

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

Prima di cominciare: si scarichi il file `StartKit1A.zip` contenente i file di esempio.

Avvertenze per la consegna: nominare i file sorgenti come richiesto nel testo del compito, 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** ed i file contenuti nello StartKit.

Rispettare le specifiche, in particolare inserire le funzioni nei file specificati fra parentesi dopo il nome della funzione. Chi non rispetta le specifiche sarà opportunamente penalizzato. **NON SARANNO CORRETTI** gli elaborati che presenteranno un numero "non ragionevole" di errori di compilazione.

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

Si deve realizzare un programma che permetta di gestire le informazioni relative ad un insieme di carte di credito ed ai pagamenti effettuati, individuando potenziali truffe. Le informazioni sono contenute in due differenti file, denominati rispettivamente `clienti.txt` e `pagamenti.txt`.

Il primo file memorizza le informazioni "statiche" relative ai clienti, ovvero **CODICE** della carta di credito (una stringa di 16 caratteri), **NOME** del proprietario (al più 63 caratteri senza spazi), **DATA DI SCADENZA** della carta; i campi sono separati da uno spazio. Le date di scadenza sono memorizzate tramite due interi, separati da uno spazio, nel formato "**MM AAAA**".

Il secondo file memorizza le informazioni "dinamiche" relative ai pagamenti effettuati con le carte di credito. Il file, per ogni riga, memorizza la **DATA DI PAGAMENTO**, il **CODICE** della carta utilizzata, l'**IMPORTO** (un numero con virgola), la **VALUTA** (una stringa di 3 caratteri); le varie informazioni sono separate dal carattere spazio. Le date di pagamento sono memorizzate tramite tre interi, nella forma "**GG MM AAAA**".

Esercizio 1 - Gestione dei clienti (`clienti.h/clienti.c`)

Il candidato realizzi un modulo che definisca una struttura dati opportuna, di nome `client`, per rappresentare un cliente, e definisca le funzioni per (a) leggere da file i clienti e (b) visualizzare a video le informazioni relative ad un cliente.

La funzione di lettura riceve come parametri di ingresso il nome di un file (da cui andare a leggere) e deve restituire un array di clienti allocato dinamicamente (si abbia cura di allocare solo ed unicamente lo spazio strettamente necessario); inoltre la dimensione del vettore deve essere restituita tramite un ulteriore apposito parametro in ingresso. Il candidato valuti l'eventuale utilizzo della funzione `rewind(...)`.

```
Client * readAllClient(char * fileName, int * dim);
```

La procedura di stampa invece riceve come parametro di ingresso una struttura dati di tipo `client`, e stampa a video tutti i dati relativi al cliente.

```
void printClient(Client c);
```

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni definite al fine di verificarne la corretta implementazione: usi a tale scopo il file `clienti.txt` fornito nello starter kit. Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

Esercizio 2 - Gestione dei pagamenti (`clienti.h/clienti.c`)

Il candidato definisca una opportuna struttura dati `spesa` per rappresentare un pagamento, e definisca opportune funzioni per (a) leggere da file i pagamenti relativi ad una certa carta (identificata dal suo codice), e (b) visualizzare le informazioni relative ad un pagamento.

La funzione di lettura, riceve in ingresso il nome del file ed il codice di una carta, e restituisce un array di strutture di tipo `spesa` con tutti i pagamenti effettuati con quella carta. Tramite un intero passato per riferimento, la funzione restituisce anche la dimensione di tale array.

```
Spesa * readSpesa(char * fileName, char * codice, int * dim);
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

La procedura di stampa, invece, riceve in ingresso una struttura dati `spesa` e la visualizza a video.

```
void printSpesa (Spesa s);
```

Il modulo deve inoltre fornire una funzione

```
int compareSpese (Spesa s1, Spesa s2);
```

che restituisca -1, 0 o 1 a seconda che `s1` sia avvenuta un giorno precedente a `s2`, lo stesso giorno o un giorno successivo, rispettivamente.

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni definite, al fine di verificarne la corretta implementazione: usi a tale scopo il file `pagamenti.txt` fornito nello starter kit. Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

Esercizio 3 - Gestione delle truffe (clienti.h/clienti.c)

Data la lista delle spese effettuate da una singola carta di credito, è necessario analizzare tale elenco al fine di identificare eventuali truffe. Si implementi la funzione:

```
int findTruffa (Spesa v[], int dim);
```

che ricevuti in ingresso un array di spese effettuate con una certa carta, e la dimensione di tale array, stampi a video le spese "sospette". In particolare, una spesa è da considerarsi sospetta qualora lo stesso giorno ne sia stata effettuata un'altra con valuta diversa. Una volta individuata una coppia di spese sospette (cioè due spese avvenute lo stesso giorno ma con valute diverse), esse devono essere entrambe stampate a video al fine di permettere all'utente di analizzare i dati. La funzione restituisce il numero di coppie di pagamenti sospetti incontrati.

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni definite, al fine di verificarne la corretta implementazione. Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

Esercizio 4 - Main (main.c)

Il candidato realizzi un programma `main.c` che provveda a caricare l'elenco dei clienti registrati in `clienti.txt`, e poi, per ogni cliente, acceda al file `pagamenti.txt` e stampi a video tutte le spese che potrebbero essere oggetto di truffa. Dopo aver stampato le spese sospette, il programma stampi anche i dati della carta di credito ed il numero di coppie di spese sospette.

Il candidato abbia cura di deallocare (al termine del programma) tutte le strutture allocate dinamicamente.

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

"clienti.h":

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

#ifndef CLIENT
#define CLIENT
#define DIM_CODICE 17
#define DIM_NOME 64

typedef struct {
    char codice[DIM_CODICE];
    char nome[DIM_NOME];
    int mese;
    int anno;
} Client;

#endif

#ifndef SPESA
#define SPESA
#define DIM_VALUTA 4

typedef struct {
    int giorno;
    int mese;
    int anno;
    char codice[DIM_CODICE];
    float importo;
    char valuta[DIM_VALUTA];
} Spesa;

#endif

Client * readAllClient(char * fileName, int * dim);
void printClient(Client c);
Spesa * readSpesa(char * fileName, char * codice, int * dim);
void printSpesa(Spesa s);
int compareSpese(Spesa s1, Spesa s2);
int findTruffa(Spesa v[], int dim);
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

"clienti.c":

```
#include "clienti.h"

/*
LETTURA/SCRITTURA di Client
*/

Client * readAllClient(char * fileName, int * dim) {
    int count, i;
    FILE * fp;
    Client * result;
    Client temp;

    if ((fp=fopen(fileName, "r")) == NULL)
        exit(1);
    count = 0;
    while (fscanf(fp, "%s %s %d %d",
                  temp.codice, temp.nome,
                  &(temp.mese), &(temp.anno)) == 4)
        count++;
    rewind(fp);

    result = (Client *) malloc(count * sizeof(Client));
    for (i=0; i<count; i++)
        fscanf(fp, "%s %s %d %d",
              result[i].codice, result[i].nome,
              &(result[i].mese), &(result[i].anno));

    fclose(fp);
    *dim = count;
    return result;
}

void printClient(Client c) {
    printf("%s Carta: %s Scadenza: %d/%2d\n",
          c.nome, c.codice, c.mese, c.anno);
}

/*
LETTURA e SCRITTURA di Spesa
*/

Spesa * readSpesa(char * fileName, char * codice, int * dim) {
    int count, i;
    FILE * fp;
    Spesa * result;
    Spesa temp;

    if ((fp=fopen(fileName, "r")) == NULL)
        exit(1);
    count = 0;
    while (fscanf(fp, "%d %d %d %s %f %s",
                  &(temp.giorno), &(temp.mese), &(temp.anno),
                  temp.codice, &(temp.importo), temp.valuta) == 6)
        if (strcmp(codice, temp.codice) == 0)
            count++;
    rewind(fp);
}
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

```
result = (Spesa *) malloc(count * sizeof(Spesa));
i=0;
while (i<count) {
    fscanf(fp, "%d %d %d %s %f %s",
           &(result[i].giorno), &(result[i].mese), &(result[i].anno),
           result[i].codice, &(result[i].importo), result[i].valuta);
    if (strcmp(codice, result[i].codice) == 0)
        i++;
}
fclose(fp);
*dim = count;
return result;
}

void printSpesa(Spesa s) {
    printf("Carta: %s Importo: %f%s Data: %d/%d/%d\n",
           s.codice, s.importo, s.valuta, s.giorno, s.mese, s.anno);
}

int compareSpese(Spesa s1, Spesa s2) {
    if (s1.anno != s2.anno)
        if (s1.anno<s2.anno) return -1;
        else return 1;
    else {
        if (s1.mese!=s2.mese)
            if (s1.mese<s2.mese) return -1;
            else return 1;
        else {
            if (s1.giorno<s2.giorno) return -1;
            else {
                if (s1.giorno==s2.giorno) return 0;
                else return 1;
            }
        }
    }
}

int findTruffa(Spesa v[], int dim) {
    int i, j;
    int result = 0;

    for (i=0; i<dim; i++) {
        for (j=i+1; j<dim; j++) {
            if (compareSpese(v[i], v[j]) == 0)
                if (strcmp(v[i].valuta, v[j].valuta)) {
                    printf("Spese sospette:\n");
                    printSpesa(v[i]);
                    printSpesa(v[j]);
                    result++;
                }
        }
    }
    return result;
}
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 12 Dicembre 2008
Compito A

"main.c":

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

int main(void) {
    Client * v;
    int dimv;
    Spesa * s;
    int dims;
    int i;
    int num;

    /* test es 1*/
    v = readAllClient("clienti.txt", &dimv);
    for (i=0; i<dimv; i++)
        printClient(v[i]);

    /* test es 2
    s = readSpesa("spese.txt", "1234567812345678", &dims);
    for (i=0; i<dims; i++)
        printSpesa(s[i]);
    printf("%d\n", compareSpese(s[0], s[1]));
    */

    /* test es 3
    printf("%d\n", findTruffa(s, dims));
    */

    /* es 4*/
    for (i=0; i<dimv; i++) {
        s = readSpesa("pagamenti.txt", v[i].codice, &dims);
        num = findTruffa(s, dims);
        if (num>0) {
            printf("Dati della carta di credito:\n");
            printClient(v[i]);
            printf("Coppie di transazioni sospette: %d\n", num);
        }
        free(s);
    }
    free(v);
    system("PAUSE");
    return (0);
}
```