Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Martedì 19 Aprile 2005 – durata 2h30m
COMPITO A

ESERCIZIO 1 (12 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 128 caratteri, senza spazi);
- un intero **`transactionId`**, recante un codice identificativo della transazione;
- un float **`value`**, contenente l'ammontare del credito concesso.

Al momento di riscuotere i crediti, il commerciante deve però poter accedere ai valori registrati nel file binario. A tal scopo, egli vuole avere un programma che, richiesto il nome del cliente, scriva su un file in formato testo gli importi relativi solo al cliente specificato. Inoltre il file deve essere avere come nome il nome del cliente con l'aggiunta dell'estensione ".txt". Ad esempio, se il cliente richiesto si chiama "Federico", allora il file dovrà chiamarsi "Federico.txt". Si realizzi:

1. una funzione **`copy(…)`** che, ricevuti in ingresso un puntatore **`source`** al file binario, un puntatore **`dest`** al file di testo, un puntatore a carattere **`name`** e un intero **`result`** passato per riferimento, copi su **`dest`** tutti gli importi presenti in **`source`** e relativi al cliente specificato con parametro **`name`**. La funzione deve tenere traccia del numero di importi di credito copiati e deve restituire tale numero tramite il parametro **`result`**. Gli importi devono essere scritti su una sola linea, separati da uno spazio, con al termine un carattere di "newline" ('\n'). Al fine di confrontare due stringhe, si utilizzi la funzione **`strcmp(…)`**, che restituisce 0 se le due stringhe passate come parametri sono identiche. **(punti 6)**
2. un programma C che chieda inizialmente all'utente il nome di un cliente. Per creare un opportuno nome per il file di destinazione, il candidato può utilizzare le funzioni di libreria:
   - **`strcpy(char *s, char *ct)`**, che provvede a copiare la stringa **`ct`** nella stringa **`s`**;
   - **`strcat(char * s, char * ct)`**, che concatena il contenuto della stringa **`ct`** in fondo alla stringa **`s`** (si faccia particolare attenzione a dimensionare opportunamente la stringa **`s`** per contenere i 4 caratteri dell'estensione ".txt"). La stringa **`s`**, al termine dell'invocazione, è sempre una stringa ben formata.

   Dopo aver aperto i file nell'opportuna modalità di lettura/scrittura, il programma utilizzi la funzione **`copy(…)`** definita al punto precedente per filtrare i dati. Il programma stampi infine a video il numero totale di crediti che sono stati copiati da un file all'altro. **(punti 6)**

ESERCIZIO 2 (12 punti)

È dato un file di testo, di nome "**`log.txt`**", contenente una sequenza di **`float`** separati da spazi. Non è noto a priori quanti numeri vi siano nella sequenza. L'obiettivo è salvare tali numeri in un **`array`** allocato dinamicamente, la cui dimensione corrisponda esattamente alla lunghezza della sequenza memorizzata sul file. A tale scopo:

1. si definisca una funzione **`read(…)`** che, ricevuto in ingresso il nome del file (puntatore a **`char`**), restituisca una lista contenente i valori presenti sul file (si supponga di avere a disposizione le operazioni primitive per la gestione di liste presentate a lezione). **(4 punti)**
2. si definisca una funzione **`convert(…)`** che, ricevuta in ingresso una lista **`l`**, un puntatore **`ar`** a una zona di memoria già allocata dinamicamente con l'opportuna dimensione, e un intero **`dim`** passato per riferimento,

copi i valori della lista nella zona di memoria. Il numero di valori copiati deve essere restituito tramite l'intero **`dim`**. **(5 punti)**

3. si definisca un programma **`main`** che legga dal file "**`log.txt`**" i valori numerici tramite la funzione **`read(…)`**; il programma deve poi allocare dinamicamente memoria sufficiente per contenere i valori letti, e tramite la funzione **`convert(…)`** deve copiare i valori della lista restituita da **`read(…)`** nell'area di memoria allocata dinamicamente. Al fine di determinare la lunghezza di una lista, il candidato può utilizzare la funzione di libreria **`int length(list)`**. Infine, il programma **`main`** deve stampare a video i valori caricati nel vettore dinamico, utilizzando la notazione sintattica tipica degli array. **(3 punti)**

Il candidato ipotizzi di avere a disposizione le operazioni primitive sulle liste viste a lezione, che pertanto possono non essere riportate nello svolgimento del compito.

ESERCIZIO 3 (4 punti)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione (si motivi opportunamente la risposta data)?

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

#define DIM 16

void ribalta(char a[], char *b, int *dim) {
      int i, size = 0;
      *dim = 0;
      for (size=0; a[size] != '\0'; size++);
      for (i=size-1; i>=0; i--) {
            *(b+size-i-1) = a[i];
            (*dim)++; }
      *(b+size) = '\0';
      (*dim)++;
}

int main () {
      char mese[] = "Aprile";
      char *other;
      int *value, i;

      other = (char *) malloc(DIM * sizeof(char));
      value = (int  *) malloc(sizeof(int));
      *value = 0;

      ribalta(mese, other, value);
      for (i=0; i < (*value) - 1; i++) {
            printf("%c", other[i]);
      printf("%d\n", *value); return 0;
}
```

ESERCIZIO 4 (4 punti)

Si descriva brevemente in che cosa consiste la differenza tra una funzione ricorsiva e una funzione ricorsiva tail. Si presentino poi due brevi e semplici esempi di funzione al fine di mostrare tale differenza.

SOLUZIONE

ESERCIZIO 1

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

#define DIM 129

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

void copy(FILE *source, FILE *dest, char *name, int *result) {

      transaction temp;

      *result = 0;
      while (fread(&temp, sizeof(transaction), 1, source) > 0) {
            if (strcmp(name, temp.customer) == 0) {
                  fprintf(dest, "%f ", temp.value);
                  (*result)++;
            }
      }
      fprintf(dest, "\n");
}

int main() {
      char name[DIM], filename[DIM+4];
      FILE *source;
      FILE *dest;
      int result = 0;

      printf("Insert customer name: ");
      scanf("%s", name);

      if ((source = fopen("log.dat", "rb")) == NULL) {
                  printf("Error opening the file %s\n", "log.dat");
                  exit(-1);
      }

      strcpy(filename, name);
      strcat(filename, ".txt");
      if ((dest = fopen(filename, "w")) == NULL) {
                  printf ("Error opening the file %s\n", filename);
                  exit(-1);
      }

      copy(source, dest, name, &result);

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

ESERCIZIO 2

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

list read(char *filename) {
      list result;
      float temp;
      FILE * f;
      if ((f = fopen(filename, "r")) == NULL) {
                  printf ("Error opening the file %s\n", filename);
                  exit(-1); }
      result = emptylist();
      while (fscanf(f, "%f", &temp) != EOF)
            result = cons(temp, result);
      fclose(f);
      return result; }

void convert(list l, float *array, int *dim) { // versione ricorsiva…
      if (empty(l)) return;
      else { *array = head(l);
            (*dim)++;
            convert(tail(l), ++array, dim);
            }
}

int main() {
      list l;
      int size, dim = 0, i;
      float *array;

      l = read("federico.txt");
      size = length(l);
      array = (float*) malloc(sizeof(float) * size);
      convert(l, array, &dim);
      for (i=0; i<size; i++)
            printf("%f\n", array[i]);
      return 0; }
```

ESERCIZIO 3

Il programma è corretto sintatticamente, viene compilato, ed in esecuzione stampa:
elirpA 7

Nella fase iniziale del programma **`main`** vengono dichiarate alcune variabili e allocata memoria dinamicamente; in questa fase si inizializza anche a 0 la variabile riferita dal puntatore **`value`**. Quindi viene invocata la funzione **`ribalta(…)`**: tale funzione esegue inizialmente un ciclo con lo scopo di determinare la lunghezza della stringa **`a`**, e poi copia (ribaltando) il contenuto di **`a`** in **`b`**. L'operazione di inversione dell'ordine dei caratteri è effettuata tramite opportune somme/sottrazioni negli indici. La funzione termina copiando anche il terminatore di stringa (al termine della stringa), e restituisce tramite il parametro **`dim`** il numero di caratteri copiati (compreso il terminatore), cioè 7.
Il **`main`** stampa tutti i caratteri della stringa **`other`**, terminatore escluso, e di seguito il valore puntato da **`value`** che, per quanto detto, è 7.