

Fondamenti di Informatica L-A (A.A. 2008/2009) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d’Esame di Venerdì 13 Febbraio 2009 – durata 2h
Compito A
Compito Modalità Fondamenti di Informatica L-A (solo a.a. precedenti)

ESERCIZIO 1 (10 punti)

Un venditore di ortofrutta è solito fare credito ai propri clienti. Al fine di tenere traccia dei crediti, utilizza una funzione del registratore di cassa che salva, su un file binario di nome “**log.dat**”, strutture dati del tipo **transaction** contenenti i seguenti dati:

- una stringa **customer**, contenente il nome del cliente (al più 127 caratteri, senza spazi);
- un intero **transactionId**, recante un codice identificativo della transazione;
- un float **value**, contenente l'ammontare del credito concesso.

Al fine di riscuotere i crediti, il commerciante deve però poter accedere ai valori registrati nel file binario. In particolare, è interessato a riscuotere tutti gli scontrini superiori ad una certa cifra. A tal scopo, egli vuole avere un programma che, richiesta una cifra come soglia, scriva su un file in formato testo di nome “**debitori.txt**” tutti gli scontrini il cui importo è maggiore al valore specificato. Si realizzi:

1. una funzione **int select (...)** che, ricevuti in ingresso un puntatore **source** al file binario, un puntatore **dest** al file di testo, e un float **threshold** passato per riferimento, copi su **dest** tutte le **transaction** presenti in **source** e di importo superiore al parametro **threshold**. La funzione deve restituire il numero di scontrini selezionati.
2. un programma C che chieda inizialmente all’utente un importo di soglia. Dopo aver aperto i file nell’opportuna modalità di lettura/scrittura, il programma utilizzi la funzione **select (...)** definita al punto precedente per filtrare i dati. Il programma stampi infine a video il numero totale di scontrini (**transaction**) che sono stati copiati da un file all’altro.

ESERCIZIO 2 (8punti)

Si scriva una funzione iterativa:

```
int isAdd(list l1, list l2, list add);
```

che riceve in ingresso tre liste di interi, determini se la lista **add** contiene in ogni elemento la somma degli elementi corrispondenti della lista **l1** e la lista **l2**. In particolare la funzione deve restituire un valore maggiore di 0 se effettivamente vale questa proprietà; deve ritornare 0 altrimenti. Si supponga per semplicità che le liste **l1** e **l2** abbiano la stessa lunghezza. Non si può supporre invece questo per la lista **add**.

Ad esempio, date le liste **l1=[1,2,3,4]**, **l2=[1,4, 3, 0]**, **add=[2,6,6,4]**, la funzione deve restituire valore 1. Invece, se invocata con **l1=[1,2,3,4]**, **l2=[1,4,5,7]**, **add=[3,2,0,1]**, oppure **add=[2,6,8]** o **add=[2,6,8,11,20]** allora la funzione deve restituire 0.

La funzione deve essere realizzata utilizzando il tipo di dato astratto **list**, definito per gli interi (non è necessario riportare la definizione nella soluzione). Si possono utilizzare le sole operazioni primitive definite durante il corso, che quindi possono NON essere riportate nella soluzione. Non si possono usare altre funzioni di alto livello.

ESERCIZIO 3 (5 punti)

Data la funzione:

```
int p1(int x, int y) {
    float a = x;

    if (a<y)
        return a / 2 + 0.5;
    else
        return p1( a, y+1) + p1( a, y+1.5);
}
```

mostrare la sequenza dei record di attivazione nel caso in cui la funzione sia invocata con parametri attuali (4, 3) e il valore di ritorno. Si ignorino eventuali warning che potrebbero essere generati in fase di compilazione.

ESERCIZIO 4 (5 punti)

Si introduca il metodo di ordinamento di un vettore “a bolle” (bubble sort) e poi si scriva lo (pseudo) codice di una funzione che lo realizza.

ESERCIZIO 5 (4 punti)

Si consideri il seguente codice:

```
void cambia(int v[][2]) {
    v[1][1] = 10;
}

int main(void)
{
    int a[2][3] = {{1,2,3}, {4,5,6}};
    int i, j;

    cambia(a);
    for (i=0; i<2; i++) {
        for (j=0; j<3; j++)
            printf("%d ", a[i][j]);
        printf("\n");
    }
    return (0);
}
```

Qualora il compilatore compili questo programma, quali sono i valori stampati a video a tempo di esecuzione? (si motivi la risposta in maniera opportuna)

Si spieghi inoltre perchè nel passaggio di una matrice come parametro di una funzione, non è significativa la prima dimensione mentre la seconda (nel caso la matrice sia bi-dimensionale) è invece necessaria

Soluzioni

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DIM 128

typedef struct {
    char customer[DIM];
    int transactionId;
    float value;
} transaction;

int select(FILE *source, FILE *dest, float threshold) {
    transaction temp;
    int result;
    result = 0;
    while (fread(&temp, sizeof(transaction), 1, source) > 0) {
        if (temp.value >= threshold) {
            fprintf(dest, "%s %d %f\n", temp.customer, temp.transactionId,
temp.value);
            result++;
        }
    }
    fprintf(dest, "\n");
    return result;
}

int main() {
    FILE *source;
    FILE *dest;
    int result = 0;
    float threshold;
    printf("Insert threshold: ");
    scanf("%f", &threshold);
    if ((source = fopen("log.dat", "rb")) == NULL) {
        printf("Error opening the file %s\n", "log.dat");
        exit(-1); }
    if ((dest = fopen("debitori.txt", "w")) == NULL) {
        printf ("Error opening the file %s\n", "debitori.txt");
        exit(-1); }

    result = select(source, dest, threshold);
    fclose(source);
    fclose(dest);
    printf("%d records copied by log.dat to %s\n", result, "debitori.txt");
    return 0;
}
```

ESERCIZIO 2

```
int isAdd(list l1, list l2, list add) {
    int problem = 0;
    while (!empty(l1) && !empty(add) && !problem) {
        if ((head(l1) + head(l2)) == head(add) ) {
            l1 = tail(l1);
            l2 = tail(l2);
            add = tail(add);
        }
    }
}
```

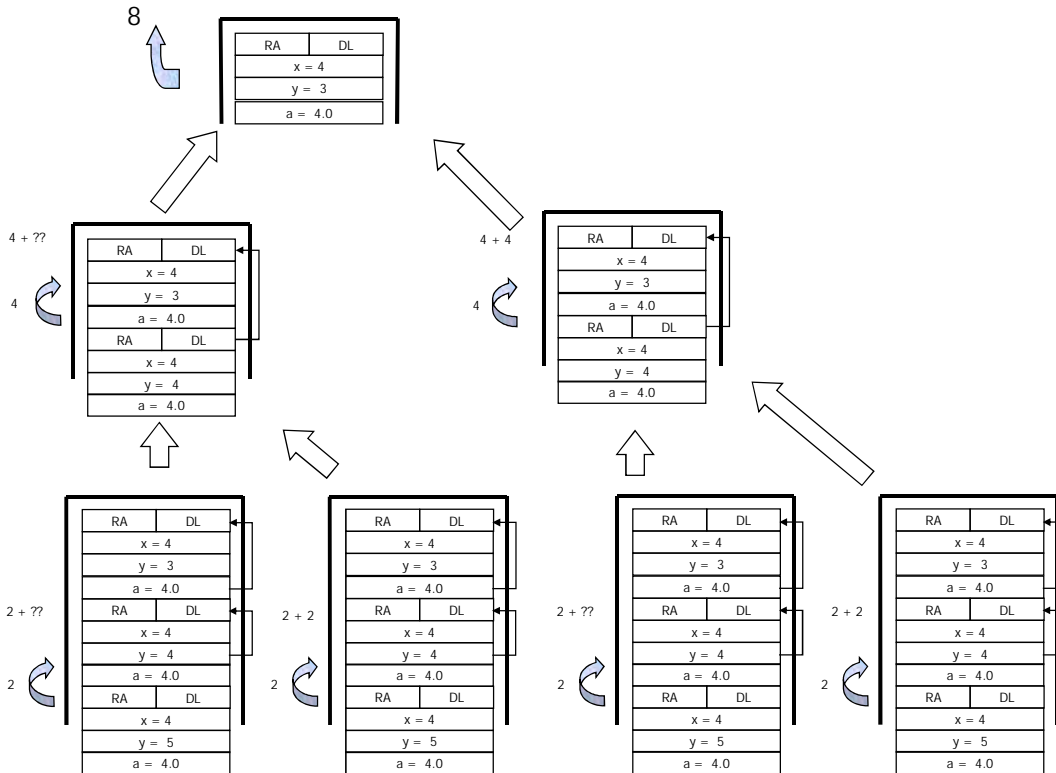
```

    }

    else problem=1;
}
if (!problem && empty(l1) && empty(add) ) return 1;
else return 0;
}

```

ESERCIZIO 3



ESERCIZIO 5

Il programma è corretto, compila correttamente e stampa:

1 2 3
10 5 6

Si noti che, siccome nella dichiarazione della funzione la seconda dimensione dell'array passato come argomento è differente dalla seconda dimensione dell'array (stando a come è stato dichiarato nel main), la cella [1][1] riferita nella funzione è differente dalla cella [1][1] riferita nel main.