente, in termini di strutture dati di tipo **Scommessa**. Ovviamente il numero delle scommesse effettuate cambia da cliente a cliente, e non è noto a priori.

Si realizzi una funzione:

```
Scommessa * seleziona(Scommessa v[], int dim, int codCliente, int *dimRes);
```

che, ricevuti in ingresso un vettore di strutture dati di tipo **Scommessa**, la sua dimensione **dim**, e il codice identificativo del cliente, restituisca un nuovo vettore allocato dinamicamente (della dimensione minima necessaria) contenente tutte e sole le scommesse in **v** ed effettuate dal cliente indicato. Tramite il parametro **dimRes** la funzione deve restituire la dimensione del nuovo vettore. Il vettore restituito inoltre dovrà presentare le scommesse ordinate in maniera crescente secondo la data.

Si realizzi poi una funzione:

```
list leggiClienti(char * fileName, Scommessa v[], int dim);
```

che, ricevuto in ingresso il nome di un file contenente l'elenco dei clienti, un vettore **v** contenente tutte le scommesse effettuate, e la dimensione **dim** di tale vettore, restituisca una lista contenente strutture dati di tipo **Cliente**. Al fine di determinare le scommesse effettuate dal singolo cliente, il candidato utilizzi le funzioni di cui all'esercizio 1 per leggere tutte le scommesse effettuate, e la funzione sopra descritta per determinare le scommesse effettuate dal singolo cliente.

Esercizio 3 – Stampa dei giocatori assidui (modulo prodotti.h/prodotti.c)

Si definisca poi una procedura:

```
void stampaAssidui(list l);
```

che, ricevuta come parametro una lista di strutture dati di tipo **Cliente**, stampi a video tutti i dati riguardanti i clienti che sono giocatori "assidui". Un cliente è da considerarsi "giocatore assiduo" se in uno stesso anno solare questi ha effettuato almeno 3 scommesse. Ad esempio, considerando i file forniti nello StartKit, risulta che l'unico giocatore assiduo è il giocatore con codice identificativo cliente "12", che nel 2011 ha effettuato tre scommesse.

Esercizio 4 – Determinazione dei giocatori assidui e deallocazione memoria (main.c)

Il candidato realizzi nella funzione **main(...)** un programma che, usando le informazioni fornite tramite i file "registro.txt" e "anagrafe.txt", stampi a video l'elenco dei giocatori assidui, 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. In particolare, all'atto di deallocazione delle liste contenenti le strutture dati di tipo **Cliente**, il candidato abbia cura di deallocare anche l'elenco delle scommesse relative ad ogni cliente.

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

"element.h":

```
#ifndef _ELEMENT_H
#define _ELEMENT_H
#define DIM_COD_SCOMMESSA 16
#define DIM_NOME 23

typedef struct {
    int cic;
    char cis[DIM_COD_SCOMMESSA];
    float importo;
    int giorno;
    int mese;
    int anno;
} Scommessa;

typedef struct {
    char nome[23];
    char cognome[23];
    int cic;
    Scommessa * elenco;
    int dim;
} Cliente;

typedef Cliente element;
int equals(Scommessa s1, Scommessa s2);
int compare(Scommessa s1, Scommessa s2);

#endif /* _ELEMENT_H */
```

"element.c":

```
#include "element.h"
#include <stdio.h>
#include <string.h>

int equals(Scommessa s1, Scommessa s2) {
    if (strcmp(s1.cis, s2.cis) == 0)
        return 1;
    else
        return 0;
}

int compare(Scommessa s1, Scommessa s2) {
    int temp;
    temp = s1.anno - s2.anno;
    if (temp == 0)
        temp = s1.mese - s2.mese;
    if (temp == 0)
        temp = s1.giorno - s2.giorno;
    return temp;
}
```

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

Prova d'Esame 4A di Martedì 21 Giugno 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 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

"scommesse.h":

```
#ifndef SCOMMESSE
#define SCOMMESSE

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

int giaPresente(Scommessa * v, int dim, Scommessa s);
Scommessa * leggiScommesse(char* fileName, int *dim);
void stampaScommesse(Scommessa * v, int dim);

Scommessa * seleziona(Scommessa v [], int dim, int codCliente, int *dimRes);
void bubbleSort(Scommessa v[], int n);
list leggiClienti(char * fileName, Scommessa v[], int dim);

void stampaAssidui(list l);

#endif
```

"scommesse.c":

```
#include "scommesse.h"

int giaPresente(Scommessa * v, int dim, Scommessa s) {
    int i;
    int trovato = 0;
    for (i=0; i<dim && !trovato; i++)
        if (equals(v[i], s))
            trovato = 1;
    return trovato;
}

Scommessa * leggiScommesse(char* fileName, int *dim) {
    FILE * fp;
    Scommessa * result;
    Scommessa temp;
    int maxDim;

    *dim = 0;
    fp = fopen(fileName, "rt");
    if (fp == NULL) {
        printf("Errore nell'aprire il file %s\n Programma terminato...\n",
        fileName);
        system("pause");
        exit(-1);
    }
    else {
```

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

```
maxDim = 0;
while (fscanf(fp, "%d %s %f %d %d %d",
             &temp.cic, temp.cis, &temp.importo,
             &temp.giorno, &temp.mese, &temp.anno) == 6)
    maxDim++;

if (maxDim > 0) {
    result = (Scommessa *) malloc(sizeof(Scommessa) * maxDim);

    rewind(fp);
    while (fscanf(fp, "%d %s %f %d %d %d",
                 &temp.cic, temp.cis, &temp.importo,
                 &temp.giorno, &temp.mese, &temp.anno) == 6) {
        if (!giaPresente(result, *dim, temp)) {
            result[*dim] = temp;
            (*dim)++;
        }
    }
}

fclose(fp);
return result;
}

void stampaScommesse(Scommessa * v, int dim) {
    int i;
    for (i=0; i<dim; i++) {
        printf("%d %s %f %d %d %d\n",
              v[i].cic, v[i].cis, v[i].importo,
              v[i].giorno, v[i].mese, v[i].anno);
    }
    printf ("\n");
    return;
}

Scommessa * seleziona(Scommessa v [], int dim, int codCliente, int *dimRes) {
    Scommessa * result;
    int i;
    int cont;
    *dimRes = 0;
    for (i=0; i<dim; i++)
        if (v[i].cic == codCliente)
            (*dimRes)++;
    result = (Scommessa *) malloc(sizeof(Scommessa) * *dimRes);
    cont = 0;
    for (i=0; i<dim; i++)
        if (v[i].cic == codCliente) {
            result[cont] = v[i];
            cont++;
        }
}
```

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

```
    bubbleSort(result, *dimRes);
    return result;
}

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

void bubbleSort(Scommessa 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--;
    }
}

list leggiClienti(char * fileName, Scommessa v[], int dim) {
    list result;
    FILE * fp;
    Cliente temp;
    result = emptylist();
    fp = fopen(fileName, "rt");
    if (fp == NULL) {
        printf("Impossibile aprire il file %s\n Exiting...\n", fileName);
        system("pause");
        exit(-2);
    }
    else {
        while (fscanf(fp, "%s %s %d", temp.nome, temp.cognome, &temp.cic) ==
3) {
            temp.elenco = seleziona(v, dim, temp.cic, &temp.dim);
            result = cons(temp, result);
        }
        fclose(fp);
    }
    return result;
}

void stampaAssidui(list l) {
    Cliente temp;
    int cont;
    int currentYear;
    int assiduo;
    int i, j;
    while (!empty(l)) {
```

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

```
temp = head(1);
assiduo = 0;
for (i=0; i<temp.dim && !assiduo; i++) {
    currentYear = temp.elenco[i].anno;
    cont = 0;
    for (j=0; j<temp.dim && !assiduo; j++) {
        if (temp.elenco[j].anno == currentYear)
            cont++;
        if (cont>=3)
            assiduo = 1;
    }
}
if (assiduo)
    printf("%s %s e' un giocatore assiduo!\n", temp.nome,
temp.cognome);
l = tail(1);
}
return;
}
```

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

"main.c":

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "element.h"
#include "scommesse.h"

int main() {

    // TEST Es. 1
    {
        Scommessa * v;
        int dim;

        v = leggiScommesse("registro.txt", &dim);
        stampaScommesse(v, dim);
        free(v);
        system("pause");
    }
    // TEST Es. 2, 3 e 4
    {
        Scommessa * v;
        list lClienti;
        list temp;
        Cliente tCliente;
        int dim;
        int i;
        v = leggiScommesse("registro.txt", &dim);
        lClienti = leggiClienti("anagrafe.txt", v, dim);

        temp = lClienti;
        while (!empty(temp)) {
            tCliente = head(temp);
            printf("%s %s:\n", tCliente.nome, tCliente.cognome);
            for (i=0; i<tCliente.dim; i++)
                printf("\t %d %s %f %d %d %d\n",
                    tCliente.elenco[i].cic, tCliente.elenco[i].cis,
                    tCliente.elenco[i].importo, tCliente.elenco[i].giorno,
                    tCliente.elenco[i].mese, tCliente.elenco[i].anno);
            printf("\n\n");
            temp = tail(temp);
        }
        stampaAssidui(lClienti);
        while (!empty(lClienti)) {
            temp = lClienti;
            tCliente = head(temp);
            free(tCliente.elenco);
            lClienti = tail(lClienti);
            free(temp);
        }
    }
}
```

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

Prova d'Esame 4A di Martedì 21 Giugno 2011 – tempo a disposizione 2h

```
        free(v);  
        system("pause");  
    }  
    return 0;  
}
```

"registro.txt":

```
12 12345ABCDEF0013 5.00 15 06 2009  
12 12345ABCDEF0015 15.00 16 06 2011  
14 12345ABCDEF0018 21.00 21 06 2011  
12 12345ABCDEF0013 8.00 21 06 2011  
17 12345ABCDEF0021 10.00 14 06 2011  
17 12345ABCDEF0021 4.00 15 06 2011  
12 12345ABCDEF0013 3.00 16 06 2011  
14 12345ABCDEF0024 20.00 21 06 2011  
12 12345ABCDEF0027 9.00 21 06 2011  
12 12345ABCDEF0028 9.00 21 06 2011
```

"anagrafe.txt":

```
Federico Chesani 12  
Carlo Giannelli 14  
Paola Mello 17  
Stefano Bragaglia 18
```